2015 IEEE Symposium Series on Computational Intelligence

Overlapping Community Detection in Social Network Using Disjoint Community Detection

Jaswant Meena
Department of Computer
Science and Automation
Indian Institute of Science
Bangalore, Karnataka-560012
Email: jaswant.meena10@gmail.com

V. Susheela Devi
Department of Computer
Science and Automation
Indian Institute of Science
Bangalore, Karnataka-560012
Email: susheela@csa.iisc.ernet.in

***Abstract*—With increasing popularity and complexity of social networks, community detection in these networks has become an important research area. Several algorithms are available to detect overlapping community structures based on different approaches. Here we propose a two step genetic algorithm to detect overlapping communities based on node representation. First, we find disjoint communities and these disjoint communities are used to find overlapping communities. We use modularity as our optimization function. Experiments are performed on both artificial and real networks to verify efficiency and scalability of our algorithm.**

I. Introduction

Social networks are widely used in human society. Introduction of social networking websites Facebook, Twitter, LinkedIn, Amazon etc, has made community detection in social networks an interesting area of research. Real world systems, such as collaboration networks, the internet, the world-wide-web, biological networks, communication and transport networks, social networks etc. which use interaction among objects can be modeled as networks in order to analyze them. These networks are represented as a graph, where nodes represent the objects and edges represent the interaction between them. Complex network analysis has become an interesting area, and community detection one of the most challenging and popular topics in this area. For analysing these systems, a scalable community detection algorithm needs to be implemented. In social networks, communities are cohesive groups of friends who know each other well within the group and have a few relations outside the group [18]. Networks are composed of community structures. A good community structure has dense connections within the community and a few connections outside the community. The Quality function is a metric to measure the quality of the community partitions. Basically it describes the goodness of the partitions. The most well-known quality function is the Girvan-Newman modularity function [3]. Community detection can be viewed as an optimization problem, in which the quality function is optimized to get more accurate community partitions [16]. The problem of detecting communities in social networks is an NP-hard optimization problem. There are other heuristic based search techniques available to find the optimal value of the optimization function. Genetic algorithms (GAs) and simulated annealing [16] are heuristic-based search techniques. Several algorithms for community detection in social networks with different approaches have been proposed so far. Community structures can be disjoint or overlapping, but in real life overlapping community structures play a very vital role. For example, a professor collaborates with researchers in different fields. Also one person can be a part of multiple groups at a time like group of family members, friends circle and clubs. The overlapping nodes play a very vital role in communication between groups. Overlapping community detection is a challenging task. In this paper, we propose an algorithm to detect overlapping community structures by employing genetic search technique. Our proposed algorithm uses modularity as the quality function. It automatically determines the number of communities without any prior information.

The paper is organized as follows. In the next section, examples of real world networks are described. Section II discusses the related work. Section III describes the new genetic algorithm that we propose including framework, objective function, genetic representation, genetic operators and proposed algorithm. Our algorithm is tested on real data and experimental results are illustrated in Section IV. Finally, Section V concludes the paper.

II. Related work

Many algorithms have been proposed to detect communities in social networks. The most well-known algorithm is the divisive algorithm proposed by Girvan-Newman [1] that iteratively detects the edges that connect vertices of different communities and removes them, so that the communities get disconnected from each other. There are some algorithms proposed for non-overlapping community detection in social networks based on genetic search techniques such as MENSGA [10], MOGA [6], GA-NET [5] and GA for single objective and multi-objective for disjoint community detection by Hafez [16]. All these algorithms give disjoint community structures.

There are methods based on different strategies to find overlapping communities such as clique percolation method (CPM) [2], line graph and line partitioning, local expansion and optimization [12], [13] and label propagation strategy COPRA [11] and SLPA [15]. Among them, CPM is the most famous and widely accepted algorithm with some restrictions. It begins by identifying all cliques of size k in a network. Once these have been identified, a new graph is constructed

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.114

such that each vertex represents one of these k-cliques. Two nodes are connected if the k-cliques that represent them share $k-1$ members. It works very well in dense graphs with small value of k i.e. for small size of the clique. However, it has one drawback in that it fails to terminate in many large complex graphs. Louvain method LM, COPRA and OSLOM are algorithms based on edge-weight strategies. The authors complement information about network topology by weighting each edge, the weight indicating the ability of the edge itself to transferring information. This supplementary source of knowledge is used to find communities. Two methods are introduced to assign weights to the edges of the network, K-path edge centrality and WERW-Kpath algorithm. Fast overlapping community detection algorithm with self-correcting ability [18] uses the modified modularity function as the objective function that is composed of density and cohesion. It is an algorithm which gives overlapping communities self correction using three test conditions. It redistributes the community for unallocated nodes. It runs the algorithm again over specific nodes to detect and correct the error in the algorithm.

