FOSSEE INTERNSHIP RESEARCH REPORT

Comparative Analysis of Community Detection Algorithms on Real, Artificial and Random Datasets

Tanisha Mandal
19BAI1116

Abstract

Community detection in a network is the process of identifying and grouping together the more densely interconnected nodes in a given graph. This graph can take the form of a social network graph, a biological network, or a representation of a local network of computers, for example.

The contribution of this paper includes the visualization of the community detection process taking place by means of eight different algorithms. The algorithms are tested on a real-world dataset and an artificially generated dataset, and the results compared and analyzed. The artificial network is generated and tested as a benchmark to evaluate the performance of all the algorithms used.

Introduction

Graphs

A graph is a nonlinear data structure used to model pairwise relations between objects. A graph comprises nodes or vertices which are connected by edges or links.

An undirected graph consists of symmetric links, whereas a directed graph includes asymmetric, 'to and from' links between vertices. This is to say that one node acts as an origin and the other as the destination for the link. Undirected links can be traversed along any direction, whereas directed links can only be traversed in the direction of the link, that is, from the origin to the destination node.

A weighted graph is a graph in which links are assigned certain numeric values known as weights. These weights may represent costs, lengths or other quantities depending on the graph. Graphs can be used to model many types of relations and processes in scientific information systems. Graphs can be used to model and solve many practical problems.

Network theory and graphs

Graphs are thus versatile tools to represent various models, and in particular, networks. Network theory is the study of graphs as a representation of either symmetric relations or asymmetric relations between discrete objects. A network can be defined as a graph in which nodes and links have attributes like names, and where the optional weight attribute represents a real-world quantity. The analysis of networks lends itself to various scientific fields like biology, physics, statistics, computer science, sociology and statistics, to derive meaningful inferences from a data store.

Social network analysis in particular has emerged as a prevalent application of community detection algorithms. With the rise of the world wide web and social networking sites, characterizing consequent communities that arise yields a wealth of information about human behavioral patterns. Thus, there is a need for algorithms that detect such communities so that further meaningful inferences can be made from them.

Community

A community is defined as a subset of nodes within a graph within which the connections or linkages are denser as compared to the rest of the network. A network is said to have community structure if the nodes of the network can be easily grouped into potentially overlapping sets of nodes, such that each set of nodes is densely connected internally. Communities hold significance in a network because they provide insight into the overall entity that it represents. Taking the example of a biological application, in metabolic networks, communities correspond to cycles or pathways whereas in protein interaction networks, communities correspond to proteins with similar functionality inside a biological cell. In social networks, the existence of communities generally affects various processes like false news or epidemic spreading. In order to completely understand and carry out studies on these processes, community detection techniques are of the utmost importance.

Clustering coefficient as a graph metric

The clustering coefficient of a graph is a measure of the degree to which nodes in a graph tend to cluster together. It is nothing but the probability that adjacent vertices of a vertex are connected. It is sometimes referred to as the transitivity of a graph. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit communities characterised by a relatively high density of links. This likelihood tends to be greater than the average probability of a link randomly established between two nodes. The clustering coefficient has therefore been used on each network that has been tested in this paper as a means of getting an idea about its relative interconnectivity.

Modularity as a CDA metric

Modularity is the measure of the structure of a given network or graph when community detection is carries out. It measures the strength of division of a network into modules (communities). Networks with high modularity have dense connections between the nodes within modules, but sparse connections between nodes in different modules. The concept of modularity is often used to optimize the process of community detection by algorithms on networks.

It is used as a means of evaluating the partition of a network into communities based on the intuition that random graph structures should not follow a community structure.

Let us assume an arbitrary network that is arbitrarily partitioned into n_c communities. It is then possible to define a matrix e of size $n_c \times n_c$ where each element e_{ij} represents the fractions of total links originating from a node in partition i and ending at a node in partition j. Then, the sum of any row (or column) of e, $a_i = P_j e_{ij}$ corresponds to the fraction of links connected to i. If the network is

random or without community structure, the value of the fraction of links can be estimated as the probability that a link begins at a node in i, a_i, multiplied by the fraction of links that end at a node in i, a_i . Thus, the expected number of intra-community links is just $a_i a_i$. We know that the real fraction of links exclusively within a partition is e_{ii}. So, we can compare the two directly and sum over all the partitions in the graph.

$$Q \equiv \sum_i (e_{ii} - a_i^2)$$

This measure is known as modularity and has been used to evaluate the community detection carried out by all the algorithms in this paper.

LFR benchmark for CDA evaluation

When it comes to detecting communities, there is still no universal agreement on what an ideal result of the community detection process looks like, and by extension, the reliability of community detection algorithms. This is where a simple network model, the planted ℓ-partition model comes into play. In this model one 'plants' a partition in a graph, which consists of a certain number of groups of nodes. Each node has a probability p_{in} of being connected to nodes of its group and a probability p_{out} of being connected to nodes of other groups. As long as $p_{in} > p_{out}$ the groups are communities, whereas when $p_{in} \leq p_{out}$ the network is considered to be a random graph.

The most popular version of the planted ℓ-partition model was proposed by Girvan and Newman, called the GN benchmark. This benchmark comprises a graph consists of 128 nodes, each with an expected degree of 16, and which are divided into four groups of 32 nodes each. The GN benchmark can be used to compare the performance of different community detection algorithms against each other. However, the GN benchmark has the drawbacks that all the nodes have the same expected degree and all the communities are of the same size. This is unrealistic because real-world networks show much more complexity in terms of degrees and community sizes.

