ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 7, July 2016

Survey on Community Detection Algorithms in Complex Network Analysis

M.SATHYAKALA

Department of Information Technology,
Institute of Road and Transport Technology,
Erode, Tamilnadu, India.

Abstract: Community detection is an essential technique for evaluating hidden details like structural and topological features of Complex network structure. Over the last few years, a variety of algorithms have been proposed to detect communities in a complex network. Even then, there are issues with how to evaluate the perfection and efficiency of techniques is indeed open. The aim of this paper is to examine and evaluate existing works and methodologies for community detection, in the expectation of supporting researchers in the field concerned.

Keywords: community detection, complex network

I. INTRODUCTION

Many structures can be modelled as complex networks in the real world. Social networks, biological networks and transportation networks are examples of such structures. The web page is known to be nodes on the internet, and the links to another web page are considered edges. Significant key findings and the application of the Community Structure Investigation in a complex network make this community detection problem a major attraction in the field of computer science[1]. It has also received significant importance from research in the fields of physics, genetics and mathematics.

Due to the increased interest in this field, a number of algorithms have been proposed over the last few years. These algorithms need to be classified, analyzed and compared. Grouping and performance analyzes help researchers in this field to select the algorithm that is best suited to their problems. It also helps to find the advantage and shortcoming of these algorithms, thus opening up the issues that need to be addressed in this field of complex network analysis.

This paper is structured as follows. The next section defines the terms and basic concepts of the problem. The proposed algorithm for community detection is discussed in Section 3. The efficiency of these algorithms is compared and its pros and cons are also discussed in section 4. Finally, the paper concludes the review of the algorithms.

1. Preliminaries

2.1 Community

A community (also called cluster) [1] in a network is a group of vertices (i.e. a sub-graph) with densely connected edges within community, and with sparsely connected edges between groups.

2.2 Representation of graphs

A graph G can be represented as Adjacency Matrix A, where A_{ij} represents the element of the matrix. $A_{ij} = 1$, if there exits an edge between nodes 'i' and 'j' in unweighted graph, $A_{ij} = w$ if there exits an edge with weight w between nodes 'i' and 'j' in weighted graph. If the graph is signed the weight value is represented as (-1, 0,+1) depending on the signed value of the edge.

If the graph is undirected then $A_{ij} = A_{ji}$, otherwise if the graph is directed then $A_{ij} \neq A_{ji}$.

II. ALGORITHMS

1.1. Traditional Methods

One of the very first methods to find community in a network is **Kernighan–Lin algorithm[2].** Kernighan-Lin is a method of partitioning a graph containing nodes and vertices into separate subsets that are connected together in an optimal manner. Since a graph can be used to represent an electrical network containing blocks, the Kernighan-Lin algorithm can be extended to partitioning circuits into sub-circuits. It's an iterative algorithm rather than a constructive one, and it's a greedy algorithm. This means that the algorithm will make changes if there is a benefit right away without considering other possible ways of obtaining an optimal solution. It's a deterministic meaning that gives the same solution for every algorithm run.

The KL algorithm strategy is: initially divide a graph into subsets of vertices, such an initial partition may be random, or based on some recommended information on the graph structure. Quality function Q are introduced to measure the quality of the graph Partition. Then move the vertex in one subset to another, or swap two different vertexes subsets so that Q has the maximum increase in each iteration. The partition with the highest value of Q is finally the result.

1.1.1. Advantage

The Algorithm is more reliable.

1.1.2. Disadvantage

Results are random because the algorithm starts with a random partition, and Computationally intensive which makes the algorithm slow. In addition, the greatest drawback of the KL algorithm is the need for prior knowledge, this implies that a poor initial partition typically leads to an undesirable final outcome or speed at convergence.

1.1.3. Time Complexity

The time complexity of the algorithm is O(n^2 log n) where n is the number of vertices in the graph.

1.2. Hierarchical Clustering

Hierarchical clustering is an algorithm that groups related items into groups called clusters, also known as hierarchical cluster analysis. The outcome is a collection of clusters whereby each cluster is independent from one another, so that each cluster's artifacts are generally identical to one another. There are essentially two directions that can be categorized:

i) Agglomerative algorithm- It is a bottom up approach in which the process starts from single vertex and iteratively adds vertices which is very similar to it.

ii) Divisive algorithms –It is a top down approach in which the process starts from entire vertex and removes the vertex and the edges between vertex that are dissimilar.

Both the above methods are based on similarity measures. The quality and performance of the algorithm mainly depends on the similarity measures used. Various similarity measures like Euclidean distance, Manhattan distance, Minkowski distance, Cosine Similarity, Jaccard Similarity and Salton Index are available in literature.

1.2.1. Divisive algorithms

Newman and Girvan in the year 2004 proposed a Girvan–Newman algorithm [3] based on divisive hierarchical clustering . The algorithm can be used when the number of cluster is unknown. In that case the algorithm itself is used to find a partition in K ∈ {1, …. n}. To decide which of these partitions to pick as the quality measure for the true partition the so called Newman-Girvan modularity is used. The betweenness measures of the edge is used find the edge score. This edge score is used to identify the classes of network. The procedure of edge removal based on betweenness score ends when the modularity reaches maximum limit. The time complexity of the algorithm is O(n^3).

