International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Survey on Community Detection

Suhas S Thorat[1], Sharmila M Shinde[2]

[1]Pune University, Dept of Computer Engineering, JSPM College of Engineering, Pune, India

[2]Pune University, Dept of Computer Engineering, JSPM College of Engineering, Pune, India

Abstract: ***Empirical studies and theoretical modeling of networks has been the subject of a large body of recent research. Network ideas have been applied with great success to topics as diverse as the Internet and the World Wide Web. The graph or the network is a powerful tool to characterize the complex relations between a set of instances by taking each instance as a vertex and the interaction between a pair of vertices as an edge. Many complex systems can be modelled and analyzed as complex networks such as technological networks, social networks and biological networks and so on. A property that seems to be common to many networks is community structure, the division of network nodes into groups within which the network connections are dense, but between which they are sparser. It has been proved that many real world networks reveal the structures of the modules or the communities that are sub graphs with more edges connecting the vertices of the same group and comparatively fewer links joining the outside vertices. The Modules or the communities reflect the topological relations between the elements of the underlying system and the functional entities.***

Keywords: community detection, network structures, partitioning, modularity optimization.

1. Introduction

A very widespread informal definition of the community concept considers it as a group of nodes densely interconnected compared to the rest of the network. In other terms, a community is a cohesive subset clearly separated from the rest of the network. Formal interpretations try to formalize and combine both these aspects of cohesion and separation. Note this definition is not always explicit: procedural approaches exist, in which the notion of community is implicitly defined as the result of the processing. Although it is not always straightforward to categorize the definitions, we regroup them in four classes: density-, pattern-, node similarity- and link centrality-based approaches [4].

Communities can have concrete applications. Clustering Web clients who have similar interests and are geographically near to each other may improve the performance of services provided on the World Wide Web, in that each cluster of clients could be served by a dedicated mirror server. Identifying clusters of customers with similar interests in the network of purchase relationships between customers and products of online retailers (like, e. g., www.amazon.com) enables to set up efficient recommendation systems, that better guide customers through the list of items of the retailer and enhance the business opportunities [4].

Community detection is important for other reasons, too. Identifying modules and their boundaries allows for a classification of vertices, according to their structural position in the modules. So, vertices with a central position in their clusters, i.e. sharing a large number of edges with the other group partners, may have an important function of control and stability within the group; vertices lying at the boundaries between modules play an important role of mediation and lead the relationships and exchanges between different communities. Such classification seems to be meaningful in social and metabolic networks. Another important aspect related to community structure is the hierarchical organization displayed by most networked systems in the real world. Real networks are usually composed by communities including smaller communities, which in turn include smaller communities, etc. The aim of community detection in graphs is to identify the modules and, possibly, their hierarchical organization, by only using the information encoded in the graph topology. The problem has a long tradition and it has appeared in various forms in several disciplines [2].

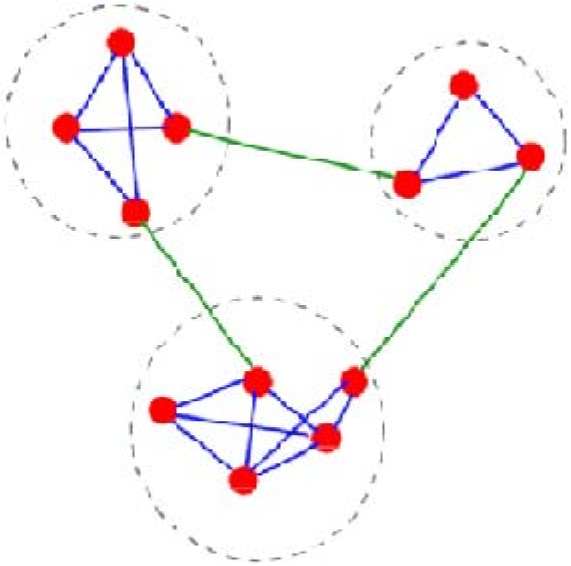

Figure 1: A simple graph showing communities, enclosed by the dashed circles

The investigation of the community structure inside networks has acquired a great relevance during the last years, in particular in the context of Social Network Analysis (SNA). This, also because of the unpredicted success of Online Social Networks (OSNs). In fact, social phenomena such as Facebook and Twitter amongst others, glue together millions of users under a unique network whose features are a goldmine for Social Scientists. Several works are focused on the Social Network analysis of these OSNs; others describe the strategies of analysis themselves [2].

2. Literature Review

The study of community structure in networks has a long history. It is closely related to the ideas of graph partitioning in graph theory and computer science, and hierarchical clustering in sociology. Finding communities within an arbitrary network can be a computationally difficult task. The number of communities, if any, within the network is typically unknown and the communities are often of unequal size and/or density. Despite these difficulties, however, several methods for community detection have been developed and each has its own advantages/disadvantages. Furthermore, the number of inter-community edges needn't be strictly minimized either, since more such edges are admissible between large communities than between small ones.