The LFR benchmark was thus introduced as a new benchmark. It improves upon the GN benchmark by introducing power law distributions of degree and community size. There is also no restriction on the expected degrees or community sizes. LFR benchmark graphs can be generated relatively quickly even for very large networks, thus it has an advantage over the GN benchmark and is a better test for the performance of any given algorithms. In this paper, a network generated by the LFR benchmark has been used for the same.

Networks used:

The choice of datasets has been made to show the varying impact of the input parameters on the output. A regular real-world network, am artificial network used as a benchmark and a random graph have been used to fairly test the working of all the algorithms used.

(i) Zachary's Karate Club network

In 1977 Wayne Zachary collected data about the members of a university karate club. Each node represents a member of the club, and each edge represents a tie between two members of the club. This is a real-world undirected network with 34 nodes (members) and 78 edges. During the course of this study, a fight arose between members of the club which led to it splitting into two groups. An interesting problem that thus arose was figuring out these two groups by examining the nature of the links with the aid of community detection algorithms. Thus, this is an apt network to perform analysis on, and the results would be notable.

(ii) LFR generated network

An artificial network with 150 nodes and 1209 edges was generated following the principles of the LFR benchmark. The input was the number of nodes and certain pre-set parameters and the output was the list of edges and nodes of the artificially generated network. The graph is undirected and weighted in nature. This artificial network was specifically generated to test and compare the performances of all the algorithms against each other, hence the name benchmark. It is a useful tool to catch outlier-like behaviour of algorithms and identify if they are unreliable when compared to alternative ones.

(iii) Random graph

A random undirected graph with 150 nodes and 1000 edges was generated using the igraph package in R. The random network model developed by Erdős-Rényi was used for the same. The function used takes the numbers of nodes and edges as parameters and assigns appropriate links to the nodes on the principle of randomness.

The testing on random graphs would be a helpful indicator as to the measure of the efficacy of the community detection algorithms. This is because the presence of communities within a random graph should be minimum. This tendency should be reflected in the output of the algorithms. This is a step that is often omitted in such analyses of algorithms.

It is intuitive, therefore, to assume that random networks would not exhibit any kind of community structure. However, in practice, random graphs can show the presence of pseudo-communities, i.e., clusters that emerge due to random areas of edge density. That said, a good community detection algorithm should be able to differentiate between a pseudo-community and a community produced by a meaningful network. Hence, this random graph is used to put the eight algorithms used in this paper to the test.

Algorithms used:

(i) Girvan-Newman Algorithm

The Girvan–Newman algorithm was developed by named after Michelle Girvan and Mark Newman in 2002. It is a hierarchical method used to detect communities that functions on the principle of progressively removing edges from the original network. The resulting structures after the deletion

of edges are the communities thus detected. It relies on the concept of edge-betweenness as a measure of the likelihood of a given edge connecting two communities.

The algorithm follows the following steps to detect communities:

1. The edge-betweenness of all edges in the network is calculated.
2. The edge(s) with the highest betweenness are removed.
3. The betweenness of all edges affected by the removal is recalculated.
4. Steps 2 and 3 are repeated until no edges remain.

The end result of this algorithm is a dendrogram of the graph, which is a hierarchical map or a rooted tree of the original graph. The leaves of the tree represent the nodes of the network and the root of the tree represents the whole graph.

(ii) Label Propagation Algorithm

Label propagation is essentially a semi-supervised machine learning algorithm. It assigns labels to unlabelled nodes from the initial state of only a small subset of nodes being labelled. The working of the algorithm entails the propagation of the labels of the initial nodes to the unlabelled nodes. In other words, it works by labelling the vertices with unique labels and then updating the labels by majority voting in the neighbourhood of the vertex. Thus, it can be used for community detection. Label propagation algorithm is known for its fast, nearly linear time run time and lack of prior information needed about a network.

The algorithm follows the following steps to detect the communities:

1. Initialize the labels at all nodes in the network.
2. Set parameter t = 1.
3. Arrange the nodes in the network in a random order and set it to X.
4. For each node chosen in that specific order, return the label occurring with the highest frequency among neighbours. Select a label at random if there are multiple highest frequency labels.
5. If every node has a label that the maximum number of their neighbours have, then stop the algorithm. Else, set t = t + 1 and go to (3).

(iii) Fast Greedy Algorithm

The Fast Greedy algorithm, also known as greedy optimization of modularity, detects communities in graphs by optimizing a modularity score, as its name suggests. This method follows a bottom-up approach to detecting communities. Initially, it treats each individual node in a network as a singleton community. Then it computes the expected improvement of modularity for each pair of communities, chooses a community pair that gives the maximum improvement of modularity and merges them into a new community. The above procedure is repeated until no community pairs merge leads to an increase in modularity.

The algorithm follows the following steps to detect the communities:

1. Each node is given the status of a community.
2. The pairwise modularity score for all the communities is calculated.
3. The pair of communities with the best modularity score is merged into a new community.

4. Steps 2 and 3 are repeated until the modularity score cannot be improved upon.

(iv) Spinglass Algorithm

The Spinglass algorithm carries out community detection by a process known as simulated annealing. Annealing is a heat treatment process in metallurgy that changes the properties of a material to increase ductility and yield other desirable qualities in a material. Similarly, this algorithm makes use of parameters called the 'start temperature', 'stop temperature' and the 'cooling factor' to detect communities. The name of this algorithm is nothing but an analogy between the statistical mechanics of complex networks and physical spin glass models. It can be considered to be rather simplistic in the sense that allows a node to be part of only one community. Thus, there is the risk of the algorithm failing when it comes to networks with overlapping communities.

