Comparative Analysis of Community Detection Algorithms

Pankaj Chejara, W. Wilfred Godfrey
Department of ICT
ABV-Indian Institute of Information Technology and Management
Gwalior, India
pankajchejara23@gmail.com, godwil@gmail.com

***Abstract*—Rapid growth in data has caused a sudden surge of interest among researchers to study the network structure for community detection. In this paper, we have provided a comparative analysis of community detection algorithms for complex networks. In comparison with earlier work on community detection, our work presents the analysis on real network data instead of using synthetic data. This analysis has been performed in two phases. The first phase uses small and medium networks (nodes:10k,edges:20k) and is a selection phase to extract best performing algorithms. The second phase identifies the best community detection algorithm through evaluation on larger complex networks(nodes:100k,edges:1000k).**

***Index Terms*—complex networks, community detection, evaluation, analysis**

I. Introduction

Real life systems are represented as networks with nodes as elementary parts and edges as communication between them [7]. These networks with scale-free properties and enormous size are known as complex networks. Complex networks are networks with millions of vertices and edges. Studying complex networks are essentially of greater utility for understanding complex real systems. Enormous size of complex networks makes it very difficult to comprehend them quickly. Community identification may simplify the process of understanding complex networks by identifying coherent substructures. Researchers across disciplines have proposed numerous community detection techniques [1] [2] [3] [4] [6] [9] [10] [12] [13]. These techniques aid in understanding network structures with lesser efforts. But one recent study has revealed that characteristics at community level are quite different from same at network level [9]. Despite the contrarian views, it is certain that network structure plays a vital role in understanding the properties of a network. A community in a social network groups a set of nodes which are similar to each other. For instance, in a WWW network, partitions, similar to communities are defined on the basis of content presented in web pages, which divides the network into groups which have similar content in each group while being different from other groups.

Community detection algorithms resemble graph partitioning methods. Graph partitioning problem deals with dividing a graph into approximately equal sized c clusters whereas community detection does not require any priori information regarding number of communities and size of them. Moreover, community detection combines similar nodes in same group whereas graph partitioning only minimizes cut section [3]. In a community, there are more intra-community links compared to inter-community links. The community structure in network can also be expressed in terms of probability. In a network if P_{in} is the probability with which nodes are connected to other nodes in same community and P_{out} is probability of linking nodes with other nodes present in different communities, condition $P_{in} > P_{out}$ implies an existence of communities otherwise partitioned network will not be better than a random graph with no significant community structure [8]. Modularity(Q) [10] is used to assess the quality of detected communities. Modularity of a partitioned network can range from 0 to 1. Higher modularity value indicates better community structure while a modularity value of 0 indicates that the detected communities are not better than the same in a random network. Modularity value greater than 0.3 signifies an existence of close-knit community in network [4]. In this paper, we have considered modularity and execution time as evaluation parameters for community detection algorithms.

Community detection techniques broadly can be divided into agglomerative methods, divisive methods and optimization methods [6]. Agglomerative method perform merging of nodes on the basis of their similarity to each other. To the contrary, divisive approach removes links between communities recursively. Optimization methods try to maximize or minimize an objective function while finding community pattern in network.

Rest of the paper is organized as follows. Section 2 provides an introduction to community finding algorithms chosen for analysis. In section 3, we provide a detailed description of modularity parameter. In section 4, we present a brief summary of datasets used for analysis and also provide details of experimental setup. In section 5, we have analyzed obtained results of this study. Finally, section 6 presents conclusion of comparative study undertaken in this paper.

II. Community Detection Algorithms

Numerous community detection methods have been proposed in past few years. These methods allow researchers to reveal communities in networks which can be used in wide

978-1-5386-1866-0/17/$31.00

range of applications ie. recommendation system, innovation diffusion, viral marketing etc. S. Emminos et al. [21] provided comparison of Louvain, Infomap, Label Propagation and Smart Local Moving algorithms using modularity and information recovery metrics. Our analysis has not considered information recovery metrics due to non-availability of ground truth reality of communities for some of datasets used in this paper. In this paper, we have used Python's igraph [20] library to compare community detection algorithms. This library provides mostly used community detection algorithms ie. Newman2006, Infomap, Louvain, Fast greedy, Label propagation, Spin-glass and Random-walktrap algorithms.

A. Newman2006

Newman utilized the concept of spectral partitioning [1]. Leading eigen vectors of modularity matrix are calculated and then network is partitioned into two sub-networks which maximize modularity. In further subdivision, modularity contribution is calculated at each step and this process is stopped when this contribution becomes negative.