Overview of Community Detection Methods

The problem of graph clustering, intuitive at first sight, is actually not well defined. The main elements of the problem themselves, i.e. the concepts of community and partition, are not rigorously defined, and require some degree of arbitrariness and/or common sense. Indeed, some ambiguities are hidden and there are often many equally legitimate ways of resolving them.

It is important to stress that the identification of structural clusters is possible only if graphs are sparse, i.e. if the number of edges m is of the order of the number of nodes n of the graph. If m >> n, the distribution of edges among the nodes is too homogeneous for communities to make sense. In this case the problem turns into something rather different, close to data clustering, which requires concepts and methods of a different nature. The main difference is that, while communities in graphs are related, explicitly or implicitly, to the concept of edge density (inside versus outside the community), in data clustering communities are sets of points which are “close” to each other, with respect to a measure of distance or similarity, defined for each pair of points [1].

Below are broad level categories of the different methods for community detection:

A. Partitioning

In these methods, the network is partitioned into a predetermined number of groups, usually of approximately the same size, chosen in a way that the number of edges between groups is minimized. These methods find communities regardless of whether they are implicit in the structure or not, and it will find only a fixed number of them. This method is not always an ideal method for finding community structure in general networks.

B. Hierarchical clustering

Hierarchical clustering is another method for finding community structures in networks. These methods use a similarity measure quantifying some (usually topological) type of similarity between node pairs. Commonly used measures include the cosine similarity, the Jaccard index, and the Hamming distance between rows of the adjacency matrix. Then the similar nodes are grouped into communities according to this measure. There are several common schemes for performing the grouping, the two simplest being single-linkage clustering, in which two groups are considered separate communities if and only if all pairs of nodes in different groups have similarity lower than a given threshold, and complete linkage clustering, in which all nodes within every group have similarity greater than a threshold [4].

C. Modularity optimization

Modularity optimization is one of the most widely used methods for community detection. Modularity is a benefit function that measures the quality of a particular division of a network into communities. The modularity optimization method detects communities by searching over possible divisions of a network for one or more that have particularly high modularity. Since exhaustive search over all possible divisions is usually intractable, practical algorithms are based on approximate optimization methods such as greedy algorithms, simulated annealing, or spectral optimization, with different approaches offering different balances between speed and accuracy [3][5][6].

D. Statistical inference

Methods based on statistical inference attempt to fit a generative model to the network data, which encodes the community structure. The overall advantage of this approach compared to the other methods is its more principled nature, and the capacity to inherently address issues of statistical significance.

E. Clique based methods

Cliques are sub graphs in which every node is connected to every other node in the clique. As nodes cannot be more tightly connected than this, there are many approaches to community detection in networks based on the detection of cliques in a graph.

3. Elements of Community Detection

Many networks of interest in the sciences are found to divide naturally into communities or modules. The problem of detecting and characterizing this community structure is a key step for understanding complex networks. The idea of community detection is closely related to data clustering, graph partitioning, and hierarchical clustering. Therefore, traditional approaches in these areas can be employed for community detection. Two key approaches that have been widely investigated in community detection are: 1) spectral clustering-based techniques and 2) network modularity optimization strategies. Spectral clustering-based approaches rely on the optimization of the process of cutting the graph representing the given network. Since this problem is NP-hard, different approximate techniques such as the normalized cuts algorithm and ratio cuts algorithm have been proposed. The main problem with spectral clustering-based techniques is that one has to know in advance the number and the size of communities in the network. Network modularity-based methods, on the other hand, rely on the modularity function Q to determine the optimal number of clusters in the network. A good partitioning of a network is expected to have high modularity Q with Q=(fraction of edges within communities)-(expected fraction of such edges), where the expected fraction of edges is evaluated for a random graph. For a directed weighted network represented by a graph G = (V, E) with N nodes and an association matrix A, the modularity function is given as [4]:

$$Q=1/W \sum_{i,j=1}^{N}[A_{ij}-(S_i^{out} S_j^{in})/W]\delta_{C_i,C_j} \quad (1)$$

Where

A_{ij} is the weight of edge $e_{i\to j}$

$S_i^{in} \sum_j A_{j,i}$, $S_i^{out} \sum_j A_{i,j}$ is the inflow, outflow of the node

$i,W= \sum_{i,j} A_{i,j}, C_i C_j$ is the community that node(i,j) belongs to

$\delta_{C_i C_j}$ is equal to 1 when i and j are in the same community and is equal to 0 otherwise.

