Community Detection Algorithms: Comparative testing on Benchmark Graphs

Syeda Javeria Ashraf
Data Science
NED University of Engineering and Technology
Karachi, Pakistan
ashraf.pg3401068@cloud.neduet.edu.pk

Adnan
Data Science
NED University of Engineering and Technology
Karachi, Pakistan
muhammadadnanakmal@gmail.com

***Abstract*— Community detection involves identifying densely connected nodes which are to a certain degree less connected to other nodes of the network. Communities exhibits Complex scenarios in a graph, elements are nodes and interaction between them are links, and more interaction are found between a group of nodes as compared to others[1][2]**

There are many algorithms to detect communities in a network, but the real problem was to test their performance on such network where we can find their limits. We will be examining the algorithms thoroughly under indistinguishable scenarios so that we can recognize the limitations of that approaches. Usually, community detection algorithms are tested on artificial networks or small networks which fail to exhibit real world scenarios. Generally, the GN benchmark was used to test algorithms because of its reasonable size, we can now analyze graphs with millions of nodes but comparing their performances on a smaller graph might not be appropriate, hence we should check the performance of an algorithm in various size and average degree, to avoid inaccurate outcome [3]. To prevail over the limitation of GN benchmark another algorithm was introduced Lancichinetti, Fortunato and Radicch LFR Benchmark – it is closer to real world (the community sizes also obey a power law distribution) [4][5] This could produce large size of networks with communities of different sizes, they also possess different properties such as average degree and degree distribution [4]. Keywords—Community Artificial networks, Benchmark graphs

I. Introduction

We are surrounded by extremely complicated and complex systems. There are millions of objects that are interconnected or have different type of connections. To focus on the significant organization of complex system we use communities [4]. In a network, communities are found when we find more links of nodes within them (locally), and those nodes are less connected to other nodes. The main purpose of community detection in a network is to find hidden structures which are present in a network. Communities are generally a graph of several subgraphs, but they don't have similar properties, but they are mode connected with each other. Here Density refers to the greater number of connection of nodes within the community as compared to the nodes of other community's nodes. A Strong community is a community which has more links with local nodes as compared to links with nodes of other communities Kint(C)>Kext(C) .Whereas a weak community is when its total internal degree(local) are less as compared to total external degree[6] Kint(C)<Kext(C). To detect community in a network there are so many algorithms available which even detect hierarchies inside communities. Despite of having many algorithms present the question still lies which one is the best to rely on, but the answer is quite tricky to give as we should be testing them on real world network scenarios which are not available. We will be using some of the algorithms and testing them using different Benchmark graphs [7]. The Girvan and Newman benchmark was most used because of its simpler structure and algorithms performed very well on them. It was used because usually research were centered on the straightforward cases with undirected and unweighted graphs. Yet real world graphs are directed and weighted graphs GN benchmark is widely used as the benchmark graph to test algorithms, but it had limitations which made it different from real world network. It had same degree of all nodes and the communities were also of same size. To overcome this limitation LFR benchmark was built, these graphs can be built in no time and provides a much harder test to algorithms and the limits of algorithms are disclosed easily[4] also it provides linear model that's why tests can be conducted on larger size of systems as the algorithm will analyze them rapidly. The goodness of community detection algorithms will be found through the LFR Benchmark

Properties common in real world network are that they follow power law distribution, scale free and small world, non-zero correlation and high transitivity [7][8]. Real world communities can also be made not by only the properties stated above but by also some other characteristics that includes community wise density and average distance [9].

We will be passing all algorithms on LFR benchmark and then some weighted and directed graphs will also be used. Through this we will jump into conclusion which algorithm is performing how and on what circumstances.

II. Algorithms

To analyze the algorithms, we have selected 6 Community Detection Algorithms which will be discussed, and tests will be conducted on different graphs. We will be discussing the specifications of each algorithm here.