Approaches to overlapping community detection based on genetic search technique include GAoCD [14], GA-Net+ [4], OCA [9], CONGA [7], OGA [17] etc. Each one has different representations, objective function, advantages and disadvantages. A method of optimal modularity uses the modularity as the objective function and most of the features of spectral clustering. This is the divisive approach to detect the disjoint partitions but it is not able to get the overlapping communities. OGA is a genetic algorithm with edge-based clustering technique to detect overlapping communities by maximizing the modularity function. This modularity function has been introduced by Shen et al 2009 [8] to support overlapping communities. The proposed restriction to the edge-based representation prevents the possibility of disjoint communities [17]. GaoCD is a genetic algorithm with node-based clustering technique using partition density function as the objective function and not the popular modularity function. The algorithm first finds the link communities by maximizing the objective function which is the partition density D, and then transforms the link communities to node communities based on a novel genotype representation method. It automatically determines the number of communities without any prior information. GA-NET+ is based on node clustering that uses node-based genetic representation. It introduces the concept of community score to measure the quality of partitions in networks, and tries to optimize this quantity by running the genetic algorithm.

III. Methodology for overlapping community detection using disjoint communities

This is a 2 step algorithm in which first we try to find disjoint communities and next these disjoint communities are used to find overlapping communities. It uses node clustering by using node representation. This algorithm uses only nodes in the encoding schema of the genetic representation. Since edges usually represent unique relations among nodes, edges represent genes of an individual. Node clustering discovers the groups of nodes that have similar characteristics. The term *genetic* comes from biological science. It uses the traditional biological operators i.e. selection, crossover, mutation and standard terms like genes, chromosomes (also referred to as individuals). The population is composed of several individuals and each individual is composed of several genes. A search technique is used which yields the fittest individual in each generation. A population of candidate solutions is used to solve an optimization problem which keeps improving as the iterations progress. The candidate solution becomes an approximation of the exact solution. The fittest individual is one which has optimal value of fitness function. fitness function is represented by an objective function described in section III-A. In the following section we will discuss our proposed algorithm including objective function, genetic representation and operators like selection, crossover and mutation.

A. Objective function

The quality function is a quantitative measure for goodness of the partitions. The most widely used quality function to measure the goodness of a community structure is Newman's modularity function shown in equation 1. Generally modularity is a metric of difference between fraction of the number of links inside the community and number of links expected in a network. Networks with high modularity have dense connections inside the community and sparse connections outside it. Modularity given in equation 1 is used as the objective function in our proposed algorithm.

$$Q = \frac{1}{2M} \sum_{i,j \in V} [A_{i,j} - \frac{K_i K_j}{2M}] \delta(c_i, c_j) \tag{1}$$

where M is the total number of edges in the network and $A_{i,j}$ is an element of the network adjacency matrix $A = (A_{i,j})_{n \times n}$. If v_i and v_j are connected by an edge, then $A_{i,j} = 1$, else $A_{i,j} = 0$; c_i and c_j represent the communities to which v_i and v_j belong, respectively; if $c_i = c_j$, then $\delta(c_i, c_j) = 1$, else $\delta(c_i, c_j) = 0$; K_i and K_j are the degrees of v_i and v_j

$$K_i = \sum_{j=1}^{n} A_{i,j} \tag{2}$$

$$K_j = \sum_{i=1}^{n} A_{i,j} \tag{3}$$

where $i, j \in \{1, 2, 3.....n\}$

Note that $|Q| \leq 1$. Higher Q values (close to 1) correspond to stronger community partition of G. Larger the value of Q, better the community structure.

B. Genetic representation

Our algorithm adapts the concept of node clustering by using node representation. It uses only nodes in encoding schema of genetic representation and edge in each gene.

