International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

# Empirical Performace Determination on Community Detection Techniques in Social Networks

**M. Mohamed Iqbal, K.Latha**

***Abstract:*** ***Community identification is the high common and extending field of interest in social and real-time network applications. In recent years, many community detection methods have been developed. This paper describes various community discovery methods such as InfoMap, Clique Guided, Louvain, Newman and Eigen Vector that have already been developed and also compares the experimental results of those proposed techniques. The proposed work in this paper experiments these community mining algorithms on the two real-world datasets Twitter and DBLP (Computer Science Bibliography) networks. The identified communities by all the community mining algorithms for these two data sets are described in this proposed work. The quality of the derived communities is evaluated by using standard Extended Modularity metric. The experiment results show that the InfoMap algorithm produces a good modularity score than other community mining algorithms for different sizes of communities on both data sets.***

***Keywords:*** ***Community Detection, InfoMap, Real time network, community structure, social network.***

## I. INTRODUCTION

In WWW (World Wide Web) social networking plays a very important role for the past few years in the wide range of applications, due to its ability to allowing social relation on top of the web for topographically distributed users. The web users act with one another, joins in on-line conversation, and swapping totally conflicting visions establishing social networks. A social network will be depicted as a graph, nodes in this graph shows persons and links depicts the relationship among the peoples. Revealing communities in huge real-time graphs like massive social networks could be an issue of appreciable interest. Figure 1 represents a basic graph of 3 clusters, surrounded by the filled circles of three different colors.

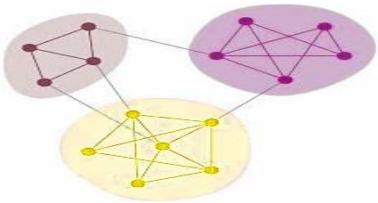

**Figure 1: A simple graph with 3 communities [12]**

Revised Manuscript Received on February 05, 2020.
* Correspondence Author
**M. Mohamed Iqbal,** Assistant Professor, Department of computer science and engineering,Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai.
**K.Latha,** Assistant Professor(Sr. Grade), Department of computer science and engineering, Anna University (B.I.T Campus), Trichirappalli-620024.



In social real world networks, discovering a community implies that discovering a cluster of peoples associated with various units like images, comments, videos, blogs or the other posts. Community Discovery is of high significance in Biology, Sociology and computing disciplines where the system is usually depicted as graphs. We can say that a network graph have the community structure if the vertices of the network graph are merely classified as group of vertices so that every group of vertices is closely interconnected. Discovering communities in such a composite graph could be a tough job. This literature study examining the various community discovery techniques and ways for discovering the good community structure in a real world network. A brief literature of the various community discovery techniques and their methods are stated in section two. Results and discussions are stated in section three. The conclusion and future work are stated in section four.

## II. COMMUNITY DETECTION TECHNIQUES

In social networks, community formations are detected by searching for the vertices that are equivalent to one another and managing those vertices in same cluster. Once the vertices of a graph, belonging to a similar cluster, are organized to make a group, then that graph is alleged to possess a community shape. Community shape is quite general in real-time networks. The community discovery issue has more broad applications and therefore discovering of communities to be very important. Exposure to the information from various sources and clusters is the main advantage of community identification. A community shape contains of objects with equivalent tastes. Discovering of communities creates swapping and diffusing information easier as a result of members of similar community usually has similar tastes.

### A. Louvain Modularity

Blonde and Guillaum et al [1] developed the community detection methodology to find communities on massive network graph. The methodology may be a greedy optimisation technique which tries to improve the modularity of a division of the graph. The optimisation is accomplished in 2 successive steps .Initially, this technique checks for tiny communities by improving modularity in a very native manner. Next, it combines vertices of an equivalent clusters and constructs a new graph whose vertices are the clusters.

This process is depicted in figure 2. Though the precise calculations complication of the strategy is not known, the strategy likes to be execute in O(nlogn) with maximum of the calculations work consumed on the optimization at the first level. This is definitely an approximate technique and nothing assure that the global maximum of modularity is attained, despite many experiments have concluded that this technique has a wonderful precision [1].