B. Infomap community detection

This algorithm was proposed by Martin Rosvall et al. [14]. It uses map equation to find community structure in network. Map equation represents description length of a random walker in a network. Partitions with good modular structure tend to have smaller description length. This is used as an objective function to find better partitions of a network. If a random walker stays longer in a region then it's description length can be compressed. Hence, partitions with better community structure will have minimum description length. This method is based on agglomeration of nodes. It begins with by considering each node as a separate module. Then, randomly selected nodes are combined together resulting in largest decrease in map equation. In subsequent steps, modules formed in previous steps are considered as nodes and the same process is repeated. This process stops when there is no further decrease in map equation.

C. Louvain community detection

This algorithm was proposed by Blondel et al. [6]. This algorithm works in multiple passes. It utilizes modularity parameter as the stopping criteria. This process stops when there is no change in modularity value. In first phase, local maxima of modularity is discovered. Each node i is considered as belonging to unique community. Adjacent nodes whose merging results in higher modularity gain are combined in same group. Once local maxima is achieved, next phase starts. In next phase, communities are treated as nodes while total of weights of inter-communities edges are taken as weight assigned to edges among new nodes. Again same process is repeated on this newly formed network. Results have shown significant improvement in terms of computational speed as compared to others.

D. Fast Greedy community detection

This algorithm was proposed by Clauset et al. [9]. It mainly focuses on networks which have sparse adjacency matrix. This method utilizes efficient data structures to speed-up community detection. It starts with considering each node as a community and maintains ΔQ (change in modularity) for each pair of communities. A maxheap is maintained which stores largest ΔQ with information of community pairs. In every step communities are combined which results in higher modularity gain. This process stops when there remains a single community. Fast greedy community detection executes in $O(md\ \log n)$ time where d is depth of dendrogram.

E. Label propagation community detection

Raghavan et al. [3] presented a near linear time algorithm to detect communities in complex networks. This method is of greater interest for a researcher due to its near linear time complexity. It works as each node is assigned a unique label in the begining and then in every step node gets a label which is owned by most of its neighbors. In case of ties randomly selected label among neighbors is used. As a stop condition strong community [11] measure is used.

F. Spin-glass community detection

This algorithm was proposed by J Reichardt [13]. This algorithm considers spin state of nodes as communities and try to minimize the spin energy. This method works on the concept that nodes with same spin state should be connected and with different spin state should be disconnected. The major objective of this method is to find ground state of spin-glass model.

G. Random-walk community detection

In [12] author used random-walk concept to find community in a network. This method is based on the principle that random walks tend to be confined to denser region of a network (ie. communities). Random walker starts from a non-clustered area and calculates distance between adjacent nodes. Two adjacent communities are chosen and merged into one. Then, distance between communities are updated. This process is repeated (N-1) times.

III. EVALUATION METRICS

In this paper, we have used modularity [10] and execution time as the evaluation factors for community detection algorithms. Modularity measures goodness of partitions of a network by capturing differences between partitions produced by community detection algorithms and partitions of a random network.

Adjacency matrix A stores elements in 0 or 1 form. If A_{ij} value is 1 then it means there is an edge between node i and node j.

$$A_{ij} = \begin{cases} 1 & \text{if node i and node j are connected,} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

TABLE I
COMPARATIVE ANALYSIS OF COMMUNITY DETECTION ALGORITHMS

Dataset		newman2006	infomap	louvain	fast greedy	spin glass	random walk	label prop
Karate	Q	0.393	0.402	0.419	0.380	0.419	0.353	0.402
	T	0.004	0.007	0.0001	0.0001	.467	0.0002	6.389
Dolphin	Q	0.491	0.528	0.518	0.495	0.528	0.489	0.486
	T	.009	0.011	.0002	.0002	0.624	0.0004	0.0001
Polbooks	Q	0.467	0.523	0.520	0.502	0.526	0.507	0.495
	T	0.012	0.027	0.0004	0.0006	1.67	0.0011	0.0002
Netscience	Q	0.952	0.929	0.959	0.955	–	0.956	0.908
	T	0.15	0.394	0.0069	0.007	–	0.0233	0.0064
Facebook	Q	0.799	0.809	0.834	0.774	0.833	0.811	0.814
	T	1.79	5.02	0.1	1.53	563.5	1.96	0.0814
Powergrid	Q	0.825	0.815	0.936	0.933	0.920	0.831	0.804
	T	4.256	7.63	.051	.0168	147.71	0.215	0.375
HiEnCo	Q	0.756	0.768	0.848	0.812	–	0.755	0.771
	T	5.666	6.42	.0486	0.216	–	0.977	0.855
Cond-2003	Q	0.343	0.674	0.760	0.678	–	0.646	0.659
	T	2.95	202.87	0.3228	30.15	–	42.19	23.35