A. Fast Greedy Modularity Optimization by Claus

A set of isolated nodes are picked. The links are iteratively added till the largest modularity is achieved. The end is achieved when modularity can't be increased any further. [10].

B. Walktrap Algorithm

Random walk principle is used. Clusters are formed by calculating distance between nodes. The clusters are then combined to bottom up [11].

C. Multilevel Algorithm

This technique involves agglomerative method. Here communities are built upon starting from single node and then combining clusters in every iteration [12].

D. Infomap Algorithm

Here again random walk process is used and clustering is done. Communities are detected by compacting the random walker movement on a network [14].

E. Louvian Algorithm

This works in two steps. Local changes optimize the modularity. A random node is selected, and modularity is calculated. Largest modularity nodes are connected to each other, and clusters are formed. Now in second part, communities are again clustered and nodes belonging to same community are joint in a single node [13].

F. Label Propagation

This creates community clusters just by network. Not other prior information is needed [15].

III. Results on LFR Benchmark

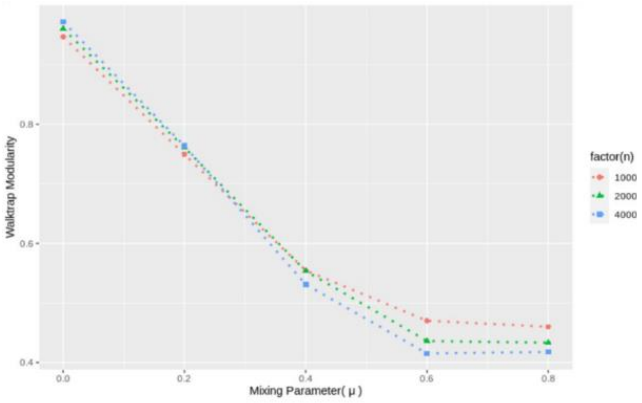

Fig. 1. The following is the representation of the results on Walktrap Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Walktrap Algorithm. Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here size of network has a little bit impact, when the size of network increases modularity decreases.

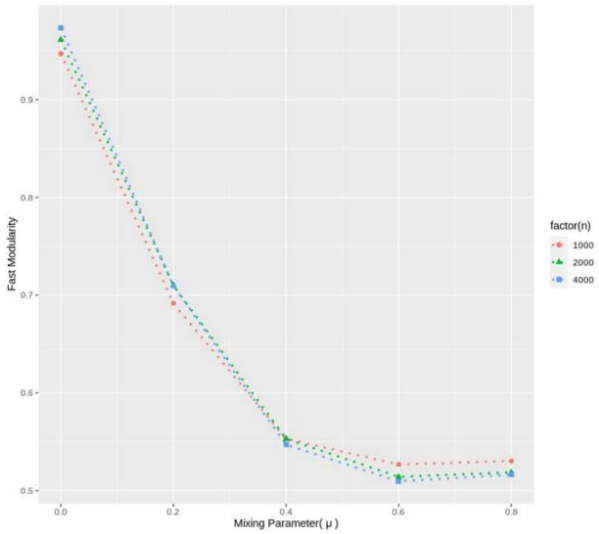

Fig. 2. The following is the representation of the results on Fast Greedy Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Fast Greedy Algorithm. Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here one more thing we can focus on is that when the size of network is relatively small the modularity is high but after some limit size of network doesn't impact on modularity.

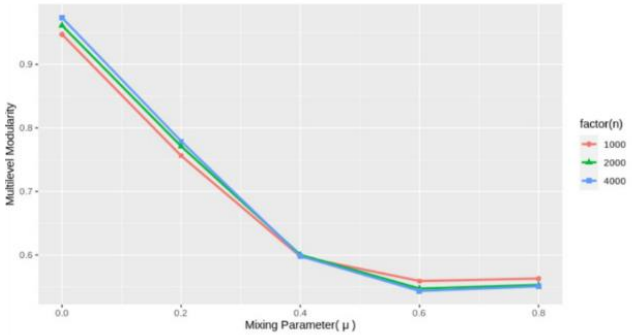

Fig. 3. The following is the representation of the results on Multilevel Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Multilevel Algorithm. Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here one more thing we can focus on is that size of network doesn't impact on modularity.