(v) Walktrap Algorithm

Walktrap algorithm was developed by Peter Pons. It is used to detect communities in networks by sampling random walks. These random walks are then used to compute distances between nodes, which are then grouped based on bottom-up hierarchical clustering. The operant principle used here is that short random walks tend to be members of the same community. Like the Spinglass algorithm, it takes into account only one community per node, which is a potential limitation when applying it to networks with overlapping communities.

The algorithm follows the following steps to detect the communities:

1. Random walks are performed in the network.
2. The distance between nodes is calculated with the help of these random walks.
3. Nodes are grouped together based on distances.
4. Steps 1 to 3 are iterated until the community detection is satisfactory.

(vi) Louvain Algorithm

Also known as the multilevel algorithm, the Louvain algorithm was created by Blondel et al. from the University of Louvain. It maximizes a modularity score for each community during the community detection process. The modularity score is nothing but a quantification of the quality of an assignment of nodes to communities, i.e., the membership of a node to a community. First small communities are found by optimizing modularity on all the nodes in a network, then each small community is grouped into one node and the first step is repeated. Thus the Louvain algorithm is a hierarchical clustering algorithm, that recursively merges communities into a single node on the basis of modularity optimization.

The algorithm follows the following steps to detect the communities:

1. Each vertex is assigned to a community of its own.
2. Vertices are re-assigned to communities greedily: each vertex is moved to the community where it achieves the highest contribution to modularity
3. If no vertices can be reassigned, each community is considered a vertex of its own, and steps 1 and 2 are repeated.

4. Steps 1 through 3 are reiterated until there is only a single vertex left or when the modularity cannot be increased.

(vii) Infomap Algorithm

The Infomap algorithm detects community structures in such a way that minimizes the expected length of a random walker trajectory. In other words, it attempts to minimize a cost function and perform partitioning based on the flow induced by the pattern of connections in a given network.

Suppose a sender of a message pretends to communicate a using a random path inside a network to a receiver, let's assume the size of this message is to be minimized. The consequent strategy would be to attribute to each node a different name (code) and send the receiver the corresponding sequence. If the receiver has a codebook, they can decode the message. Considering the path is described in binary language by N codes of the same size, the minimum length L of each code word is: L = log2 N.

(viii) Leading Eigenvector Algorithm

The leading eigenvector algorithm relies on the concepts of differential math to detect communities. It defines a modularity matrix, B, which is B=A-P, where A is the adjacency matrix of the undirected network, and P contains the probability that certain edges are present in the model. In other words, a P[i,j] element of P is the probability that there is an edge between vertices i and j in a random network. The leading eigenvector method works by calculating the eigenvector of the modularity matrix for the largest positive eigenvalue and then separating vertices into two communities based on the sign of the corresponding element in the eigenvector. If all elements in the eigenvector are of the same sign that means that the network has no underlying community structure.

Implementation:

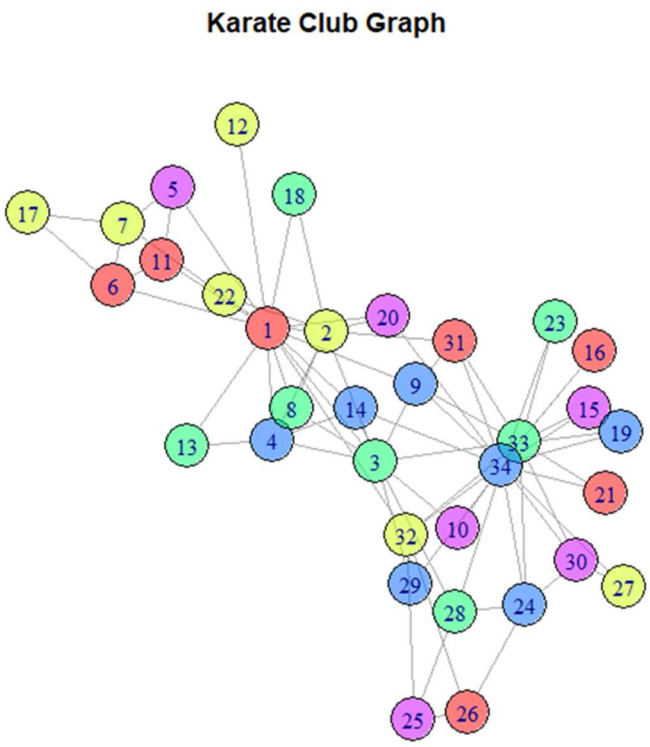
The visualizations of the networks have been carried out in the R programming language in conjunction with its companion IDE RStudio. The LFR benchmark network was generated with code in the C++ language authored by its developers Andrea Lancichinetti and Santo Fortunato.

Visualizations:

(i) Zachary's Karate Club network

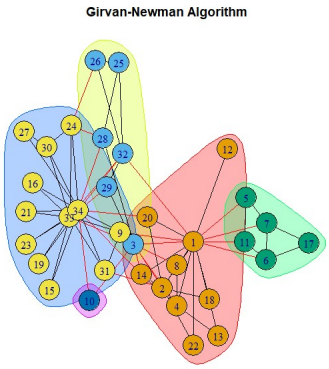
This network was found to have a clustering coefficient of 0.2556818.

