International Journal of Engineering & Technology, 7 (4.6) (2018) 306-313

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Overlapping Community Detection Algorithms: A Comparative Study

Subham Datta[1]*, R. Dinesh[2],Tapas Saha[3], R. Subramanian[4]

[1,2,3,4]Dept. of Computer Science, Pondicherry University, Puducherry, India, 605014.
**Corresponding author E-mail:subhamdatta@yahoo.com*

Abstract

In complex networks a node may belong to many communities resulting in a highly overlapping community structure. This provides multiple information about such nodes by analyzing the communities they belong to. Recent advances in benchmarking haslead to the fact that most of the popular CAA (Community Assignment Algorithms) works only when the extent of overlap in a network is modest. GCE happens to be one such CAA with the ability to report communities in a network graph with huge accuracy. In this paper, we have discussed several existing state of art methods for detecting of overlapping communities with their approaches and disadvantages. Also we presented experimental evidence of how the extension of GCE algorithm (EGCE) outperforms the other existing overlapping community detection algorithm. At the same time we have analyzed the performances of the other existing algorithms.

Keywords:*Overlapping Community Detection, Greedy Clique Expansion, Complex Networks, Extended Greedy Clique Expansion.*

1. Introduction

Many of the real systems can be modeled as networks or graphs, where nodes denotesobject or user (for example, a social network) and edges denotes the relationship or interactions among the users. A node may belong to more than one group, and as the system grows, more of these nodes may belong to even more groups. It is important to be able to characterize or understand the behavior of each node. It becomes a complex issue as communities grow bigger in size and the degree of overlap increases. In this paper such community detection algorithms are presented and analyzed.

Several definitions of a community has already been proposed, however, no generally accepted definition of a community has been explicitly stated. Forunato [1] has stated that community structure to be those groups of nodes that share a common property or interest in a graph. What we can confidently say about a community, based on information obtained from a number of papers that had already dealt with community detection, is that there is a highly dense number of edges (interactions) within a community while lesser number of edges going out of a community. The concept of modularity [2] measures how dense a community is.

A number of algorithms for detecting overlapping communities are available in literature. It would be of great value if it is known which algorithm provides good, even best, performance over the others across different datasets. A comparison on some of the available methods was recently done by Xie et al. [3]. It has also been discussed in literature that several of these existing algorithms perform poorly when several of the nodes belong to many communities [4]. In the paper of Madhusudan et al.[5], the authors have proposed an extension to Greedy Clique Expansion (GCE) [4] to take care of this issue.Since each algorithm has its own core technique, it is important to know their performance on synthetic and real networks.

The contributions of this paper are –

• In this paper a detailed comparison between several state of art techniques are given.
• We have selected top six algorithms which has good accuracy in overlapping community detection for our experiment. Extension of GCE algorithm (EGCE) shows the best accuracy in overlapping community detection among the six selected algorithms.
• Finally in this paper we discussed how this same extension can be used for existing algorithms to improve their performance.

In the next section we will summarize the methods which have been used in our study. Section 3 contains the experiments that we have performed for comparing the selected overlapping community detection algorithms. In section 4, we have shown the result of our work. Finally in section 5, we concluded with some observations on overlapping community detection methods.

2. Overlapping Community Detection Methods

Xie et al. [3] has classified overlapping community detection algorithms into five categories - clique percolation based, link partitioning based, agent based, local expansion and optimization based, and fuzzy based detection. In Table 1 [3], the overlapping community detection algorithms under each category is described with their advantages and disadvantages.The following are the selected algorithms with their implementation that have been used to study and evaluate in this paper.