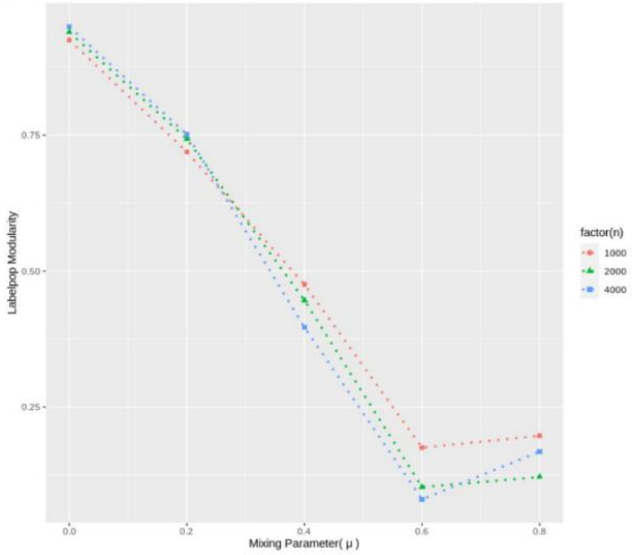

Fig. 4. The following is the representation of the results on Label Propagation Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Label Propagation Algorithm.
Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here the size of network is behaving little differently. When the size of network is small modularity is high. When size of network is on avg 2000 the modularity is quite low. And when the size of network increases more, the modularity is again restored.

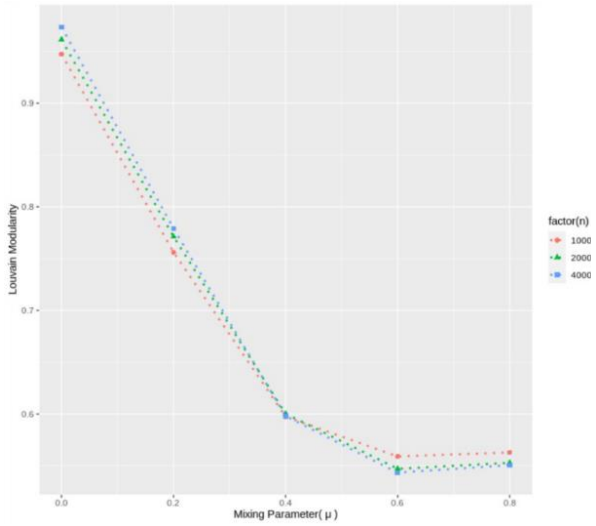

Fig. 5. The following is the representation of the results on Louvian Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Louvian Algorithm. Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here one more thing we can focus on is that number of network doesn't impact on modularity.