The original network before clustering is as follows:

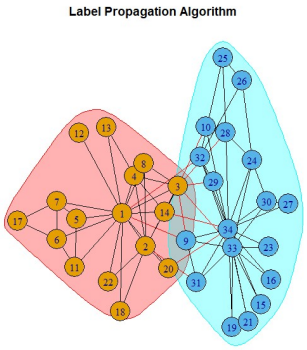


Following are the visualizations of the communities detected by the various algorithms and the measure of modularity and number of communities detected.

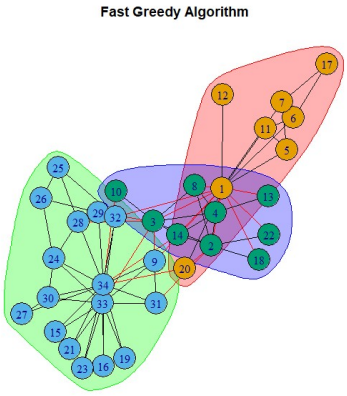
(i) Girvan Newman Algorithm



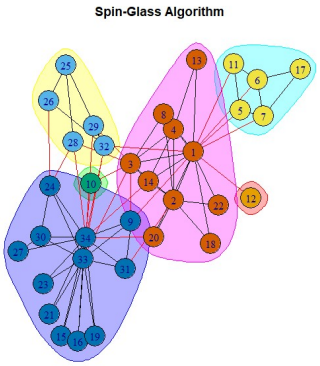
(ii) Label Propagation Algorithm



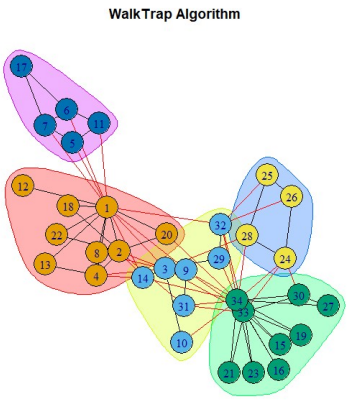
(iii) Fast Greedy Optimization Algorithm



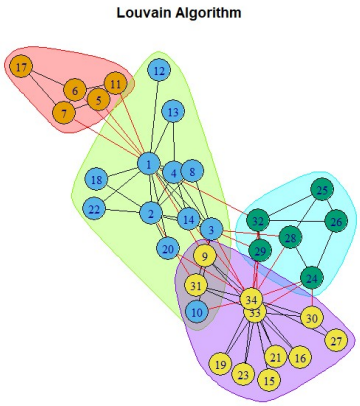
(iv) Spinglass Algorithm



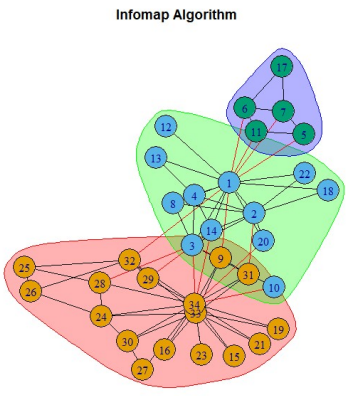
(v) Walktrap Algorithm



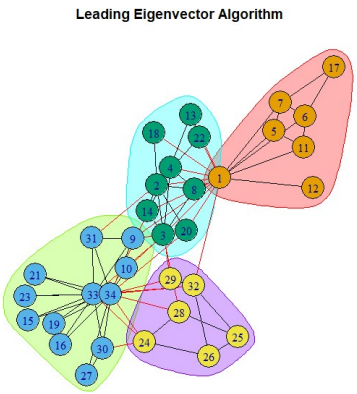
(vi) Louvain Algorithm



(vii) Infomap Algorithm



(viii) Leading Eigenvector Algorithm

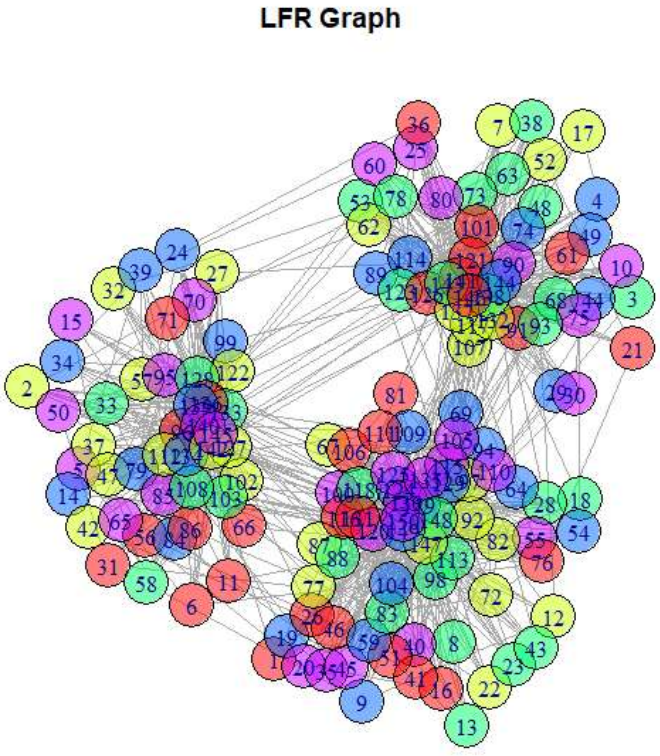


S. No.	Algorithm	Modularity	No. of communities detected
1	Girvan-Newman	0.4012985	5
2	Label Propagation	0.3714661	2
3	Fast-Greedy	0.3806706	3
4	Spinglass	0.4063116	6
5	Walktrap	0.3532216	5
6	Louvain	0.4188034	5
7	Infomap	0.4020381	3
8	Leading Eigenvector	0.3934089	4