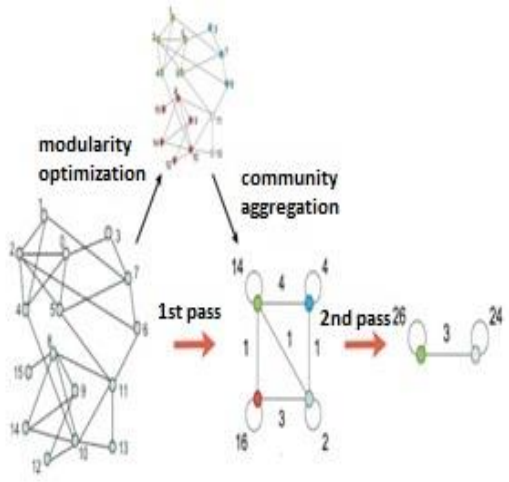


**Figure 2: Louvain method for community detection [1]**

### B. Newman Algorithm

The algorithm widely used to find a community is the Newman algorithm [2]. The method is significant because in the field of community detection it marked the beginning of a new era. This technique identifies edges in network that lies between two communities and then eliminates them, leaving only the communities themselves behind. These edges are identified by one of the graph metrics called as betweenness centrality. It is a measure that appoints a number to each edge that is high if the edge connects the path of every pair of nodes. The main aim of the algorithm is to find the inter community edges and eliminating of them which leads separate communities like Figure 3.

The steps of the algorithm are:

1. Calculating the centrality scores for all the available edges in graph
2. Elimination of edge with biggest centrality score and if any ties with different edges, one among them is elected at random.
3. Recalculation of centralities for all the remaining edges of resulting graph
4. Iterating the process from step 2 to step 3 until there is no connected component in the resultant graph.

The Girvan–Newman algorithm produces results of affordable quality and is standard as a result of it's been implemented in a number of ordinary software packages such as igraph and Networkx in Python. However it additionally runs slowly, taking the running time of $O(m^2n)$ on a graph of n nodes and m links, creating it speculative for graphs of more than a couple of thousand nodes.

### C. Infomap Algorithm

The key idea of the Infomap technique[3] based on closely the Louvain methodology, nearby vertices are merged into components, which eventually are merged into super components and so on. Initially, every individual vertex is appointed to its own component. Then, in arbitrary successive steps, every vertex is shifted to the nearby component that leads to the biggest reduction of the map equation. If no pass leads to a reduction of the map equation, the vertex remains in its native component. This process is reworked, for every time in a new arbitrary subsequent order, till no pass causes a reduction of the map equation. Now the graph is reconstruct, with the components of the last level creating the vertices at this level, and, precisely as at the preceding level, the vertices are grouped into components. This hierarchic reconstruction of the graph is replicated till the map equation can't be decreased farther. Using this technique, reasonably sensible communities of the graph may be identified in very small duration. Let us call this the core algorithm and see however it may be improved. The nodes appointed to the identical components are compelled to move collectively when the network is remodeled. As a outcome, what was an best move early in the algorithm might have the other impact later within the algorithm. As a result of two or a lot of components that combined each other and creates one single component when the graph is reproduce will not ever be separated again in this method, the precision may be enhanced by dividing the components of the ultimate state of the core algorithm in one of the two-following-ways:

i. Submodule movements. First, every cluster is considered as a graph on its own and the core algorithm is adapted to the current graph. This procedure generates one or a lot of sub components for every component. Then all sub components are moved back to their individual components of the preceding step. At this stage, with an equivalent partition as within the previous step however with every sub component being freely movable between the components, the core algorithm is re-adapted on the sub components.

ii. Single-vertex movements. Initially, every vertex is re-appointed to be the sole representative of its own component, so as to grant for single-vertex movements. Then everyone vertices are back down to their individual components of the preceding step. At this stage, with an equivalent partition as within the preceding step however with every individual vertex being candidly movable between the components, the core algorithm is re-applied on the single nodes.

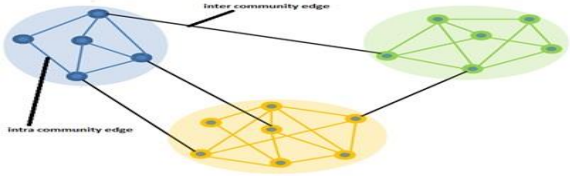


**Figure 3: A Graph with intra community edges and inter community edges [2]**

In practice, this method have a tendency to do again the two extensions to the main algorithm in sequence and as long because the clustering is enhanced. Moreover, this algorithm has a tendency to apply the sub module movements recursively. That is, to find the sub modules to be moved, the algorithm first splits the submodules into subsubmodules, subsubsubmodules, and so on till no more splits are possible. Finally, as a result of the algorithm is random and fast, this able to restart the algorithm from scratch whenever the clustering can't be enhanced further and also the algorithm terminates. The implementation is simple and, by repeating the search quite once, 100 times or more if possible, the ultimate partition is less likely to correspond to a local minimum. For every iteration, this method record the clustering if the confession length is shorter than the preceding shortest confession length.