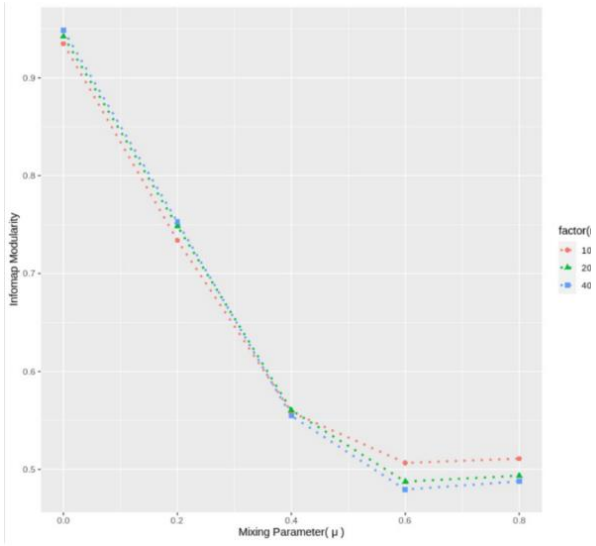

Fig. 6. The following is the representation of the results on Infomap Algorithm. Here on x-axis, we have taken Mixing Parameter (μ) and on y-axis we have Modularity of Infomap Algorithm. Here we are seeing that when Mixing parameter is small the modularity is high and gradually decreases as mixing parameter is increased. Here one more thing we can focus on is that when the size of network is relatively small the modularity is high but after some limit size of network doesn't impact on modularity. This mimics the behavior of Fast Greedy Algorithm.

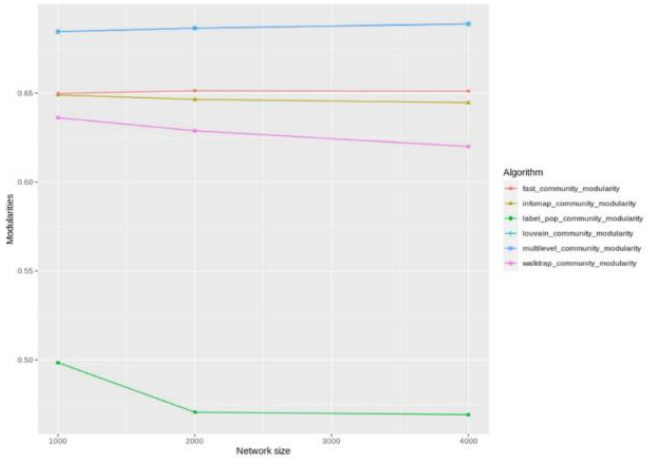

Fig. 7. Here in this figure we have compared the modularities of all the 6 algorithms. The Multilevel Algorithm has captured the best modularity and Label propagation has captured the worst modularity.

IV. Conclusion

In this study we carried forward the work which were previously done. We have answered the question: How do community detection algorithm compare in terms of different parameters and different graph sizes? We have implemented multiple algorithms for community detection on LFR benchmark and we have studied the effect of modularity on different algorithms. We have used multiple sizes of networks for this study, and we have concluded that the modularity decreases as the mixing parameter increases. The size of graph has impact on modularity that behaves differently for different algorithms. We have also compared the modularity of all algorithms and found out that Multilevel Algorithm performs the best and has captured the best modularity.

Acknowledgment

We would like to thank Dr. Qasim Pasta for his constant support on this Research project. This would not be possible without his guidance.

References

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep. 424, 175 (2006).

[2] M. E. J. Newman, SIAM Rev. 45, 167 (2003).

[3] M. Girvan and M. E. Newman, Proc. Natl. Acad. Sci. U.S.A. 99, 7821 (2002).

[4] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi, "Benchmark graphs for testing community detection algorithms", Phys. Rev. E 78, 046110 (2008)

[5] Andrea Lancichinetti and S. Fortunato, Phys. Rev. E 80, 016118 (2009).

[6] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. PNAS, 101:2658–2663, (2004).

[7] Newman, M.E.J.: The Structure and Function of Complex Networks. SIAM Rev. 45, 167– 256 (2003).

[8] Guimerà, R., Danon, L., Díaz-Guilera, A., Giralt, F., Arenas, A.: Self-Similar Community Structure in a Network of Human Interactions. Phys. Rev. E 68, 65103 (2003)

[9] Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical Properties of Community Structure in Large Social and Information Networks. In: WWW, ACM, Beijing (2008)

[10] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E 70, 066111 (2004).

[11] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In Computer and Information Sciences-ISCIS 2005, pages 284–293. Springer, (2005).

[12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. P10008,(2008).

[13] M.E.J. Newman. Fast algorithm for detecting community structure in networks. Physical Review E, 69:066133, (2004).

[14] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, (2008).

[15] Usha Nandini Raghavan, R´eka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 76 (3):036106, (2007).