(ii) LFR generated network

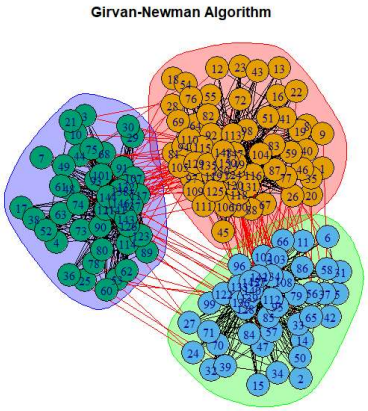
This network was found to have a clustering coefficient of 0.3433802.

The original network before clustering is as follows:

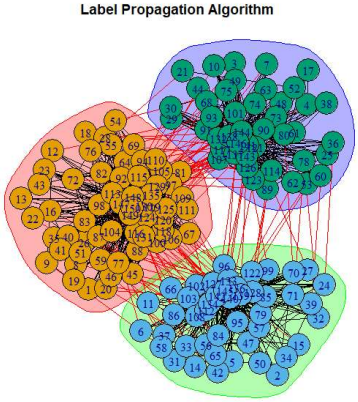


Following are the visualizations of the communities detected by the various algorithms and the measure of modularity and number of communities detected.

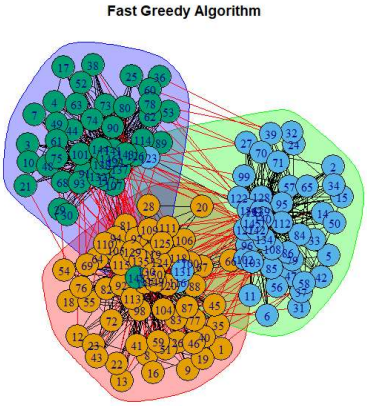
(i) Girvan Newman Algorithm



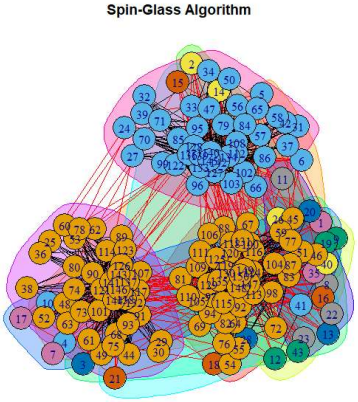
(ii) Label Propagation Algorithm



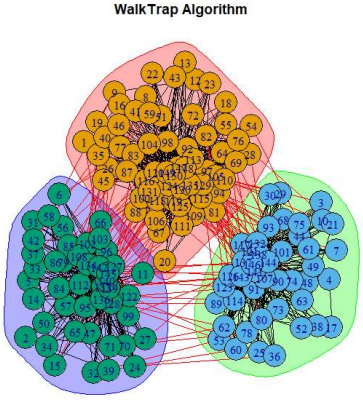
(iii) Fast Greedy Optimization Algorithm