### D. Clique guided community detection

A new approach developed by Diana and Mostofa et al [4] for rapid and effective community detection method. Clique guided community detection includes two stages. During the initial stage, this method locates the disjoint cliques. In the second stage, the cliques identified from the first stage are utilized to manage the joining of individual nodes till a better quality result is achieved. For the primary stage, this method developed an algorithm named MACH (Maximum in group Heuristic), that is a novel method to calculate disjoint clique based on the heuristic-based branch-and-bound method. As the method is adopted the community merge consumes O(k) time, where k represents total communities identified. If the joining is not balanced during this step, it might do O(n) joins, and it consumes $O(n^2)$ time for this phase[2].

### E. Leading Eigenvector method

Newman and Girvan et al [5] developed this technique. The main theme of this technique is the spectral optimization of modularity by utilizing the Eigen values and Eigenvectors of the modularity matrix of the given network graph. First by using modularity matrix of the graph, the principal eigen vector is identified, and then the graph is divided into two components so that modularity improvement is increased depends on the principal eigenvector. Afterwards the modularity improvement is identified at each successive step in the subdivision of a network graph. It halts once the contribution of modularity is negative. Its running time complexity of every network graph sub division is O(N(E+N)), or $O(N^2)$ on a scattered graph, in which N is the number of vertices in each graph before partition and E is the total available links in the graph.

## III. RESULTS AND DISCUSSIONS

### A. Data set Description

This paper experiments the community detection algorithms on two real-world datasets. 1) Twitter [6] user follower network graph. It includes 318,233 Twitter account holders with 3, 545, 258 directed links. The undirected graph is obtained by eliminating all non mutual edges and also removes isolated vertices from the network graph, because each community detection techniques described in this study not assists directed graph. After this preprocessing, the network graph has 190500 vertices and 1001528 undirected links. 2) DBLP dataset [7] which is a co-authorship social network graph in computer science, in this graph every vertex indicates an author, every link represents co-authorship of a research paper and two authors are linked with edges if they combinedly produce at least one paper.. There are a total of 317,080 vertices and 1,049,866 undirected links in DBLP Network. The network characteristics of the two Networks are shown in Table 1

**Table1. Network characteristics of two datasets**

| Network Characteristics | Twitter* | DBLP** |
|---|---|---|
| Nodes | 318,233 | 317080 |
| Edges | 3, 545, 258 | 1049866 |
| Average clustering coefficient | 0.2304 | 0.6324 |
| Diameter (longest shortest path) | 19 | 21 |

*graph characteristics of Twitter social network
**graph characteristics of co-authorship network in computer science society

### B. Comparison of Community Discovery Methods

In this sub division, first, the size dissemination of the identified communities by different methods is studied and calculates the percentage of vertices appointed to a largest cluster derived by all the algorithms. Then the identified clusters are divided into four categories <1:50>, <51:250>, <251:500>, <501+> based on the community size and this study analyzes the purity of communities in each derived group with the popular quality measurement metric of Extended Modularity. For the implementation of community detection algorithms, the proposed work utilizes the python packages igraph and NetworkX on both Twitter and DBLP networks. Table 2 and 3 depicts the total count of clusters (communities) and the density of the biggest cluster (community) derived by the proposed community detection algorithms. This paper also incorporates the percentage of nodes that are allocated to the biggest community derived by each algorithm. From the tables 2 and 3, it shows that InfoMap, Newmaan, Clique, Eigen Vector all produce extremely large communities for both datasets and some of the community having more number of users from the entire network. For instance, the densest community structure produced by the eigen vector algorithm contains 13,6401 nodes that means throughout the 70% of total nodes of Twitter data set is clustered into one larger cluster. In the DBLP Network, the Eigenvector method generates the densest community of size 315,569, which is 99.5% of all its total users. From this, it shows that the Eigenvector method produces minimum number of communities with huge sizes than other community detection methods for both the data sets. A more elaborate division of total communities in various size length is represented in table 4 and 5. It shows that Clique and Infomap generates maximum number of communities in preferable ranges of <1:50> and <51:250> than other methods.

Another interesting statement is that Eigenvector does not produce any communities in the range <51:250> and<251:500> on Twitter Network and also does not generate any communities on DBLP Network for the range of <1:50>, <51:250>and<251:500> . Based on the community size dissemination, it shows that InfoMap, Louvain and Newman methods can generate a reasonable number of communities with desirable sizes and perform better than other two community detection methods.