Copyright © 2018Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

2.1. The CFinder

The CFinder [6], one of the most commonly evaluated algorithm, makes use of the Clique Percolation Method to detect overlapping communities. CFinder generates communities of k-cliques, where k is the number of nodes, and practically, it starts finding communities starting from k 3 unless otherwise specified. If two k-cliques share (k-1) nodes, then they are adjacent [6]. The method first selects a clique of size k, then adds a neighboring node in the clique, thus formed and the previous clique share (k-1) nodes. Otherwise, the k-clique will not add that node, but look for the other neighboring nodes. This process of adding more cliques of size k stops when there are no more cliques of the same size, to be added, are found. When k is incremented, cliques of the incremented size are added if they are adjacent. Thus, the union of all these k-cliques, (k+1) cliques, and so on, gives the final communities. Now, CFinder lists each community per line. A node listed in more than one of the communities represents the overlapping (common/shared) node.

2.2. Greedy Clique Expansion

The Greedy Clique Expansion [4] is based on local expansion of a fitness function. It works by first finding all maximal cliques of a particular size as seeds and greedily expanding them one by one based on a local fitness function. The fitness function used here was defined by Lancichinetti et al [7] –

$$F_M = \frac{K_{in}^M}{(K_{in}^M + K_{out}^M)^\alpha}$$

where α is the parameter which adjusts the size of the community.$K_{in}^M$is total internal degree of nodes of community M and K_{out}^M is total external degree of nodes of community M. The expansion stops when the fitness of a community becomes lowered with the addition of a node. As each seeds are expanded, if community C' is within a distance of some with already accepted community C, then they are near duplicates and C' is dropped. As the seeds are expanded, a node present in one community may also be found in another community. Nodes like these are where the overlapping of communities occurs for the entire network.

2.3. Link Community

Line graph and link partitioning is the heart of this algorithm by Ahn [8]. The algorithm begins by taking a node and comparing its similarity to its neighboring nodes exactly like the single-linkage hierarchical clustering. However, in this algorithm, the similarity between edges is calculated using Jaccard Index –

$$J(E_{xz}, E_{xz}) = \frac{|N_x \cap N_y|}{|N_x \cup N_y|}$$

where E_{xz} and E_{yz} are edges where one of their endpoints meet at vertex z. N_x is the neighborhood of vertex x including x. The node with highest similarity is joined (clustered) to the first node. This process repeats for all other nodes, forming a dendrogram, until there are no nodes to compare and cluster. Then, the dendrogram is cut at a certain threshold given by –

$$D \equiv \frac{2}{e}\sum_c e_c \frac{e_c - (n_c - 1)}{(n_c - 1)(n_c - 2)}$$

wheree_c is the number of edges in partition c, n_c is the number of nodes in partition c and $E = \sum_c e_c$. Thus, the communities are detected.

2.4. Community Overlap Propagation Algorithm

In Community Overlap Propagation Algorithm(COPRA) [9], each node is given a label. In the next step, a node checks for the label of its neighbors, and it replaces its label with the label used by the maximum number of neighbors. This process is repeated for a certain number of times. Now, all nodes with the same label will be placed into the same community. Nodes with multiple labels denote the overlapping nodes. In overlapping community, a node has its own belonging coefficient and a label that identifies it to a community where it belongs. A node calculates this belonging coefficient of its neighbors and changes its own with this new value. This belonging coefficient for a node, which is a member of a particular community, is given by summing all the belonging coefficients of its neighbors (inside the same community) and dividing this sum by the total number of its neighbors(inside the same community). In COPRA, we have to specify the number of nodes with which we want the algorithms to find the community. By default, the algorithm finds the community with respect to one node.

2.5.ClusterONE

The Clustering with Overlapping Neighbourhood Expansion (ClusterONE) [10] is mainly meant to detect overlapping communities in protein-protein interaction networks. At first, the algorithms selects a node with the highest degree and grows greedily. This is the first seed. Now, after this growth is completed, the algorithm looks for the next highest degree node that was not covered yet. This process continues until all nodes has been exhausted. While a seed is growing it is possible that the community thus formed becomes very similar to an already detected community. If such was the case, then these two are merged based on a similarity measure Om. Say, there are two communities X and Y, Om is defined as –

$$Om = \frac{|X \cap Y|}{|X||Y|} * |X \cap Y|$$

Communities with less than three nodes are dropped. Also, if a community's density value is less than a certain threshold, then it is dropped. The threshold is given by –

$$del = \frac{e^{in}(X)}{V^{\frac{(V-1)}{2}}}$$

where, $e^{in}(X)$ is the total internal weight of the links of community X, and *v* represent the total number of nodes.