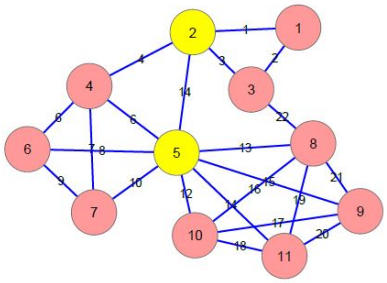

Fig. 1: Network with 3 community structure with two overlapping node 2 and 5

1) Population generation: The population is a collection of individuals and POPSIZE gives the information about number of individuals. In this node-based representation, an individual g of the population consists of n genes $\{g_1, .., g_n\}$, where $k \in \{1, 2, 3, ..., n\}$ is the identifier of a node, n is the number of nodes and each k^{th} entry i.e. g_k takes one of the adjacent edges of node k. We have the population matrix of size POPSIZE$\times$n. According to graph theory two edges are adjacent if they have a common node. An individual g with two different indices i and j may have similar entry i.e. $g_i = g_j$. Table 1 and 2 show two individuals which can be generated from Figure 1. It can be seen from Table II that $g_5 = g_8$.

2	1	3	7	13	9	10	13	17	16	19

TABLE I: $Individual_1$

2	1	22	4	11	9	8	22	15	18	19

TABLE II: $Individual_2$

2) Decoding : In decoding we find communities for an individual, which consists of nodes. A directed graph is formed from this individual. The interpretation of a gene g_k with value e $=< v_k, vj >$ is that node v_k and node vj have one edge in common, and should be classified to same community if both node have reachability from each other. Breadth first search (BFS) is applied on this directed graph to get connected component. Likewise, all the communities are found, and the nodes within the same community constitute a node community.

C. Genetic Operators

1) Crossover : Two individuals are selected randomly from the population to perform crossover. Let us take $individual_1$ and $individual_2$. A partition index is selected at random ranging from 1 to n. After crossover two new individuals are generated, let us say $Child_1$ and $Child_2$. $Child_1$ will have genes from 1 to partition index from $individual_1$ and from partition index to n from $individual_2$. Likewise, $Child_2$ will have genes from 1 to partition index from $individual_2$ and from partition index to n from $individual_1$. Table III and IV shows the two individuals $Child_1$ and $Child_2$ generated from crossover of $individual_1$ and $individual_2$ shown in table 1 and 2. Here, we assume partition index $= 5$. It can be seen that after crossover is carried out over two individuals, the new individuals created are valid strings according to our representation scheme.

2	1	3	7	13	9	8	22	15	18	19

TABLE III: $Child_1$ after crossover

2	1	22	4	11	9	10	13	17	16	19

TABLE IV: $Child_2$ after crossover

2) Mutation : In mutation a gene g_k is selected at random in a random individual and updated by a randomly selected edge $e_i \in \{1, 2, 3, ..., m\}$ with the condition that it should be adjacent to the k^{th} node. For example, the mutation operation on $individual_2$ at index 5 is shown in table V.

2	1	22	4	10	9	8	22	15	18	19

TABLE V: $Individual_2$ after Mutation at index 5

3) Selection: In this operation individuals are arranged in descending order of their fitness values in a manner described below. First, the fitness value of an individual is computed using the modularity function. Then we normalize the fitness value of each individual. An individual is selected for the next generation by deterministic selection based on the normalized fitness value. For example, if the normalized fitness value for an individual is 2.791, then 2 copies of this individual will be passed on to the next generation. The rest of the individuals to make up a population size of POPSIZE are the ones with the highest values in the fractional part of the fitness function.

D. Algorithm to detect disjoint communities

This algorithm takes a graph as an input along with parameters POPSIZE, Pc, Pm, and iteration (or generation) as input and provides disjoint communities as the output. The parameter POPSIZE denotes the number of individuals and each individual is represented by a string of nodes of size n, where n is the number of nodes in the graph. The parameter Pc and Pm represent crossover probability and mutation probability respectively. In this algorithm the graph is represented as an adjacency matrix. The population is initialized as a matrix of size POPSIZE$\times n$, and the initial population is generated with the method described in section III-B1. The next step is to perform crossover (as described in section 4.3.1) and then mutation (as described in section 4.3.2) over the current population. Crossover is only performed when a random number generated between 0 to 1 is less than the crossover probability Pc, and similarly mutation is performed only when a random number again generated between 0 to 1 is less than mutation probability Pm. The Pc and Pm should lie between 0 and 1, Pc should be high and Pm should be