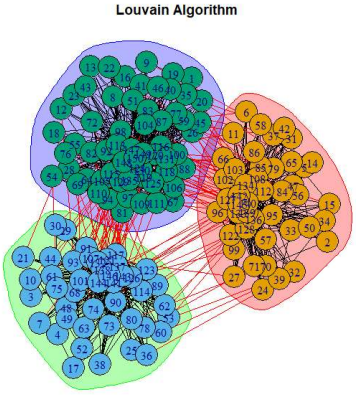
(iv) Spinglass Algorithm



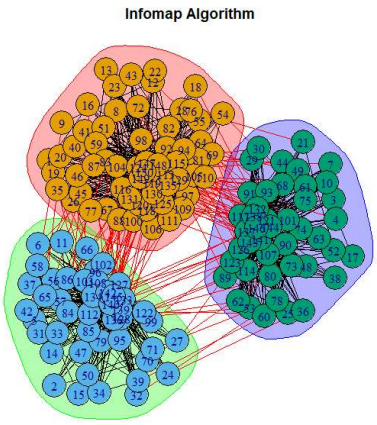
(v) Walktrap Algorithm



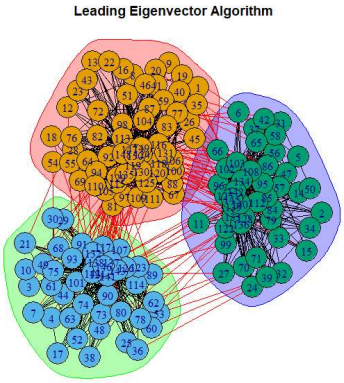
(vi) Louvain Algorithm



(vii) Infomap Algorithm



(viii) Leading Eigenvector Algorithm

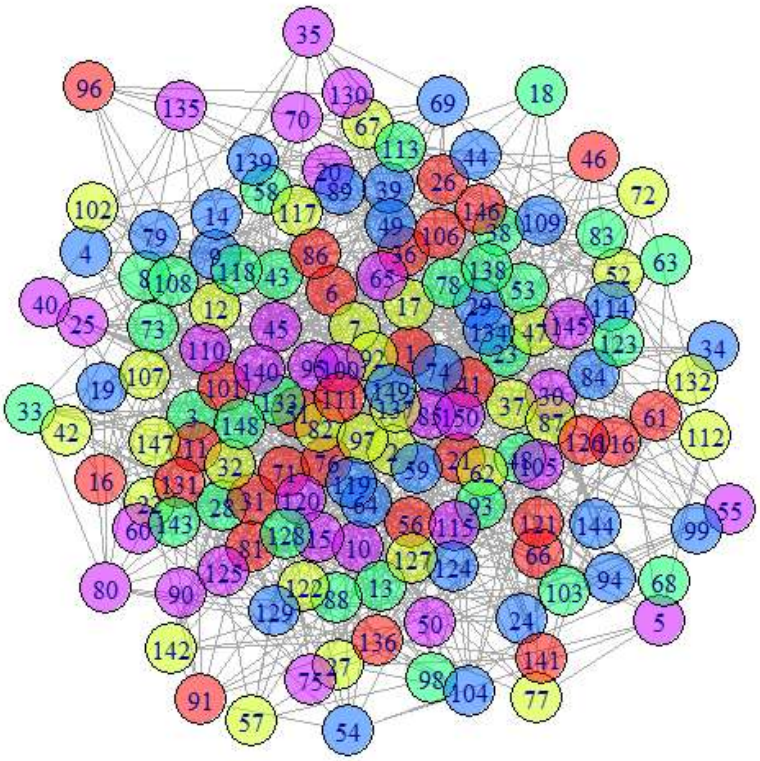


S. No.	*Algorithm*	*Modularity*	*No. of communities detected*
1	Girvan-Newman	0.5509484	3
2	Label Propagation	0.5509484	3
3	Fast-Greedy	0.4580778	3
4	Spinglass	0.1489547	10
5	Walktrap	0.5509484	3
6	Louvain	0.5509484	3
7	Infomap	0.5509484	3
8	Leading Eigenvector	0.5509484	3

(iii) Random graph

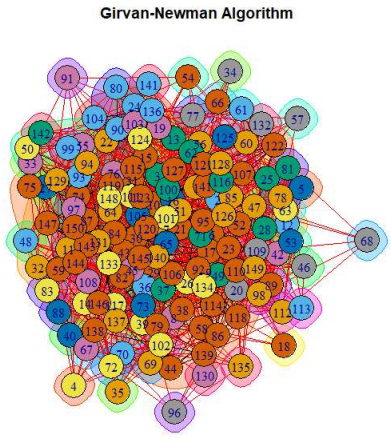
This network was found to have a clustering coefficient of 0.08657351.

The original network before clustering is as follows:

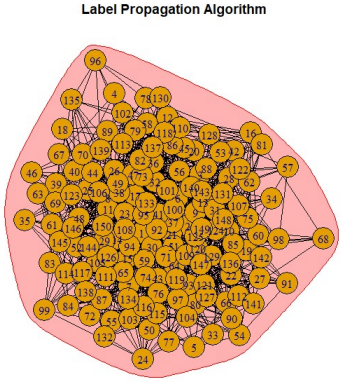


Following are the visualizations of the communities detected by the various algorithms and the measure of modularity and number of communities detected.

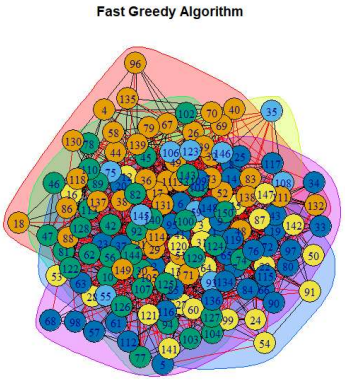
(i) Girvan Newman Algorithm



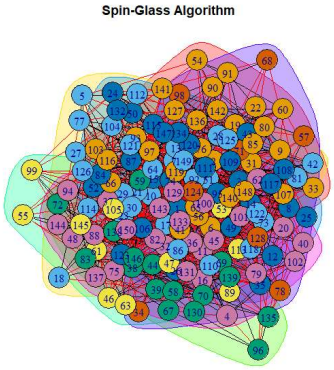
(ii) Label Propagation Algorithm



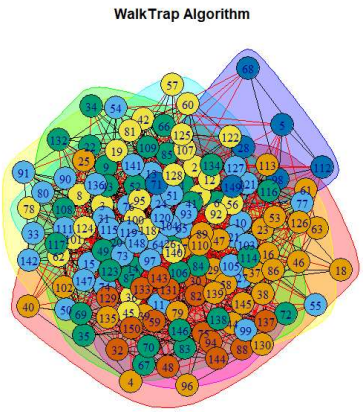
(iii) Fast Greedy Optimization Algorithm