The other notable works are given below

i) Tyler et al.[5] proposed a modification of the Girvan–Newman algorithm, to improve the speed of the calculation.

ii) R. Zhang et al.[6] work the community detection algorithm based on the node clustering coefficient and the edge clustering coefficient

1.2.2. Agglomerative algorithm

Fast-Newman is an example of Agglomerative algorithm[4]. The links of the original graph are iteratively added from a set of isolated nodes, to produce the greatest possible increase in the modularity of Newman and Girvan at each step. It merges node clusters with the most gain or the least loss of modularity to ultimately find communities. It Combines node clusters to the most modularity gain or the minimum loss of modularity to eventually discover communities.

The other notable works are given below

i) R. Langone et al.[7] work in Kernel spectral clustering for community detection in complex networks.

ii) L. Lin et.al.[8] work in A new community detection based on agglomeration mechanism.

1.2.3. Advantage of Hierarchical method

The benefits of hierarchical approaches are that they can start without prior knowledge of the number of vertices.

1.2.4. Disadvantage of hierarchical method

Failed to discriminate good from bad in a number of outcomes obtained and the performance is strongly dependent on the measure of similarity adopted. It always yields a hierarchical result, whereas in many cases this is not optimum. It does not scale well for large networks.

1.2.5. Time complexity

If n represents the number of elements to be clustered and k represents the number of clusters, then the time complexity of the hierarchical algorithms is O (kn^2).

1.3. Modularity Based Methods

A Modularity based methods works on the concept of maximizing modularity in optimization methods for detecting community structure in networks. A lot of work have been done on this path. Interestingly, several heuristic strategies have been implemented to find high-modularity partitions within a reasonable period. They are categorized as greedy approach, spectral techniques, and Simulated Annealing.

1.3.1. Greedy Approach

A greedy approach from Newman was the first algorithm invented to optimize modularity. It is an agglomerative method of hierarchical clustering, in which groups of vertices are joined successively to form larger communities in such a way that modularity improves after combining. The next work by Clauset et al. improves the above method by reducing the computational complexity by using three data structures max heap, binary tree and simple array. This work is a promising one for large network of about 10^6 vertices. The other important works are given below

i) Wakita and Tsurumi [9] work in Finding community structure in mega-scale social networks

ii) Xiang et al. [10] work in Finding community structure based on subgraph similarity

iii) H. Du et.al. [11] work in an algorithm for detecting community structure of social networks based on prior knowledge and modularity Complexity.

1.3.2. Spectral Techniques

The variant work on the spectral techniques are i) the Modularity optimization using the eigenvalues and eigenvectors of the modularity matrix. Here the spectral method, divides a network into more than two communities by repeated division and then uses the leading eigenvector of the modularity matrix to assign communities. ii) Modularity optimization using the eigenvalues and eigenvectors of the Laplacian matrix here the underlying spectral algorithm is analogous to the problem of standard spectral graph partitioning that uses the Laplacian matrix's eigenvalues and eigenvectors. This algorithm begins with a single community and divides each community recursively into two smaller ones if the subdivision produces a higher Q value and the recursive process ends when no further splits are possible or when communities have been found and then the final community structure with the highest Q value is the outcome of detection.t. and iii) Equivalence of two categories of spectral algorithms for maximizing modularity here The method of optimizing spectral modularity using the modularity matrix's eigen values and eigenvectors can be formulated as a spectral algorithm that relies on the Laplacian matrix's eigen values and eigenvectors. This formulation suggests that the two forms of approaches to modularity optimization above are equivalent.

The noted works are

i) White et.al. [12] work in a spectral clustering approach to finding communities in graph

ii) M. E. J. Newman [13] work in Finding community structure in networks using the eigenvectors of matrices and Spectral methods for network community detection and graph partitioning.

1.3.3. Simulated annealing [14]

It is a probabilistic technique for the global optimization problem to locate a good approximation to the global optimum of a given function in a large search space. Arbitrary partitioning of nodes into communities may be the initial point for all these methods, including communities in which each node belongs to its own community. A node and a community are picked at random at each iteration. This community may be an established community or an empty community added to increase the number of communities. Then the node is relocated to this new community from its original community, which will shift by modularity value Q. Such methods stop when within a specified number of iterations, no new update is approved.

1.3.4. Advantage

It is an important measure to assess the quality of community obtained.

1.3.5. Disadvantage

The modularity suffers from a problem resolution limit and, therefore, it is unable to detect small communities. This performance can be obtained with the loss of small communities.

2. Quality metrics

Most of the community detection algorithms strive to optimize a goodness metric that basically indicates the efficiency of the communities identified from the network in order to obtain the best partition of the graph and thus important communities. The objective of the algorithm for group detection would be to obtain the best network partition that would optimize the metric. A broad range of such metrics have been suggested that can detect the consistency of the communities acquired for a given partition.

Let us consider a function f (c) that, on the basis of the connectivity of nodes in it, implies the fairness of the community c. Then the following are the some metrics used to measure the quality of community obtained.