very low because mutations are rarely required in genes. After crossover and mutation we have a population of size two times the population size of the previous generation(iteration) because we add the newly generated individuals in the old population. The next step is to perform selection to extract n fittest individuals out of 2n individuals. The fittest individual of the current generation is compared to the fittest individuals of the previous generation. An unfit individual in population of current generation is replaced with the fittest individual in current population. The three genetic operators of crossover, mutation and selection are repeatedly applied to each generation. On termination, we get the fittest individual that is used to compute disjoint communities using decoding method described in section III-B2. The disjoint communities resulting from the decoding of the fittest individual is the one selected by us.

Algorithm 1 : Disjoint community detection

Inputs: graph, POPSIZE, Pc, Pm, iteration
Output: DC
// n is the ♯ nodes and m is the ♯ edges
//A is the adjacency matrix and DC is the disjoint community matrix
//POPSIZE is the size of population, and Pc,Pm $\in (0, 1)$

Initialize the population as the zero matrix of size POPSIZE$\times n$.
Generate the initial population using method Population Generation
Initialize the fittest individual at random from population
$gen \leftarrow 0$
while gen $<=$ iteration **do**
  $ind \leftarrow 0$
  while ind $<=$ POPSIZE **do**
    Take two individual I_1 and I_2 at random from population
    if $random(0, 1) < Pc$ **then**
      $\{child_1, child_2\} = crossover(I_1, I_2)$
      Population= Population $\cup\{child_1, child_2\}$
    end if
    if random(0,1)$<$ Pm **then**
      $child_1$ = mutation(I_1)
      Population= Population $\cup\{child_1\}$
    end if
    if random(0,1)$<$ Pm **then**
      $child_2$ = mutation(I_2)
      Population = Population $\cup child_2$
    end if
    $ind \leftarrow ind + 2$
  end while
  population = selection(population)
  keep track of fittest individual
  Evaluate the best n individual in terms of fitness from population so that the size is POPSIZE$\times n$
  $gen \leftarrow gen + 1$
end while
DC $\leftarrow$ Decode(fittest individual)
return DC

E. Algorithm to detect overlapping communities

This phase computes overlapping communities using Algorithm 2. This algorithm performs its functionality on disjoint communities obtained in Algorithm 1. First, it extracts a set of boundary nodes, then each boundary node is examined. Internal links and external links of each boundary node are evaluated with respect to each community c. in_{deg} corresponds to internal links (links within community c) and out_{deg} corresponds to external links (links within community c^1, where c$\neq c^1$). $MinMax_{rate}$ as described in equation 4 is the ratio of min(in_{deg}, out_{deg}) and max(in_{deg}, out_{deg}), where min denotes minimum and max denotes maximum between two numbers.

$$MinMax_{rate} = \frac{min(in_{deg}, out_{deg})}{max(in_{deg}, out_{deg})} \tag{4}$$

A node is classified as an overlapping node if its $MinMax_{rate}$ is greater than the overlapping criterion ϵ. $MinMax_{rate}$ gives an intuition about the closeness of in-degree and out-degree of a node with respect to a cluster c, more closeness implies better contribution. Membership defines the number of communities a node is connected to, and it is decided based on this closeness criterion. For example, if membership of a node is two, it means that the number of communities it is connected to is 2.

Algorithm 2 : Overlapping community detection

Inputs: A, DC, M, ϵ
Output: OC
// m is the ♯edges
//A is the adjacency matrix
//DC is the disjoint community matrix
//OC is overlapping community matrix

Find out a set of boundary nodes (BN) from disjoint communities.
Repeat for each boundary node b $\in$ BN
Repeat for each disjoint community DC_i
Find out number of internal links i.e. in_{deg} and external links i.e. out_{deg} of each boundary node b with respect to community DC_i.
in_{deg} = ♯edges within community DC_i
out_{deg} = ♯edges within community DC_j
$MinMax_{rate}$ = $\frac{min(in_{deg}, out_{deg})}{max(in_{deg}, out_{deg})}$
if $MinMax_{rate} \geq \epsilon$ **then**
  The boundary node b is an overlapping node
  OC(b,i) = 1;
  OC(b,j) = 1;
end if
end for
end for
return OC

