An EDA-based Community Detection in Complex Networks

Mohsen Ghassemi Parsa, Nasser Mozayani, Ahmad Esmaeili
Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran
m_parsa@comp.iust.ac.ir, mozayani@iust.ac.ir, aesmaeili@iust.ac.ir

***Abstract*—Communities are basic units of complex networks and understanding of their structure help us to understand the structure of a network. Communities are groups of nodes that have many links inside and few links outside them. Community detection in a network can be modeled as an optimization problem. We can use some measures such as Modularity and Community Score for evaluating the quality of a partition of nodes. In this paper, we present a new algorithm for detecting communities in networks based on an Estimation of Distribution Algorithm (EDA) with the assumption that the problem variables are independent. EDAs are those evolutionary algorithms that build and sample the probabilistic models of selected solutions instead of using crossover and mutation operators. In this paper, we assess our algorithm by synthetic and real data sets and compare it with other community detection algorithms.**

Keywords—complex networks; community detection; estimation of distribution algorithm; graph mining

I. Introduction

The community detection is one of the major problems in the analysis of complex networks. By Community detection, we attempt to determine groups of nodes that are dense according to links that exist between them.

There are many community detection algorithms in complex networks. These algorithms can be found in different categories such as methods based on graph partitioning, hierarchical clustering algorithms, agglomerative/divisive algorithms, optimization of an objective function, evolutionary-based algorithms, and other categories which are fully described in [1]. Some of the related works are discussed as follows. Girvan-Newman (GN) algorithm [2] removes links according to the Betweenness values until the Modularity (Q) measure reaches its maximum value. Infomap [3] attempts to optimally compress the information of the graph to detect communities. Clauset-Newman-Moor (CNM) [4] begins with nodes of network without links and then starts to add the links according to the modularity measure. Louvain algorithm [5] is based on a local optimization of modularity. GA-Net [6] uses a genetic algorithm with Community Score (CS) objective function to detect communities. Li and Song [7] use Extended Compact Genetic Algorithm (ECGA) to detect communities. ECGA is a kind of estimation of distribution algorithm in which variables are divided into independent cluster.

Estimation of Distribution Algorithms (EDAs) are those evolutionary algorithms that build and sample the probabilistic models of selected solutions [8]. In other words, instead of applying genetic operators such as crossover and mutation, EDAs build the model of selected individuals of the current population and sample to form a new population. EDAs are categorized into various algorithms based on whether the problem variables are independent or not, the way genomes are represented, and the probabilistic model (e.g. probability vector, Markov, Bayesian network, tree) that [8] describes them.

In this paper, for detecting communities in complex networks we use a Univariate Marginal Distribution Algorithm (UMDA) [9] and extend it to integer representation. UMDA is a univariate estimation of distribution algorithm (EDA) that uses probability vectors as the probabilistic model with the assumption that the problem variables are independent. This assumption results in the simplicity of the approach and removes the computational overhead of linkage learning. We assess our algorithm using some synthetic and real data sets and the experimental results are compared with other algorithms.

The main advantage of our algorithm compared to the greedy approaches such as CNM [4] is broader search of the space of all possible partitions of a network by an evolutionary algorithm. Because of using an EDA algorithm to detect communities, our algorithm converges more quickly compared to GA-based algorithm such as GA-Net [6]. In addition, our algorithm does not have the overhead of linkage learning because the problem variables are assumed independent of each other unlike the algorithm of Li and Song[7] that is based on ECGA.

This paper is organized as follows: Section 2 defines the problem of community detection as an optimization problem and describes two measures for evaluating the quality of a partition of nodes. Section 3 describes our algorithms in detail. Our experimental results are presented in section 4, and Section 5 concludes the paper and explains the future works.

II. Problem definition

We can represent the structure of a network with a graph in which nodes represent entities and links show interaction between them. Community detection of a network means

978-1-4799-5359-2/14/$31.00 ©2014 IEEE

clustering nodes in groups with dense intra-cluster and sparse inter-cluster connection. Fig. 1 shows an example of a network with three communities. Community detection can be modeled as an optimization problem with an objective function such as modularity. The resulting optimization problem is NP-complete [10] and evolutionary algorithms are often useful to solve these problems. In the remainder of this section, we describe modularity and community score measures as objective functions.

A. Modularity

Newman and Girvan have introduced modularity (Q) in [2] as a criterion for stopping of removing links. Modularity is a proper criterion for the quality of a partition. We adopt the definition from [4].

The elements of the adjacency matrix of the network are denoted as A_{vw}. If nodes v and w are connected $A_{vw} = 1$ and $A_{vw} = 0$ otherwise. Let c_v be the community of node v. The degree of a node v is defined as follows:

$$k_v = \sum_w A_{vw} \tag{1}$$

And modularity is defined as follows:

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \tag{2}$$

If $i = j$ then the δ-function $\delta(i,j)$ is 1 and 0 otherwise, and the number of links in the network is $m = \frac{1}{2} \sum_{vw} A_{vw}$. Modularity (Q) can be rewritten as

$$Q = \sum_{i=1}^{n_c} (e_{ii} - a_i^2) \tag{3}$$

where, n_c is the number of communities; and e_{ii}, a_i are defined as follows:

$$e_{ii} = \frac{1}{2m} \sum_{vw} A_{vw}\, \delta(c_v\, , c_w) \tag{4}$$

$$a_i = \frac{1}{2m} \sum_{v} k_v \;\; \delta(c_v\, , i) \tag{5}$$

B. Community Score

Community score (CS) was defined in [6]. Let $S = (I\, , J)$ be sub-matrix of adjacency matrix A, where I is a subset of the rows $X = \{I_1, I_2, \dots, I_n\}$ of the matrix A, and J is a subset of the columns $Y = \{J_1, J_2, \dots, J_n\}$ of the matrix A. Let a_{iJ} denote the mean value of the i_{th} row of sub-matrix S, and a_{Ij} the mean of the j_{th} column of sub-matrix S. These quantities can be defined more formally as