Modularity presents goodness score for partitions of a network. This score is calculated by finding the difference between fraction of edges inside a community and the same in a random network. Fraction of edges inside a community is computed as follows

$$= \frac{\sum_{u,v} A_{u,v}\delta(c_u, c_v)}{\sum_{u,v} A_{u,v}} \tag{2}$$

Function $\delta(c_u, c_v)$ considers only edges whose both vertices are grouped in same community. Here, c_u represents community of node u and c_v represents community of node v. Value of $\delta(c_u, c_v)$ is 1 if c_u equals to c_v and 0 otherwise. Denominator of equation (2) counts each edge twice and hence the total number of edges is given by,

$$m = \frac{1}{2}\sum_{u,v} A_{u,v} \tag{3}$$

So, equation (2) can be rewritten as

$$= \frac{1}{2m}\sum_{u,v} A_{u,v}\delta(c_u, c_v) \tag{4}$$

For trivial cases in which the entire network is considered as a single community (equation 4) achieves highest fraction of 1. Therefore, in order to avoid trivial cases the expected fraction of edges are subtracted. If k_v is degree of node v and k_w is degree of node w. Then the expected number of edges between node v and w would be

$$= \frac{k_v k_w}{2m} \tag{5}$$

Now, modularity can be written as

$$Q = \frac{1}{2m}(\sum_{u,v} A_{u,v} - \frac{k_v k_w}{2m})\delta(c_u, c_v) \tag{6}$$

If a community structure is not better than a random network then modularity value is 0. Network with modularity 0.3 are expected to have a strong community structure. The algorithms are also compared for their efficieny with respect to experimental running times.

IV. DATA SETS & EXPERIMENTAL SETUP

In this paper we have used two kinds of data sets, medium and large. Medium data sets (Table II) are used for comparing performance of community finding methods discussed in this paper and to select the best performing methods (Table III). The medium datasets are karate [15], dolphin [18], polbooks, netscience [9], facebook [5], powergrid [17], hiEnCo [16] and Cond-2003 [16]. The large datasets are complex networks from Stanford datasets [5].

TABLE II
DATA SETS FOR ANALYSIS

Dataset	nodes	edges
Karate	34	78
Dolphin	62	159
Facebook	4039	88234
Powergrid	4941	6594
Polbooks	105	441
HiEnCo	8361	15751
Cond-2003	31163	120029
Netscience	1589	2742

A. Datasets

Karate data set is a friendship network of 34 karate club's students at a US University. This club is split into two groups as a consequence of spat between their group's leader. Dolphin data set is association network among 64 dolphins living in community. Polbooks contains books of politics published during 2004 and an edge between them represent co-purchase made by buyer on Amazon. Netscience dataset provides co-authorship network of scientists working in the area of network theory. Facebook dataset is social network obtained from facebook social site through survey. All data replaced by anonymous data to protect identity of users. Powergrid network data is topological representation of western state power grids of united states. HiEnCo is a weighted network of co-authorship of scientist posting under HIgh Energy Theory Archive. Cond-2003 data set is updated co-authorship network

of scientist posted under Condensed Matter E-Print Archive between 1995 and 2003.

TABLE III
COMPLEX NETWORK DATA SETS FOR ANALYSIS

Dataset	nodes	edges	communities
Youtube	1,134,890	2,987,624	8,385
DBLP	317,080	1,049,866	13,477
Amazon	334,863	925,872	75,149

Youtube dataset contains user's group formed over youtube (video sharing website) by users. This data set has been provided by [19]. Amazon dataset is collected by Amazon website. Products which are bought together represent a link in this dataset. For each product category, links connecting product of that category are considered as single community. DBLP is a co-citation network. Nodes in this network represent authors and link between two nodes represents that corresponding authors have coauthored a paper together.

B. Experimental Setup

We have used igraph python library for community detection algorithms and these algorithms are executed on system with Core i7 processor with 4GB RAM.

V. RESULTS

In our analysis, we have found that in first phase louvain, newman2006, label propagation and fast greedy algorithms performed better than others in terms of modularity and execution time. Although label propagation method offered an efficient solution to community detection but poorly performed on cond-2003 dataset in terms of execution time. However, spin-glass community detection method has performed similar to these algorithms in terms of modularity but running time of spin-glass method is much higher. Spin-glass method failed to perform over netscience, hiEnCo and cond-2003 datasets because it requires fully connected graph in order to work.

In case of complex network, louvain outperformed newman2006 and fast greedy algorithms. Newman2006 community detection method produced community structure with very low modularity value. Fast greedy algorithm finds out community with higher modularity value as compared to newman2006 but the execution time was significanlty higher.