Table 1: Summary of state of art algorithms on overlapping community detection.

Category	Approach	Methods	Advantage	Disadvantage
Clique Prelocation	Assume that sets of complete sub-graphs that overlaps makes up communities. Search for adjacent cliques to detect communities.	1. Clique community statistics 2. Label Propagation 3. Weighted Network Modules	Simple approach that detects by searching for specific structures. Useful for dense networks.	With the increase in the number of nodes the some methods show non termination.

		4. Sequential Algorithm		
Line Graph and Link Propagation	Clustering followed by detection of overlaps based on inclusion of links in multiple clusters.	1. Communities as group of links. 2. Line Graphs and Link partitions 3. Adjustable extent of overlapping 4. Map equation 5. Maps of random walks 6. Link based extended modularity	The time complexity depends upon the maximum node degree.	No quality guarantees better than node based approaches
Local Expansion and Optimization	Grow natural or partial community based on a local benefit function.	1. Graph clustering into overlapping sub-graph 2. Modified clustering 3. Expansion from Random seed 4. Local calculation of community changing resolution levels 5. Statistical significance 6. Markov random walk under constraint 7. d-Dimensional vector mapping 8. Maximal node strength selection 9. Agglomerative framework 10. Maximum cliques as seed communities 11. Induced independent maximal cliques as cores	Use of local functions that optimizes the community properties.	In some of the methods the quality of the detected communities depends upon the seeds used.
Fuzzy Detection	Strength of association is quantified to detect the overlaps.	1. Non linear constrained optimization 2. Spectral clustering framework 3. Probabilistic mixture generative model 4. Minimum description length 5. Clearing the fog 6. Multiplicative mixture models 7. Sampled spectral distance embedding 8. Overlapping stochastic block models 9. Overlapping seed expansion 10. Non-negative matrix factorization	The belonging factor or membership vector can be calculated from the data.	Detection of membership dimensionality.

		11. Affinity propagation clustering algorithm		
Agent Based and Dynamical Algorithms	Mainly an extension of label propagation algorithm	1. Label propagation algorithm 2. Speaker-Listener based information propagation process 3. Nash local equilibrium based game theoretic framework 4. Particle competition 5. Potts model approach 6. Information-based replica correlation	The interaction between the nodes are taken into consideration.	Complexity in determination of the node's existence in multiple sub-graphs.

2.5. Extended GCE (EGCE)

This algorithm begins by finding the communities using GCE. For each community detected, it takes a particular community and calculates the Interaction Probability for each node of the neighboring community. The Interaction Probability is defined as –

$$IP_{u,M} = \frac{|(u,v):\{(u,v) \in E \text{ and } v \in M\}|}{|M|}$$

Where E is the total number of edges in the community, u and v are nodes in community M, and u does not belong to M. Then, it selects a node that has the minimum Interaction Probability with respect to the selected community, and clusters over it. After this, it finds the clusters with minimum Interaction Probability and merges this cluster with the selected community. This process continues until all sub-communities within the selected community are exhausted.

3. Experiment

We have used synthetic dataset to compare the performance for each algorithms. For testing the algorithms we start by setting up the various parameters of the LFR benchmark [11,12]. The following table (Table 2) shows the different parameters except for the maximum degree of a node, max_k 50, degree distribution parameter, t_1 2, community size distribution parameter, t_2 1, (both distributions follow the power law) and average degree k 20, which were kept constant. Small graph corresponds to number of nodes, N 1000 with overlapping nodes, O_n 250 and small community corresponds to minimum community size, min_C 10 with maximum community size max_C 50. Large graph is taken to be N 5000 with O_n 1250 while large community is taken as min_C 20 with max_C 100.