4. Computational Complexity

The massive amount of data on real networks currently available makes the issue of the efficiency of clustering algorithms essential. The computational complexity of an algorithm is the estimate of the amount of resources required by the algorithm to perform a task. This involves both the number of computation steps needed and the number of memory units that need to be simultaneously allocated to run the computation. Such demands are usually expressed by their scalability with the size of the system at study. In the case of a graph, the size is typically indicated by the number of vertices n and/or the number of edges m. The computational complexity of an algorithm cannot always be calculated. In fact, sometimes this is a very hard task, or even impossible. In these cases, it is however important to have at least an estimate of the worst-case complexity of the algorithm, which is the amount of computational resources needed to run the algorithm in the most unfavourable case for a given system size. The notation $O(n^{\alpha} m^{\beta})$ indicates that the computer time grows as a power of both the number of vertices and edges, with exponents α and β, respectively.

Algorithms with polynomial complexity form the class P. For some important decision and optimization problems, there are no known polynomial algorithms. Finding solutions of such problems in the worst-case scenario may demand an exhaustive search, which takes a time growing faster than any polynomial function of the system size, e.g. exponentially. Problems whose solutions can be verified in a polynomial time span the class NP of nondeterministic polynomial time problems, which includes P. A problem is NP-hard if a solution for it can be translated into a solution for any NP-problem. However, a NP-hard problem needs not be in the class NP. If it does belong to NP it is called NP-complete. The class of NP-complete problems has drawn a special attention in computer science, as it includes many famous problems like the Travelling Salesman, Boolean Satisfiability (SAT), Linear Programming, etc. The fact that NP problems have a solution which is verifiable in polynomial time does not mean that NP problems have polynomial complexity, i.e., that they are in P. In fact, the question of whether NP=P is the most important open problem in theoretical computer science. NP-hard problems need not be in NP (in which case they would be NP-complete), but they are at least as hard as NP-complete problems, so they are unlikely to have polynomial complexity, although a proof of that is still missing. Many clustering algorithms or problems related to clustering are NP-hard. In this case, it is pointless to use exact algorithms, which could be applied only to very small systems. Moreover, even if an algorithm has a polynomial complexity, it may still be too slow to tackle large systems of interest. In all such cases it is common to use approximation algorithms, i.e. methods that do not deliver an exact solution to the problem at hand, but only an approximate solution, with the advantage of a lower complexity. Approximation algorithms are often non-deterministic, as they deliver different solutions for the same problem, for different initial conditions and/or parameters of the algorithm. The goal of such algorithms is to deliver a solution which differs by a constant factor from the optimal solution. In any case, one should give provable bounds on the goodness of the approximate solution delivered by the algorithm with respect to the optimal solution. In many cases it is not possible to approximate the solution within any constant, as the goodness of the approximation strongly depends on the specific problem at study. Approximation algorithms are commonly used for optimization problems, in which one wants to find the maximum or minimum value of a given cost function over a large set of possible system configurations [4].

The problem of maximizing the network modularity has been proven to be NP complete. For this reason, several heuristic strategies to maximize the network modularity such as Girvan Newman algorithm, the fast clustering algorithm, the external optimization method and the Newman Leicht mixture model-based approach have been proposed. Although most of the modularity-based community detection algorithms have focused on binary and undirected networks, in recent years there have been some extensions to weighted and directed networks. However, these approaches are limited to networks with a small number of clusters. Recently, Blondel et al. introduced an alternative greedy algorithm, which is known as the Louvain method, to find the hierarchical structure of undirected weighted graphs. Compared to other methods, this method performs better in terms of the computation time especially for networks with a large number of nodes.

5. Conclusion

The rate of information development growth has been increased tremendously because of the World Wide Web. Despite the fact that exploration on community detection began around 50 years prior, there is still a long trail to stroll in this field. This review accentuates some of the methodologies for community detection. As to date, two paradigms exist to discover the community structure of a network. The former is based on the analysis of the global features of the network, for example its topology. These approaches are characterized by high computational complexity and high quality results. The latter paradigm relies on exploiting local information, for example those acquirable by nodes and their neighborhoods. The computational cost of these techniques is lower than those exploiting global features, but the reliability decreases. There could be numerous possibilities to improve the efficiency and performance of various algorithms of community detection.

6. Acknowledgement

I would like to thank my guides Prof. Sharmila M. Shinde and Prof. Darshana R. Patil for their help and guidance throughout this project and the semester, without them this would not have been possible.

References

[1] Songwei Jia, Lin Gao, Yong Gao, and Haiyang Wang, “Anti-triangle centrality based community detection in complex networks ”, IET Systems Biology, 2013
[2] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, Alessandro Provetti, “Generalized Louvain method for community detection in large networks ”, Proceedings of the 11th International Conference On Intelligent Systems Design And Applications, 2011
[3] Ludo Waltman and Nees Jan van Eck,“A smart local moving algorithm for large-scale modularity-based community detection ”, Physics Reports, arXiv:1308.6604, 2013
[4] Santo Fortunato, “Community detection in graphs ”, Physics Reports, vol. 486, 2010
[5] Newman, M.E.J., Girvan, M., "Finding and evaluating community structure in networks", Phys. Rev. E, 2004
[6] Clauset, A., Newman, M.E.J., Moore, C., "Finding community structure in very large networks", Phys. Rev. E, 2004