TABLE IV
COMPARATIVE ANALYSIS OF COMMUNITY DETECTION ALGORITHMS ON COMPLEX NETWORKS

Dataset		newman2006	louvain	fast greedy
Youtube	Q	0.0	0.685	-
	T	37.33	14.14	-
Amazon	Q	0.0	0.925	0.876
	T	13.35	4.68	736.02
DBLP	Q	0.0241	0.809	0.735
	T	10.01	4.38	2596.2

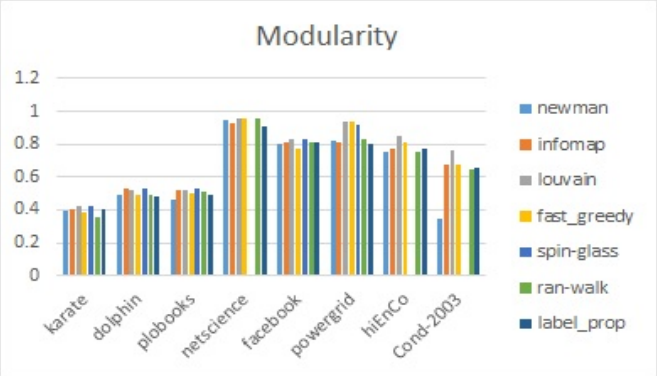

Fig. 1. Comparison of modularity for various algorithms

VI. CONCLUSION

The analysis presented in this paper has shown that louvain community detection algorithm has outperformed other community finding methods undertaken. This method took least time on complex networks. However, this analysis has also revealed that there are community detection methods i.e spin glass, fast greedy, infomap worked approximately as good as louvain but are computationally expensive. First phase of analysis allowed us to select computationally efficient and high modular structure producing community method (newman2006, louvain, fast greedy) for further analysis. Presented results have shown that louvain community detection method has performed best among all community detection algorithms for both phases.

REFERENCES

[1] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices. Physical review E 74.3, 066133, 2006.
[2] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. Plos one 6.4, e18209, 2011.
[3] U.N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. Physical review E 76.3, 036106, 2007.
[4] M. E. J. Newman, Fast algorithm for detecting community structure in networks. Physical review E 69.6, 066133, 2004.
[5] J. Leskovec, A. Krevl, SNAP datasets : Stanford large network dataset collection, https://snap.stanford.edu/data, June 2014.
[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre. Fast unfolding of communities in large networks, Journal of statistical mechanics: theory and experiment 2008. 10, 2008: P10008.
[7] A. Lancichinetti, S. Fortunato. Community detection algorithms: a comparative analysis. Physical Review E 80, 056117, 2009.
[8] A. Condon, R. M. Karp, Random struct. Algor. 18, 116, 2001.
[9] A. Clauset, M.E.J. Newman and C. Moore. "Finding community structure in very large networks." Phys. Rev. E 70, 066111, 2004.
[10] M. E. J. Newman. Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113, 2004.
[11] F. Fadichi, C. Castellano, F. Cecconi, V. Loreto and D. Parisi. Proceedings of national academy of sciences 101, 2658, 2004.
[12] P. Pons, M. Latapy. "Computing Communities in Large Networks Using Random Walks." In: Yolum ., Gngr T., Grgen F., zturan C. (eds) Computer and Information Sciences - ISCIS 2005. ISCIS 2005. Lecture Notes in Computer Science, vol 3733. Springer, Berlin, Heidelberg
[13] J. Reichardt and S. Bornholdt. "Statistical mechanics of community detection." Phys, Rev. E 74, 016110, 2006.
[14] M. Rosvall and C. T. Bergstrom. "Maps of information flow reveal community structure in complex networks." PNAS 105, 1118, 2008.

[15] W. W. Zachary. "An information flow model for conflict and fission in small groups." Journal of Anthropological Research 33, 1977, pp. 452-473.
[16] M. E. J. Newman, Proc. Natl. Acad. Sci. USA 98, 2001, pp. 404-409.
[17] D. J. Watts and S. H. Strogatz, Nature 393, 1998, pp. 440-442.
[18] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behavioral Ecology and Sociobiology 54, 2003, pp. 396-405.
[19] A. Mislove and M. Marcon and K. P. Gummadi and P. Druschel and B. Bhattacharjee. "Measurement and Analysis of Online Social Networks." Proceedings of the 5th ACM/Usenix Internet Measurement Conference, 2007.
[20] Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. http://igraph.org
[21] S. Emmons, S. Kobourov, M. Gallant, and K. Brner, "Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale." Ed. Constantine Dovrolis. PLoS ONE 11.7 (2016): e0159161. PMC. Web. 27 Aug. 2017.