**Table 2. Communities Generated by different Algorithms on Twitter Dataset.**

| Network Characteristics | Twitter* | DBLP** |
|---|---|---|
| Nodes | 318,233 | 317080 |
| Edges | 3, 545, 258 | 1049866 |
| Average clustering coefficient | 0.2304 | 0.6324 |
| Diameter (longest shortest path) | 19 | 21 |

*communities identified in each algorithm is listed
**total no. of nodes of largest community identified
***percentage=(number of nodes in largest community/total number of nodes in network)*100

**Table 3. Communities Generated by different Algorithms on DBLP Dataset.**

| Network Characteristics | Twitter* | DBLP** |
|---|---|---|
| Nodes | 318,233 | 317080 |
| Edges | 3, 545, 258 | 1049866 |
| Average clustering coefficient | 0.2304 | 0.6324 |
| Diameter (longest shortest path) | 19 | 21 |

**Table 4. Communities Grouped in different sizes on Twitter Dataset**

| Size Range* | InfoMap | Newman | Louvain | Clique | Eigen Vector |
|---|---|---|---|---|---|
| <1-50> | 16232 | 3040 | 415 | 27754 | 0 |
| <51-250> | 756 | 100 | 10 | 456 | 0 |
| <251-500> | 9 | 20 | 13 | 3 | 0 |
| <500+> | 2 | 46 | 127 | 0 | 2 |

*grouping of identified communities of four different sizes

**Table 5. Communities grouped in different sizes on DBLP Dataset**

| Algorithm | Number of communities identified* | Largest Community size** | Percentage of nodes in largest community*** |
|---|---|---|---|
| InfoMap | 18,537 | 13126 | 6.90% |
| Newman | 9350 | 51781 | 27.20% |
| Louvain | 7409 | 34955 | 18.40% |
| Clique | 12,312 | 680 | 0.40% |
| Eigen vector | 5834 | 136401 | 71.50% |

*grouping of identified communities of four different sizes

In order to find the accuracy of the identified community clusters in different algorithms, this paper utilizes the Extended Modularity Measure. Extended modularity is a standard metric which is used to calculate the purity of overlapping community clusters [9-10]. The interpretation of the metric is given in equation (1). Here , $C_i$ is the ith community , $N_v$ represents total number of community clusters the node v resides in, $N_w$ represents total number of community clusters the node w resides in and M is the square matrix in which $M_{vw}$ =1 indicates that there is a link among node v and node w and $M_{vw}$ = 0 indicates there is no link among node v and node w. $d_v d_w/2m$ indicates the anticipated no. of links among node v and node w. $d_v$ and $d_w$ represents the degree of node v and node w in the whole network graph respectively. m is the total number of links in the given network graph. The extended modularity have the value within the range of -1 to 1. The high value of this metric indicates the more purity of the derived community in terms of modularity.

$$EQ = \frac{1}{2m}\sum_i \sum_{v,w \epsilon C_i} \frac{1}{N_v * N_w} \left[M_{vw} - \left[\frac{d_v * d_w}{2m}\right]\right] \quad (1)$$

From Equation (1) it shows that every community accords a value regarding the modularity score. To calculate the impact of the various sizes of communities on the modularity score , the proposed method divides the sum of modularity score in Equation(1) as 4 components which is the total score of all community clusters of range <1:50>, <51:250>, <251:500> , <501+> and the modularity score caused by every cluster group represented in Figure 4 and 5. It shows that Infomap performs better than others because it attains best modularity score which is caused by all its 4 categories of its community clusters but Louvain, Newman and Eigenvector modularity scores are often caused by the huge community clusters of range <501+>. Actually, Infomap's modularity value is very near to the highest value attained by Louvain method for both data sets. From these result discussions, it shows that the communities of different sizes generated by InfoMap algorithm are more strongly connected with their nodes than other community mining algorithms.

This observation also suggests that the modularity score needs to be maximized although managing the size of the community structure. It is important to observe that commonly overlapping community clusters [8] posses minimum modularity than non overlapping community clusters therefore the minimum modularity of Clique is anticipated response.