For EGCE and GCE we have kept the parameters minimum clique size, k 4, overlap to discard eta 0.6, fitness exponent, α 1.0, clique coverage heuristic threshold Φ 0.75. For COPRA we tested the algorithm with two types of settings - first with the parameters vs (number of vertices) from 1 to 10 and for each vertex the algorithm repeats 10 times with the repeat option. Secondly, with v (find communities with respect to a vertex) option set to 1000, and the extra simplify option set to true, we test the algorithm again, the later option is used so that the communities thus founded are not included multiple times. For CFinder, we use the undirected and un-weighted version, where clique size k starts from 3. For ClusterONE and linkCOMM, we have used their default parameters.

Table 2: Parameters for generating benchmark graphs

mu(μ)	Numbers of Nodes (N)	min_C	max_C	O_m	O_n
0.1	1000/5000	10/20	50/100	3 to 10	250/1250
0.2	1000/5000	10/20	50/100	3 to 10	250/1250
0.3	1000/5000	10/20	50/100	3 to 10	250/1250
0.4	1000/5000	10/20	50/100	3 to 10	250/1250

3.1. Test on Synthetic Benchmark

As mentioned, we have used the LFR benchmark to generate synthetic network which is undirected and un-weighted with different parameters as shown in the experimental set up. Note that, in LFR benchmark all nodes are assigned to at least one community, that is no nodes are isolated. We generated ten instances of the benchmark graph and these were used as input for each algorithms. The way the benchmark graphs are generated is shown in the following algorithm –

```
1.  let f[1]="smallGraph smallCommunity"
2.  let f[2]=="smallGraph largeCommunity"
3.  let f[3]="largeGraph smallCommunity"
4.  let f[4]="largeGraph largeCommunity"
5.      for each mu from 0.1 to 0.4
6.          for each f [ ] from 1 to 4
7.              for each Om from 3 to 10
8.                  for each instances from 1 to 10
9.                      generate benchmark graph and commu-
    nity
10.                     input benchmark graph to -
11.                       GCE,
12.                       COPRA,
13. CFinder,
14. ClusterONE,
15. LinkCommunity, and get corresponding output graph
16.                 end for
17.             end for
18.         end for
19.     end for
```

3.2. Evaluation Criteria

We have used the Normalized Mutual Information [7] to evaluate each synthetic community compared to the respective algorithms (those mentioned above) on synthetic network. The following algorithm summarizes the comparison and evaluation of each overlapping community detection algorithm with each synthetic communities that was obtained from the previous section –

```
1.  let f[1]="smallGraph smallCommunity"
2.  let f[2]="smallGraph largeCommunity"
3.  let f[3]="largeGraph smallCommunity"
4.  let f[4]="largeGraph largeCommunity"
5.      for each mu from 0.1 to 0.4
6.          for each f [ ] from 1 to 4
7.              for each Om from 3 to 10
8.                  for each instances from 1 to 10
9.                    compare each synthetic commu-
    nity with-
10.                       output graph of GCE,
11.                       output graph of COPRA,
12.                       output graph of CFinder,
13.                       output graph of ClusterONE,
14.                       output graph of LinkCommuni-
    ty,
15. end for
16.                 compute mean and standard devia-
    tion
17.             end for
```