IV. EXPERIMENTS

In this section we analyze the performance of our algorithm on an artificial network and three real world networks - karate, protein interaction and bottlenose dolphins network. Experiments are performed on a machine with intel(R) core(TM) 2 duo 2.00GHz processor, 2GB RAM

and 160GB hard disk, running a windows 7 32-bit operating system. Our algorithm has been implemented on MATLAB R2013a, and NetMiner 4 tool has been used to analyze the network. We employed standard parameters for the genetic algorithm, crossover probability P_c 0.8, mutation probability P_m 0.2, population size 25 and iterations 1000. The parameter ϵ used in Algorithm 2 denotes the threshold for $MinMax_{rate}$ and is taken as 0.6. The $MinMax_{rate}$ should cross this threshold to qualify for the overlapping criterion. The optimal modularity which corresponds to best partitions, is obtained after completion of 1000 iterations. Algorithm 1 is a hard partitioning algorithm that provides disjoint communities for the network. It is difficult to decide communities for some nodes, thus they are misclassified. Algorithm 2 is applied on these disjoint communities for detecting overlapping nodes.

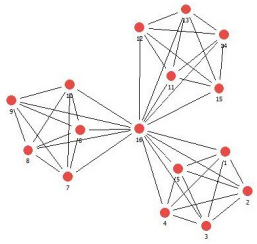

Fig. 2: This an artificial network taken for experiment

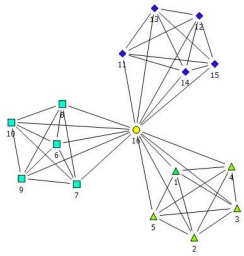

Fig. 3: Resulting network with three community structures, and each one is represented by a different color. It has only one overlapping node, represented by a yellow circle with label 16.

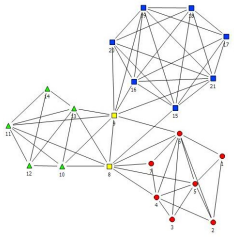

Fig. 4: Community structure obtained by our algorithm in protein-protein interaction network with greatest modularity 0.4817 and corresponds to 3 communities represented by color, and 2 overlapping nodes represented by a yellow color.

A. *Experiment on an artificial dataset*

We test our algorithm on an artificial network (see figure 2) designed by Yanan Cai [14] having 16 nodes and 45 edges. It is difficult to decide to which cluster node 16 belongs because node 16 has equal contribution to all three communities and its $MinMax_{rate}$ is 1.0 which is greater than $\epsilon = 0.6$. It is the only node satisfying the overlapping criteria with membership 3. Figure 3 shows the artificial network in figure 2 which consists of 3 communities with node 16 as the only overlapping node. Each community is represented by a unique color and shape. The overlapping node is represented by a yellow circle.

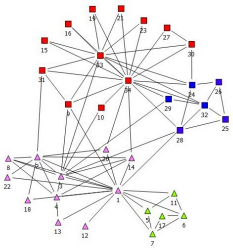

Fig. 5: Community structures obtained by our algorithm in Karate Club network with optimal modularity Q 0.4198 and Q_{ov} 0.4349, and correspond to 4 communities represented by colors and overlapping nodes represented by yellow color

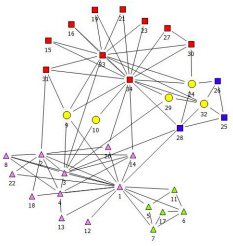

Fig. 6: Communities in karate network are represented by unique colors, and overlapping nodes represented by yellow color.

Node	IN-degree	OUT-degree	MinMax rate	Membership
8	4	4	1.00	2
9	4	4	1.00	2

TABLE VI: Details about overlapping nodes obtained in protein network

B. *Experiments on three real world networks.*

1) Protein interaction network: Protein interaction network is built according to metabolism response relationship between the biological proteins, consisting of 21 nodes and 61 links. It is a typical overlapping community network with three real communities, and having two overlapping nodes. Our algorithm identifies all the three communities, and correctly classifies all the nodes except two. Also our algorithm has

been successful in classifying overlapping nodes {8,9}. Only these nodes are able to qualify for the overlapping criterion. $MinMax_{rate}$ for the nodes 8 and 9 are 1.0 and 1.0 respectively, which is greater than $\epsilon = 0.6$. The $MinMax_{rate}$ 1.0 of a node concludes that node has exactly equal contribution to multiple communities. Resulting community structure for this network is shown in figure 4. Each community is represented by a different color and the overlapping nodes 8 and 9 are represented using yellow color.