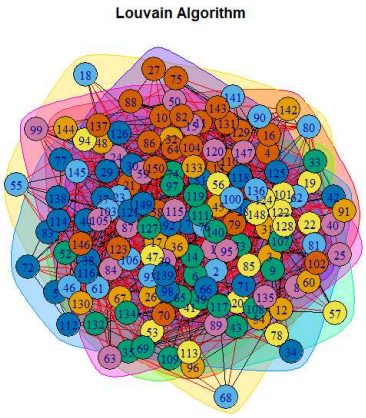
(iv) Spinglass Algorithm



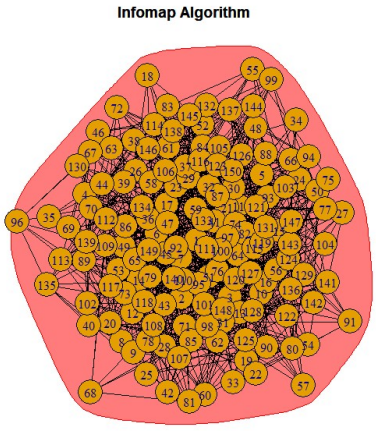
(v) Walktrap Algorithm



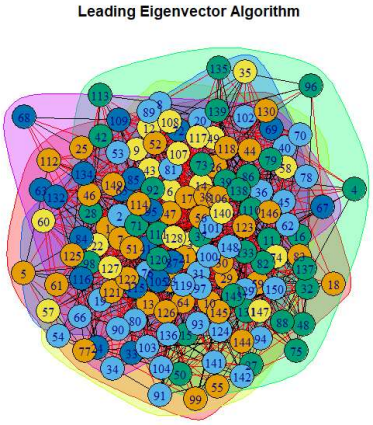
(vi) Louvain Algorithm



(vii) Infomap Algorithm



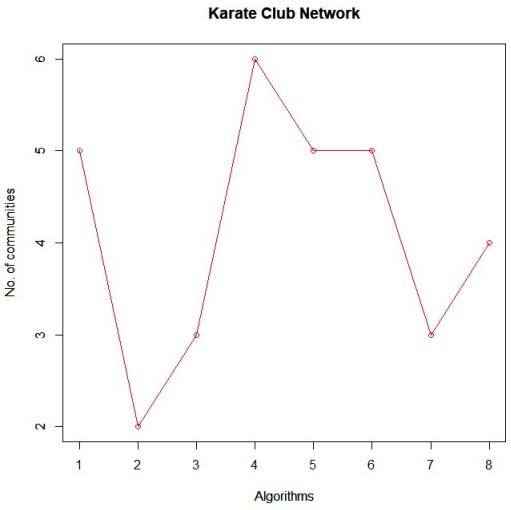
(viii) Leading Eigenvector Algorithm

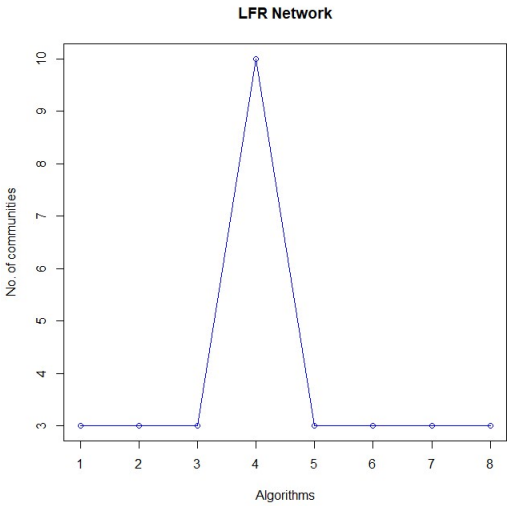


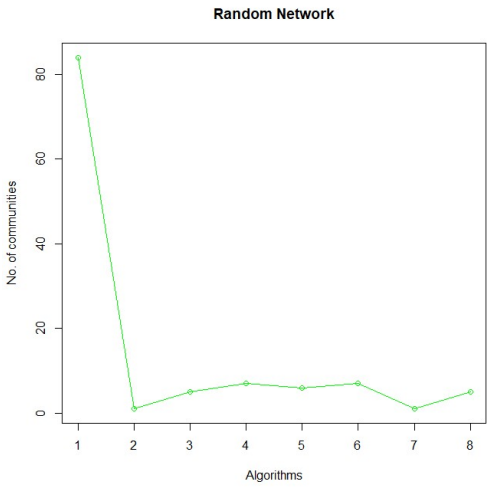
S. No.	Algorithm	Modularity	No. of communities detected
1	Girvan-Newman	0.079879	84
2	Label Propagation	0	1
3	Fast-Greedy	0.208242	5
4	Spinglass	0.244869	7
5	Walktrap	0.190559	6
6	Louvain	0.2194065	7
7	Infomap	0	1
8	Leading Eigenvector	0.2007015	5

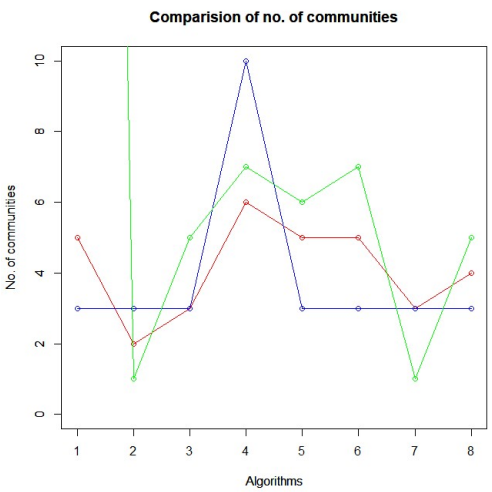
Result:

Comparison of number of communities detected:

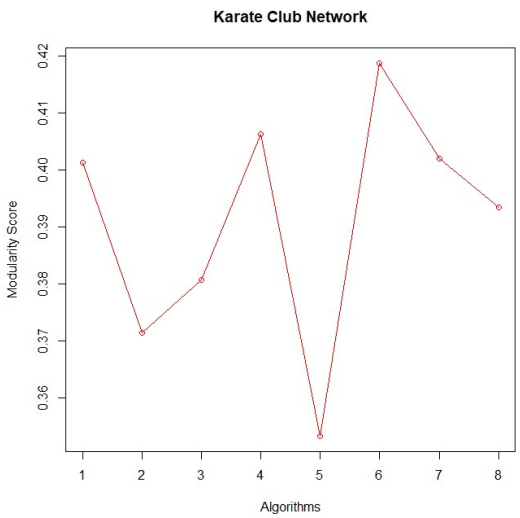


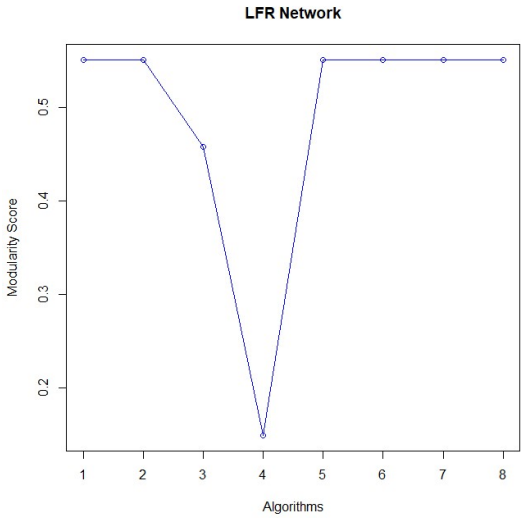


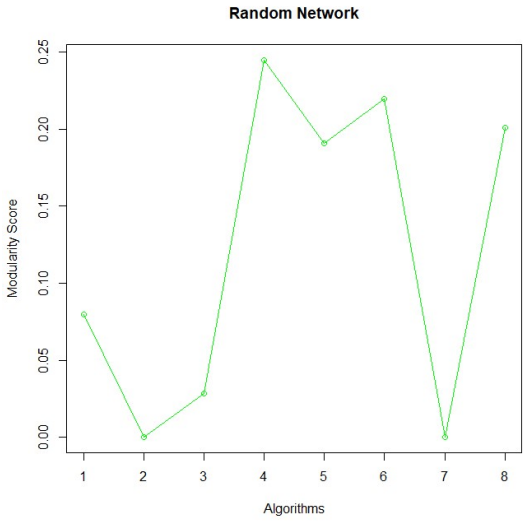




Comparison of modularity scores:







For reference, the algorithms tested as shown were:

1. Girvan-Newman
2. Label Propagation
3. Fast Greedy
4. Spinglass
5. Walktrap
6. Louvain
7. Infomap
8. Leading Eigenvector

Inferences

Plotting the results of the number of communities yielded and the modularity scores of each algorithm, the inferences that can be made are interesting.

Let the number of communities detected in the Karate Club Network be taken as a reference point. The LFR benchmark performs pretty consistently across most algorithms as expected. The common result of 3 communities detected is obtained except for the Spinglass algorithm which detects 10 communities. More communities are detected in the random network than in the other two networks in general. This maybe because of random clusters of densely interconnected nodes existing in the network, that are recognized by the algorithms. It is seen that Label Propagation and Infomap algorithms fail when it comes to the random network, i.e., they view the entire network as a singleton community rather than detecting discrete sub communities within it. Consequently, the modularity score is 0 for these two instances.

Comparing the modularity scores, the LFR benchmark once again produces a consistent result. The modularity score of 0.5509484 is obtained for most algorithms, except the Label Propagation and the Spinglass algorithms. The scores obtained on the LFR benchmark are higher than those on the Karate Club network. The scores of the algorithms with the random network are significantly lower, thus furthering the point that the communities detected on the other networks are meaningful and hold a significance. The Spinglass algorithm, however, is an exception to this. Its modularity score on the random network is higher than that on the LFR benchmark.

Conclusion

Thus, eight community detection algorithms were tested against a real-world network, an artificial network and a random network. The results were noted and analyzed. The artificial network is used as a benchmark for the performance of the algorithms and the random network put to test whether the results of the algorithms is meaningful or not. It was found that most algorithms performed consistently against the benchmark, and their performance was lower on the random network as expected. It is found that the Louvain algorithm performed the best with the most consistent output whereas the Spinglass algorithm produced results that were outliers and unreliable.

References

1. Yang Z., Algesheimer R. & Tessone C. J. (2017). A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *www.nature.com/scientificreports*
2. Jain P., & Tomar D. S. (2019). Review of Community Detection over Social Media: Graph Prospective. *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 2*
3. Lancichinetti A. & Fortunato S. (2010). Community detection algorithms: a comparative analysis. *Complex Networks and Systems, Institute for Scientic Interchange (ISI)*
4. Slota G. M., Berry J., Phillips C. A. & Rajamanickam S. (2018). Scalable Generation of Graphs for Benchmarking HPC Community-Detection Algorithms. *Rensselaer Polytechnic Institute, Troy, NY*; *Sandia National Laboratories, Albuquerque, NM*

5. Danon L., Diaz-Guilera A., Duch J. & Arenas A. (2005). Comparing community structure identification. *Departament de Fisica Fonamental,Universitat de Barcelona, Marti i Franques 108086 Barcelona, Spain*
6. Wagenseller P., Zhao Y., Wang F. & Avram A. (2021). Community-based location inference in social media using supervised learning approach. *https://doi.org/10.1007/s13278-021-00769-5*
7. Orman G. K. & Labatut V. (2009). A Comparison of Community Detection Algorithms on Artificial Networks. *International Conference on Discovery Science, Oct 2009, Porto, Portugal. pp.242-256*
8. Yang J., McAuley J. & Leskovec J. (2014). Community Detection in Networks with Node Attributes. *Stanford University*