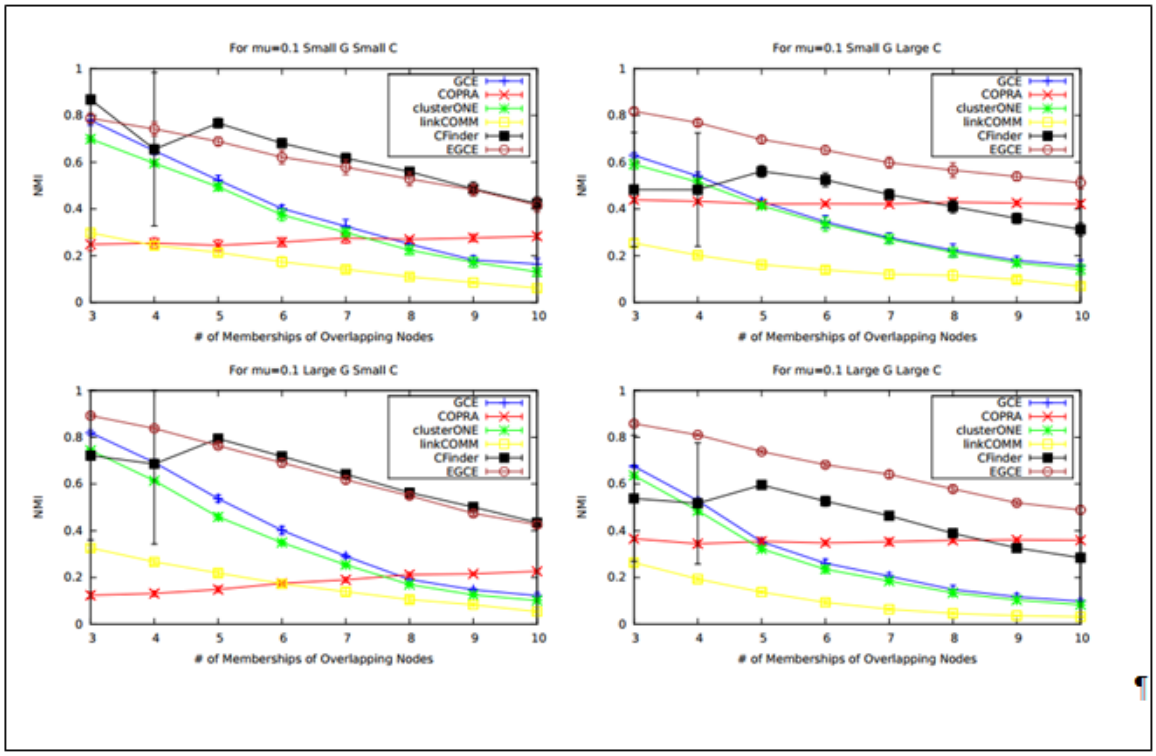


Figure 1: NMI vs Om for mu=0.1

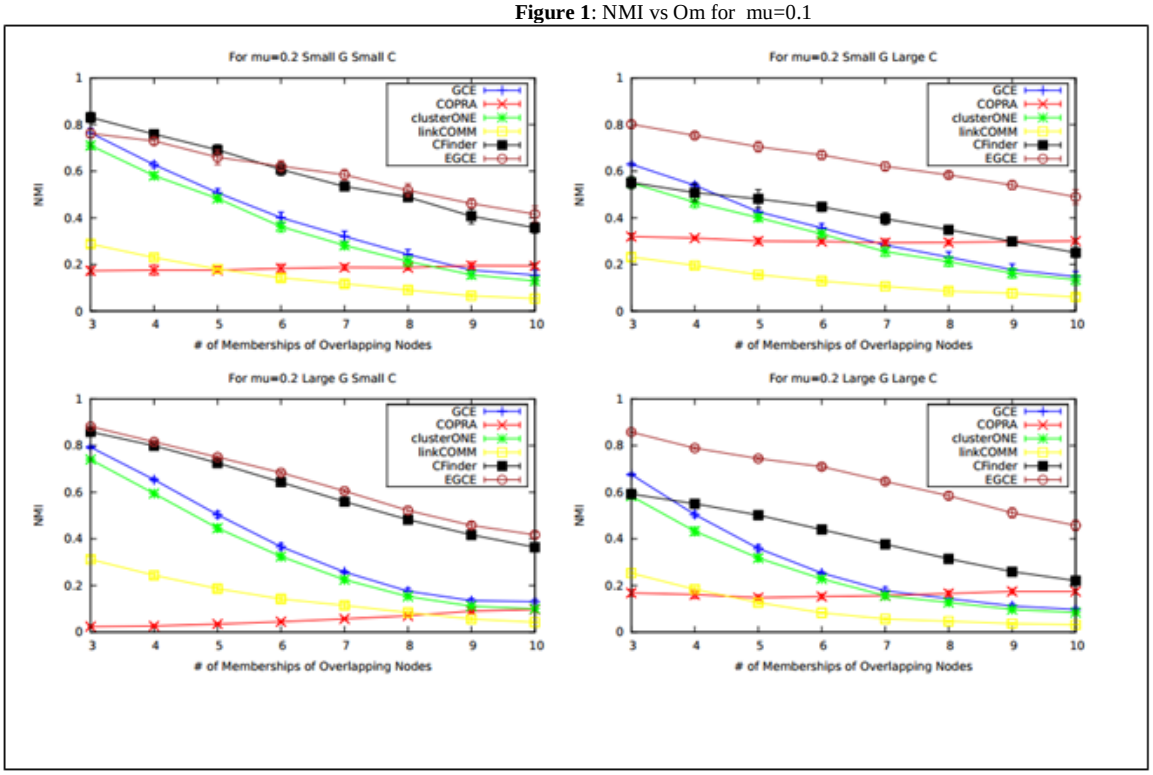


Figure 2: NMI vs Om for mu=0.2

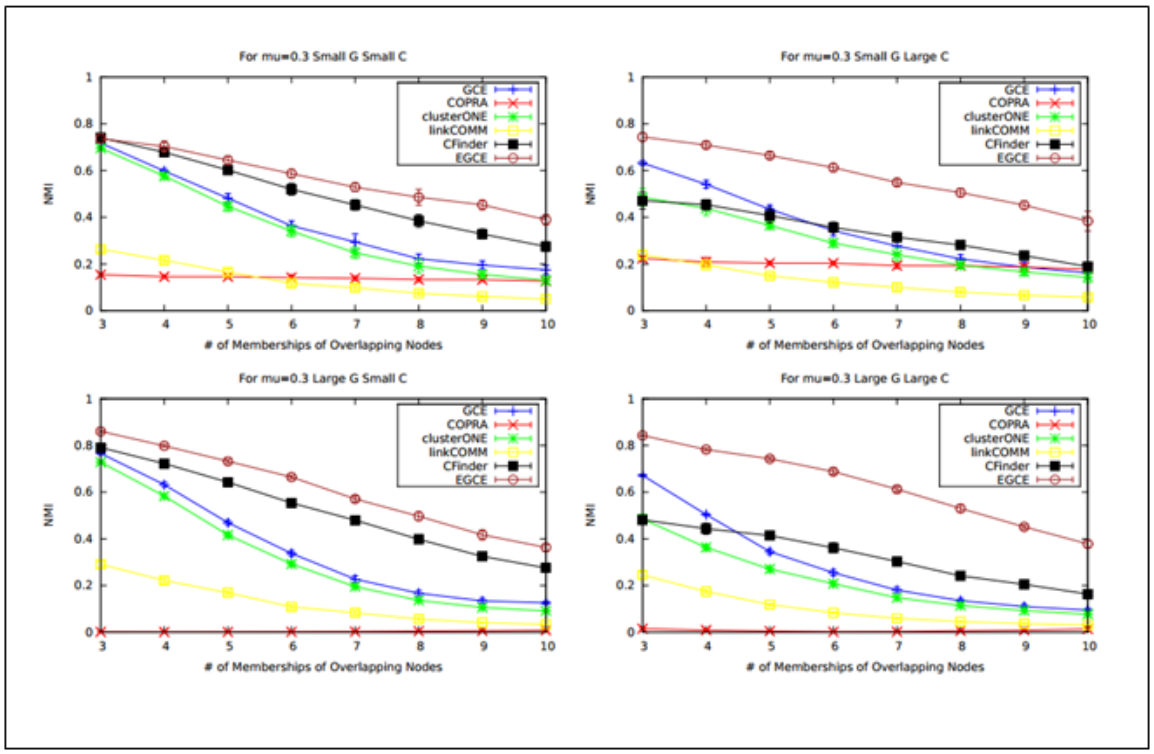


Figure 3: NMI vs Om for mu=0.3

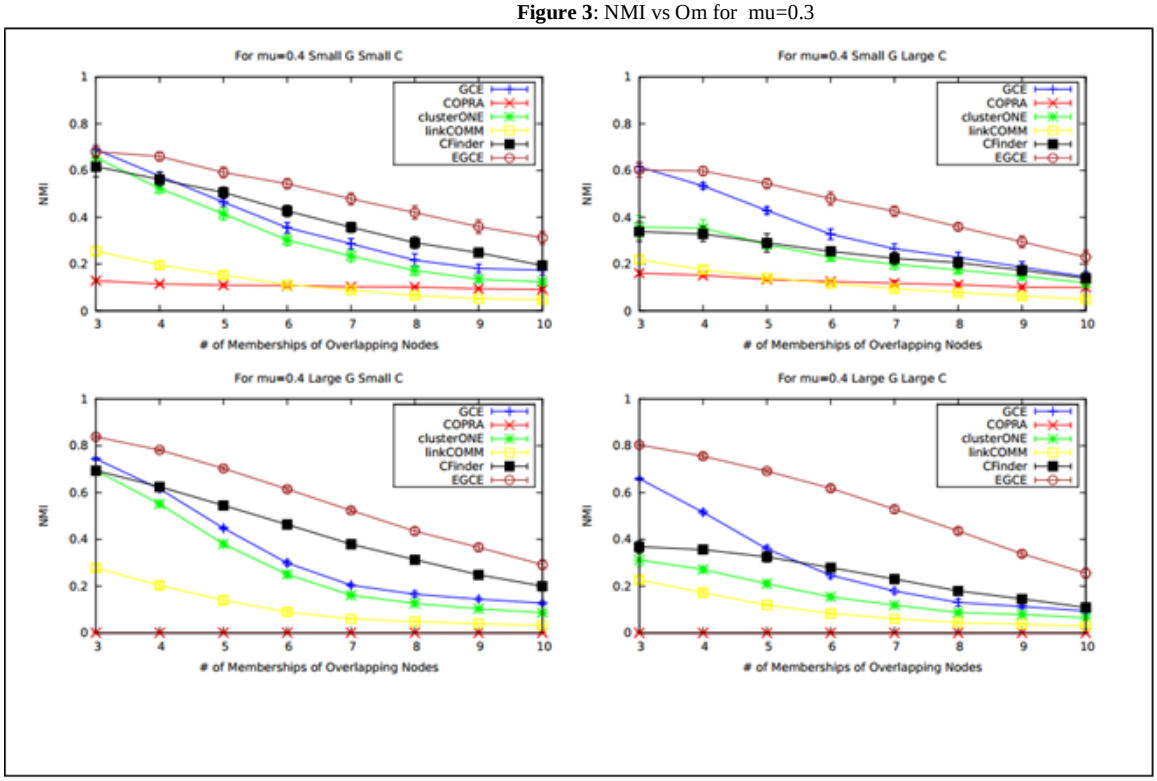


Figure 4: NMI vs Om for mu=0.4

4. Result

For small graph, when the fraction of outgoing edges are very small ($\mu = 0{:}1$ and $\mu = 0{:}2$) the EGCE and CFinder are performing appreciably well as shown in Figure 1. But as the overlapping membership of the nodes increase, performance of most algorithms, for finding overlapping communities, decreases [13]. From Figure 2 through 4, CFinder's performance decreases. Also, observe that COPRA and linkCOMM maintains an almost linear performance even as nodes with overlapping memberships increases, while their NMI values are low. GCE and clusterONE's performance are mostly at the same level, however, their NMI value shows clusterONE performs worse than GCE . But as the overlapping membership increases, except for COPRA, all other algorithms seem to deteriorate in performance. Nevertheless, the NMI value for COPRA is very small compared to GCE and EGCE.