Node	IN-degree	OUT-degree	MinMax rate	Membership
9	3	2	0.66	2
10	1	1	1.00	2
24	2	3	0.66	2
29	1	1	1.00	3
32	3	2	0.66	2

TABLE VII: Table shows the details about overlapping nodes obtained in karate club network.

Network	Nodes	Edges	C	OP	Q	Q_{ov}
Protein	21	61	3	0.0952	0.4817	0.4991
Karate	34	78	4	0.1471	0.4198	0.4349
Dolphin	62	159	5	0.1452	0.5285	0.5778
Polbooks	105	441	4	0.0571	0.5266	0.5295
Football	115	613	7	0.11	0.5851	0.6083
Jazz	198	2742	3	0.1263	0.4330	0.4443
Polblogs	1490	9517	905	0.2262	0.1360	0.1891
Net-Science	1589	2742	619	0.0396	0.8464	0.8905
Power	4941	6594	1696	0.3163	0.5685	0.9476

TABLE VIII: Q_{ov} corresponding to best partition occurred by optimal modularity Q for each network where OP is the overlap proportion and C is the number of communities.

Network	Our Algorithm	GA-Net
Protein	1.0	1.0
Karate	0.8910	0.7071
Dolphins	0.8448	0.3400

TABLE IX: Normalized mutual information (NMI) value for our algorithm and GA-NET.

Network	GN	MENSGA	Our Algorithm
Karate	0.4013	0.4198	0.4198
Dolphins	0.4706	0.5272	0.5285
Football	0.5996	0.6041	0.5851
Polbooks	0.5168	0.5262	0.5266
Jazz-Musicians	0.4051	0.4447	0.4389

TABLE X: Modularity(Q) score obtained by GN(Girvan-Newman), MENSGA and our algorithm.

2) Karate Club Network: The Zacharys Karate Club network was generated by Zachary, who studied the friendship of 34 members of a karate club over a period of two years. During this period, because of disagreements, the club was divided into two groups almost of the same size. This network consists of 34 nodes and 78 edges with 2 real communities. Communities found by primary partitions are represented by triangles and squares (see figure 5). Our algorithm further divides the real community structure which is represented using the same shape with different colors.

Figure 5 shows the disjoint community structures with modularity 0.4198, detected by our algorithm 1 for disjoint community detection. Also our algorithm has been successful in detecting a set of overlapping nodes {9,10,24,29,32} by using algorithm 2 over disjoint communities with Q_{ov} 0.4349. Only these nodes match the overlapping criterion. Detailed explanation is given in table VII. The resulting overlapping community structure corresponding to karate network is shown in figure 6. Each community is represented by a different color and each overlapping node is represented by yellow color.

3) Results analysis : Table VIII shows the results after obtaining overlapping communities for a number of networks. It can be seen that if the overlap proportion is small, then the value of Q_{ov} is close to Q. As the proportion of overlap increases, Q_{ov} increases as compared to Q. It is to be noted that if there is no overlap, then Q_{ov} is equal to Q. Figure 7 shows the relationship between Q and Qov. The Q denotes the modularity corresponding to optimal partitions obtained in Algorithm 1. The Q_{ov} denotes the modularity corresponding to overlapping communities obtained in Algorithm 2. It is observed that Q_{ov} has larger value than Q.