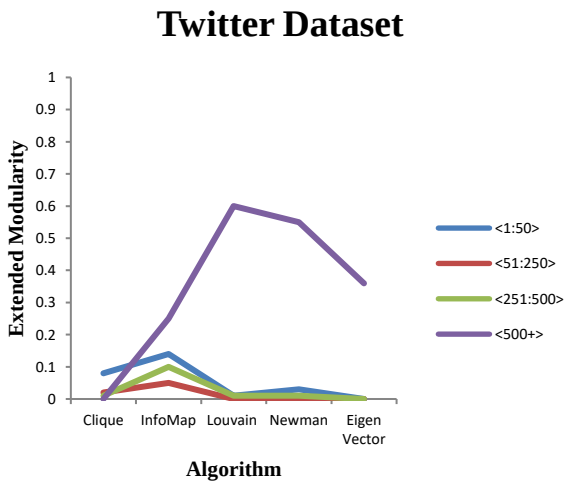


**Figure 4: Modularity score of communities for different algorithm on Twitter Dataset**

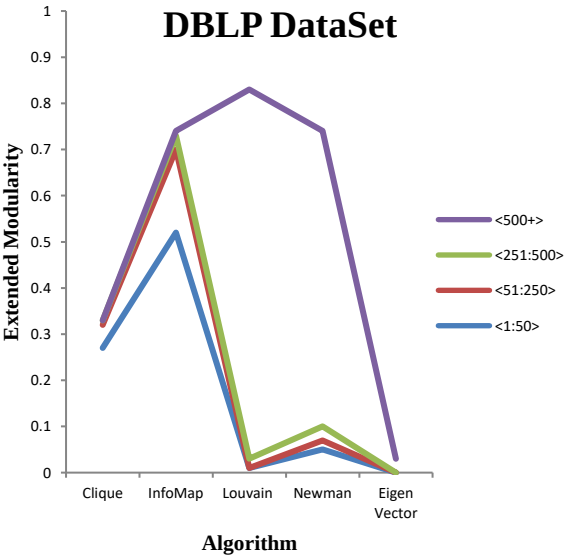


**Figure 5: Modularity score of communities for different algorithm on DBLP Dataset**

## IV. CONCLUSION AND FUTURE WORK

Community discovery plays a major part in the massive social networks and further assists in grasping the shape of social networks. Community mining methods are extensively utilized to analyze the structural and topographical characteristics of real-time networks. The proposed work compares various community discovery methods for 2 social networks twitter and DBLP. The communities identified by the different algorithms are grouped into four categories such as <1:50>, <51:250>, <251:500> and <501+> in order to measure the community distribution of various sizes. The proposed work presents and compares the performance of each community mining algorithm by using an Extended Modularity measure. The number of communities derived by each algorithm is not uniform for both data sets. In future work, we planned to find the influential (seed) nodes in the given real-time networks and derive the communities from these seed nodes in a bottom-up manner. Thereby we can predict and produce the number of communities that equals the number of seed nodes.

## REFERENCES

1. Blonde, V. D, Guillaume, J.-L., Lambaste, R. & Lefebvre E (2008) , “Fast unfolding of communities in large networks” Journal of Statistical Mechanics: Theory and Experiment.
2. “Girvan-Newman community detection method” https://arxiv.org/pdf/0906.0612v2.pdf
3. “Infomap-community-detection method” http://www.mapequation.org/code.html.
4. Diana Palsetia , Md. Mostofa Ali Patwary (2014), “Clique Guided Community Detection”, IEEE International Conference on Big Data.
5. M. E. J. Newman (2006), “Modularity and community structure in networks” PNAS, vol. 103, no. 23, pp. 8577–8582.
6. “Twitter Social Network Dataset” http://konect.uni-koblenz.de/
7. “DBLP Co-Authorship Network Dataset” http://snap.stanford.edu/data/com-DBLP.htm
8. H. Shen, X. Cheng, K. Cai ( 2009) , “Detect Overlapping and Hierarchical Community Structure in Networks” Physica A: Statistical Mechanics and Its Applications 388.8.
9. J. Pinney and D. Westhead (2007), “Betweenness based decomposition methods for social and biological networks” Interdisci-plinary Statistics and Bioinformatics, pp. 87–90.
10. M. E. J. Newman and M. Girvan (2004), “Finding and evaluating community structure in networks” Physical Review E, vol. 69, no. 2, p. 026113.
11. S. Fortunato and M. Barthelemy (2007),“Resolution limit in community detection” Proceedings of the National Academy of Sciences, vol. 104, no. 1, p. 36.
12. Mehjabin Khatoon, W. Aisha Banu (2015), “A Survey on Community Detection Methods in Social Networks”,I.J. Education and Management Engineering, 1, 8-18.

## AUTHORS PROFILE

**Mohamed Iqbal M**, He is working as Assistant Professor, Department of computer science and engineering in Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai-62. He is also Part-Time PhD Research Scholar in Anna University, Chennai-600025.

**Dr. K. Latha**, she is working as Assistant Professor(Sr. Grade), Department of computer science and engineering in University College of Engineering, Anna University (B.I.T Campus), Trichirappalli-620024. She Published papers in various Scopus indexed journals, National and International conferences and Journals. She is specialization in Information Retrieval, Information Extraction, Data Mining and Cloud Computing.