When μ and Om values increases ($\mu = 0{:}3$ and $\mu = 0{:}4$), EGCE still maintain highest NMI values amongst all as can be seen in Figure 3 and Figure 4. It is worth noting that, in the case of large graph with small community, EGCE maintains a decent perfor-

mance with respect to its NMI value considering all cases of μ from 0.1 to 0.4 (Figure 1 through 4). Hence, the results show that EGCE is performing better than all the other overlapping community detection algorithms we used in this study.

5. Conclusion

In all cases, we have used a fixed membership, which is with respect to O_m and O_n values (that is, membership of overlapping nodes are not altered apart from those set by the LFR benchmark). If we were to use varying memberships, we would have set the values of O_m and O_n to 0 each, because we would have taken the membership from a different file. This case was not considered in this experiment.

We would like to conclude that EGCE's performance was undoubtedly the best in identifying overlapping communities. One last observation is during the experiment, it was found that COPRA, on setting its vs option to a certain $v_1 \ldots v_n$ and repeat option to some value r, is computationally slow. For all the others they performed as was claimed in each paper. Thus, even though COPRA was found to be one of the best algorithms in [3], we would like to conclude otherwise.

References

[1] Santo Fortunato(2010),"Community detection in graphs".*Physics Reports*, 486(3-5),pp.75-174.

[2] Newman, Mark EJ(2006),"Modularity and community structure in networks",*Proceedings of the National Academy of Sciences*, 103(23),pp.8577- 8582.

[3] JieruiXie, Stephen Kelley, and Boleslaw K Szymanski(2013),Overlapping community detection in networks: the state of the art and comparative study. *Acm computing surveys (csur)*, 45(4), 43.

[4] Conrad Lee, Fergal Reid, Aaron McDaid, and Neil Hurley(2010), Detecting highly overlapping community structure by greedy clique expansion. *arXiv preprint arXiv*:1002.1827,

[5] Paul, M., Anand, R., & Anand, A. (2015). Detection of Highly Overlapping Communities in Complex Networks. *Journal of Medical Imaging and Health Informatics*,5, 1099–1103.

[6] GergelyPalla, ImreDer´enyi, Ill´esFarkas, and Tam´asVicsek(2005),Uncovering the overlapping community structure of complex networks in nature and society. *Nature*,435, 814-818.

[7] Andrea Lancichinetti, Santo Fortunato, and J´anosKert´esz (2009), Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*11, 033015.

[8] Yong-YeolAhn, James P Bagrow, and Sune Lehmann (2010), Link communities reveal multiscale complexity in networks.*Nature* 466, 761-764.

[9] Steve Gregory (2010), Finding overlapping communities in networks by label propagation. *New Journal of Physics*12, 103018.

[10] Nepusz, Tamás, Haiyuan Yu, and Alberto Paccanaro 2012), Detecting overlapping protein complexes in protein-protein interaction networks. *Nature methods* 9, 471.

[11] Lancichinetti A, Fortunato S.(2009), Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review*E 80, 016118.

[12] Lancichinetti A, Fortunato S, Kertész J.(2009) Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*11, 033015.

[13] SubhamDatta, Dinesh Karunanidy, J. Amudhavel, ThamizhSelvamDatchinamurthy, Subramanian Ramalingam (2017), A study on Identification of Static as well as Dynamic Protein Complex and Functional Modules in PPI Network. *IIOAB Journal* 8, 239- 251.