Figure 7 concludes that node detected as an overlapping node is not only a part of one community but it also has significant contribution to other communities. Figure 8 shows the convergence of modularity Q. We observe that modularity converges to its optimal value after around 600 iterations for each dataset. This optimal value of modularity provides disjoint communities that are used in overlapping community detection algorithm. Table IX shows the comparison of our algorithm to GA-NET [5] in terms of normalized mutual information (NMI). The algorithm GA-NET consists of a parameter alpha that is taken as 1.5. Both the algorithms are run 10 times to get average NMI value. Figure 10 shows that our algorithm achieves better performance than GA-NET. The normalized mutual information is described in equation 5. The Normalized Mutual Information is a similarity measure. Let us consider two partitions A and B of a network that are composed of communities, let us say C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that are also in the community j of the partition B.

$$I(A,B) = \frac{-2(\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} C_{ij} log(\frac{C_{ij}N}{C_{i.}C_{.j}})}{\sum_{i=1}^{c_A} Ci.log(\frac{C_{i.}}{N}) + \sum_{j=1}^{c_B} C_{.j} log(\frac{C_{.j}}{N})} \quad (5)$$

where c_B and c_A are the number of communities in the partition B and A respectively, $C_{i.}$ and $C_{.j}$ are the sum of the elements of C in row i and column j respectively, and N is the number of nodes. If $A = B$, then $I(A,B) = 1$ else $I(A,B) = 0$.

Table X and figure 9 show that our algorithm works better than the other algorithms in most cases. In the other cases, our results are comparable with results of the other algorithm

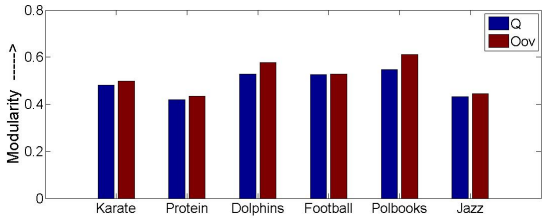

Fig. 7: Relationship between Q and Qov in different dataset.

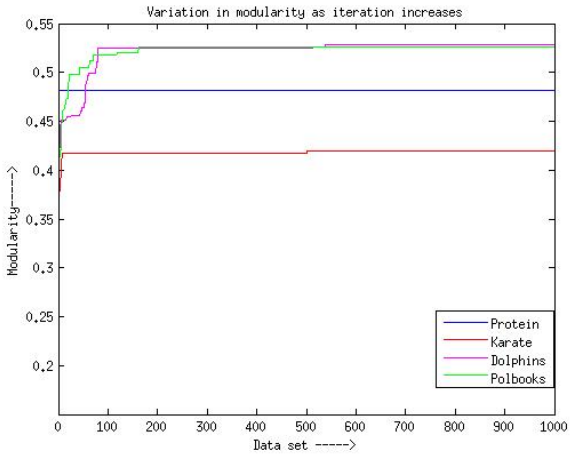

Fig. 8: Modularity occurred at each iteration in each dataset which is represented by different color. Modularity achieved its optimal value after a certain iteration.

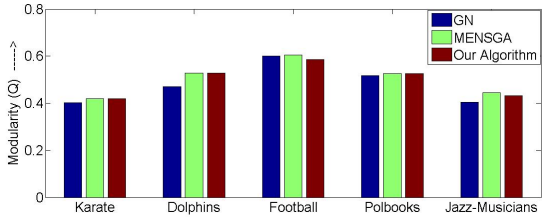

Fig. 9: Modularity(Q) score of GN(Girvan-Newman), MENSGA and our algorithm in different dataset.

V. Conclusion

In this work we present an algorithm for detecting overlapping community structures. Our algorithm depends on nodes for genetic representation but the edges associated with the nodes are used in representation. Our algorithm for disjoint communities is working well and giving good results. The disjoint community structure is then used to find the overlapping communities. Whenever the overlapping community structure

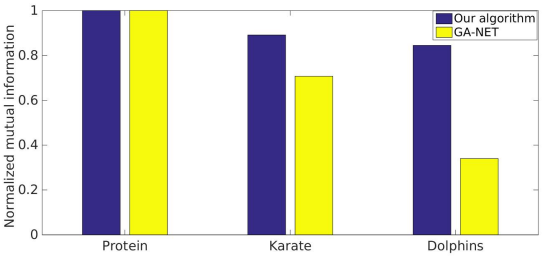

Fig. 10: Comparison of our algorithm to GA-NET with respect to NMI value.

is better, Qov is higher than the modularity Q obtained for disjoint datasets. The normalized mutual information(NMI) and modularity have been used to compare our method with other GA based algorithms and in most cases, our algorithm is giving good results.