2.1. Internal density [15]

$$f\ (c)\ =\ \frac{2*|E_c^{in}\ |}{|c\ |(|c\ |-1)} \qquad (1)$$

Where E_c^{in} refers number of edges within community and is defined as the ratio of number of internal edges in the community to the possible number of edges in the community.

2.2. Edge inside [15]

$$f\ (c)\ =\ |E_c^{in}\ | \qquad (2)$$

is defines as the inter edge strength of the community.

2.3. Average degree [15]

$$f\ (c)\ =\ \frac{2*|E_c^{in}\ |}{|c\ |} \quad (3)$$

Is defined as the average degree of nodes of c by considering internal edges of the community only.

2.4. Fraction over median degree (FOMD) [15]: By measuring the number of nodes in c that have an internal degree greater than dm, the median value of the degree of all nodes in V determines how tightly the group is bound. The total number of nodes within the community is normalized by this ranking.

$$f\ (c)\ =\ \frac{\left||(u,v):u\in c\&v\in c\ |>d_m\right|}{|c\ |} (4)$$

2.5. Cut Ratio (or Ratio Cut)[16]: Calculates the fraction of the edges of all possible edges left outside community c.

$$f\ (c)\ =\ \frac{|E_c^{out}\ |}{|c\ |*(N-|c|)} \quad (5)$$

2.6. Conductance [17]: measures the ratio of the total number of outgoing edges from c to the total number of edges with in c.

$$f\ (c)\ =\ \frac{|E_c^{out}\ |}{2*|E_c^{in}\ |+|E_c^{out}\ |} \quad (6)$$

2.7. Normalized cut [16]: This normalizes the cut score

$$f\ (c) = \frac{|E_c^{out}\ |}{2*|E_c^{in}\ |+|E_c^{out}\ |} + \frac{|E_c^{out}\ |}{2*(m-|E_c^{in}\ |)+|E_c^{out}\ |} \quad (7)$$

III. Conclusion

We reviewed different algorithms in this paper, ranging from conventional ones to the recently proposed ones. For each algorithm/method, we evaluated the benefits and disadvantages. We also supplied many widely used algorithm scoring functions. We now seem to have an efficient toolkit for researching group structure in networks as a result of considerable progress in recent years. In both the speed and sensitivity of group structure algorithms, there is definitely still space for development, and there are many complex networked systems requiring research using these methods.

References

[1]. Fortunato, S., 2010. Community detection in graphs. *Physics reports*, *486*(3-5), pp.75-174.

[2]. Kernighan, B.W. and Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, *49*(2), pp.291-307.

[3]. Newman, M.E. and Girvan, M., 2004. Finding and evaluating community structure in networks. *Physical review E*, *69*(2), p.026113.

[4]. Girvan, M. and Newman, M.E., 2001. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, *99*(cond-mat/0112110), pp.8271-8276.

[5]. Tyler, J.R., Wilkinson, D.M. and Huberman, B.A., 2003. Email as spectroscopy: Automated discovery of community structure within organizations. In *Communities and technologies* (pp. 81-96). Springer, Dordrecht.

[6]. Zhang, R., Li, L., Bao, C., Zhou, L. and Kong, B., 2014, June. The community detection algorithm based on the node clustering coefficient and the edge clustering coefficient. In *Proceeding of the 11th World Congress on Intelligent Control and Automation* (pp. 3240-3245). IEEE.

[7]. Langone, R., Alzate, C. and Suykens, J.A., 2012, June. Kernel spectral clustering for community detection in complex networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[8]. Lin, L., Luo, T., Fu, J., Ji, Z. and Xiao, D., 2011, August. A new community detection based on agglomeration mechanism. In *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering* (Vol. 1, pp. 352-355). IEEE.

[9]. Wakita, K. and Tsurumi, T., 2007, May. Finding community structure in mega-scale social networks. In *Proceedings of the 16th international conference on World Wide Web* (pp. 1275-1276).

[10]. Xiang, B., Chen, E.H. and Zhou, T., 2009. Finding community structure based on subgraph similarity. In *Complex networks* (pp. 73-81). Springer, Berlin, Heidelberg.

[11]. Du, H., Feldman, M.W., Li, S. and Jin, X., 2007. An algorithm for detecting community structure of social networks based on prior knowledge and modularity. *Complexity*, *12*(3), pp.53-60.

[12]. White, S. and Smyth, P., 2005, April. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM international conference on data mining* (pp. 274-285). Society for Industrial and Applied Mathematics.

[13]. Newman, M.E., 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, *74*(3), p.036104.

[14]. Liu, J. and Liu, T., 2010. Detecting community structure in complex networks using simulated annealing with k-means algorithms. *Physica A: Statistical Mechanics and its Applications*, *389*(11), pp.2300-2309.

[15]. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. and Parisi, D., 2004. Defining and identifying communities in networks. *Proceedings of the national academy of sciences*, *101*(9), pp.2658-2663.

[16]. Shi, J. and Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), pp.888-905.

[17]. Wei, Y.C. and Cheng, C.K., 1991. Ratio cut partitioning for hierarchical designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *10*(7), pp.911-921.