The non-overlapping community detection gives good non-overlapping communities. Since only the boundary nodes need to be checked, checking the boundary nodes on whether they can belong to more than one community is a good way of dealing with overlapping communities.

References

[1] M E Newman and M Girvan. Finding and evaluating community structure in networks, *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 69, no. 2, pages. 026113.1–15, 2004

[2] Palla and Gergely. Uncovering the overlapping community structure of complex networks in nature and society, *Nature*, vol. 435, no. 7043, pages. 814–818, 2005

[3] M E Newman. Modularity and community structure in networks, *Proc Natl Acad Sci U S A*, vol. 103, no. 23, pages. 8577–8582, 2006

[4] Clara Pizzuti. Overlapped Community Detection in Complex Networks, *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM*, no. 8, pages. 859–866, 2009

[5] Clara Pizzuti. GA-Net: A Genetic Algorithm for Community Detection in Social Networks, *Proc. of the* 10^{th} *International Conference on Parallel Problem Solving from Nature*, vol. 5199, pages. 1081–1090, 2008

[6] Clara Pizzuti. A Multi-objective Genetic Algorithm for Community Detection in Networks, *ICTAI, IEEE Computer Society*, pages. 379–386, 2009

[7] Steve Gregory. An Algorithm to Find Overlapping Community Structure in Networks, *Springer-Verlag*, no. 12, pages. 91–102, 2007

[8] Huawei Shen, Xueqi Cheng and Kai Cai. Detect overlapping and hierarchical community structure in networks, *CoRR*, vol. abs/0810.3093, 2008

[9] Arnau Padrol-Sureda, Guillem Perarnau-Llobet, Julian Pfeie, and Victor MuntAl's-Mulero. Overlapping Community Search for social networks, *Data Engineering (ICDE), 2010 IEEE 26th International Conference on* , pages. 992–995, 2010

[10] LI Yun, LIU Gang and LAO Song-yang. A genetic algorithm for community detection in complex networks, *Journal of Central South University, Springer*, 2013

[11] S. Gregory. Finding overlapping communities in networks by label propagation, *Arxiv preprint arXiv:0910.5516*, 2009

[12] Andrea Lancichinetti and Santo Fortunato. Detecting the overlapping and hierarchical community structure in complex networks, *New J. Phys. p*. 2009

[13] Lancichinetti and Santo Fortunato. Finding Statistically Significant Communities in Networks, *PLoS ONE*, Public Library of Science, vol. 6, no. 4, pages. e18961, 2011

[14] Yanan Cai and Chuan Shi. A Novel Genetic Algorithm for Overlapping Community Detection, *Springer*, vol. 7120, pages. 97-108, 2011

[15] Xie, Jierui and Szymanski, Boleslaw K. and Liu, Xiaoming. SLPA: Uncovering Overlapping Communities in Social Networks via A Speaker-listener Interaction Dynamic Process, *CoRR*, vol. abs/1109.5720, 2011

[16] Hafez, Ahmed Ibrahem. Genetic Algorithms for community detection in social networks, *Intelligent Systems Design and Applications (ISDA), 12th International Conference on IEEE*, pages. 460-465, 2012

[17] Brian Dickinson, Benjamin Valyou and Wei Hu. A Genetic Algorithm for Identifying Overlapping Communities in Social Networks Using an Optimized, *Scientific Research* , 10.4236/sn.2013. 2013

[18] Laizhong Cui, Lei Qin, and Nan Lu. A Fast OverLapping Community Detection Algorithm with Self-Correcting Ability, *The Scientific World Journal*, vol. 2014, Article ID 738206, 2014

[19] Pablo M. Gleiser and Leon Danon. Community Structure in Jazz, *Advances in Complex Systems*, vol. 6, no. 4, pages. 565–573, 2003

[20] V. Krebs. Books about US Politics. 2004

[21] W. W. Zachary. An information flow model for conflict and fission in small groups, *Journal of Anthropological Research*, vol. 33, pages. 452–473, 1977

[22] Girvan and Newman. Community structure in social and biological networks, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pages. 7821–7826, 2002

[23] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks, *Nature*, volume. 393, pages. 440–442, 1998

[24] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations, *Springer-Verlag, Behav. Ecol. Sociobiol*, vol. 54, no. 4, pages. 369–405, 2003