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij} \quad , \quad a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij} \tag{6}$$

The volume v_s is the total number of links in the sub-matrix $S = (I\, , J)$, and it is defined as

$$V_s = \sum_{i \in I\, , j \in J} a_{ij} \tag{7}$$

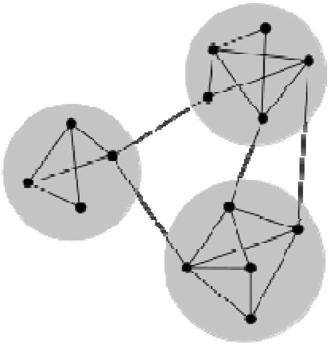

Fig. 1. An example of a network with 3 communities

The power mean of S of order r, denoted as $M(s)$, is defined as follows:

$$M(s) = \frac{\sum_{i \in I} (a_{iJ})^r}{|I|} \tag{8}$$

Moreover, the score of S is defined as below

$$Q(s) = M(S) * v_s \tag{9}$$

Finally, given a partitioning $\{S_1, S_2, \dots, S_k\}$ of A, the community score of it is defined as

$$CS = \sum_{i}^{k} Q(S_i) \tag{10}$$

Community detection problem in a complex network can be formulated as the problem of maximizing the CS objective function.

III. The Proposed Method

In this section, we present our algorithm in detail, by giving genome representation, objective functions, initialization, and its process.

A. Representation

We use locus-based adjacency representation, that is proposed in [11] for clustering problems, and our representation is same as GA-net [6]. Let the size of an individual be N and each gene is associated with a node in the network. Also in this representation, an individual represents a graph. If gene i gets the value $j \in \{1, \dots, N\}$, there is a link between nodes i and j (in the corresponding graph of the individual), and it is interpreted as nodes i and j are in the same community.

A decoding step is necessary to specify all the connected components of the corresponding graph of an individual and it can be performed in linear time. In this representation, we do not need to specify the number of communities. Fig. 2(a) shows a network graph and in (b) there is a genotype and the corresponding graph of the genotype is shown in (c).

B. Objective function

We use both modularity and community score as the objective functions.

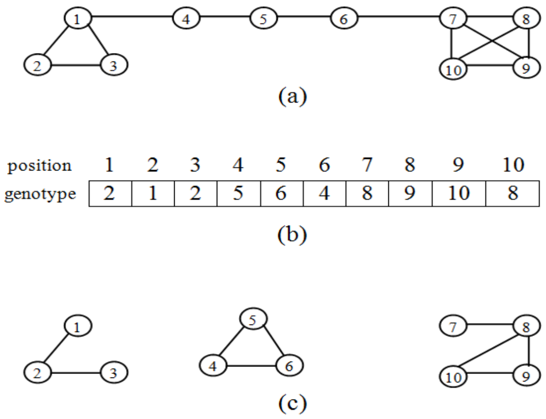

Fig. 2. (a) Original network; (b) Genotype of a partitioning; (c) The corresponding graph of the genotype that the components are communities

C. Initialization

For initializing the population, the value of each gene of an individual can only be the label of its corresponding node in the network, or the labels of its neighbors of the corresponding node. For example in the network of Fig. 2(a) the 7th position of genotype can only get 7 or one of the values 6, 8, 9 or 10.

D. The proposed algorithm

For detecting communities in complex networks, we use an UMDA with integer representation. UMDA is a univariate EDA that uses probability vectors as the probabilistic model with the assumption that the problem variables are independent. In each generation, some individual of the current population are selected and the probability vector of each gene is computed. This is done according to different values of the selected individuals in the gene. The probability vectors are considered as a probabilistic model of the selected individual in current population and the next population is built from sampling of it. Fig. 3 shows a generation of our approach. The algorithm is presented more formally in algorithm 1.

```
Algorithm 1: EDA-based community detection
1- Initialize population: the value of gene $i$ can be $i$ or one of the
   neighbors of the node $i$ in the network
2- Evaluate individuals of the population
3- Perform tournament selection to select $n$ individual from the
   population
4- Compute probability vector of each gene, according to the value of
   the genes in the selected individuals
5- Sample from the distributions (according to probability vectors)
6- Mutate samples with a low probability $P_m$
7- Create a new population from the output of step 6 and the best
   individual of last population (elitism selection)
8- Stop the algorithm if the best individual of the population has not
   changed during $g$ last generations, go to step 2 otherwise
```

For simplicity, step 6 and 7 are not included in the Fig. 3 and because of avoiding premature convergence, in step 6 we offer a mutation-like operator. By increasing P_m, the ability of the algorithm to explore the search space is increased.

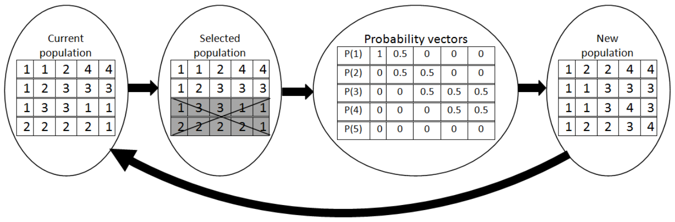

Fig. 3. One generation of our algorithm

IV. Experiments

In this section, the proposed algorithm is assessed and compared with other community detection algorithms. We use the library of SNAP[1] for executing Girvan-Newman (GN) [12], Clauset-Newman-Moore (CNM) [4], and Infomap [3] algorithms.

We use the Normalized Mutual Information (NMI) to evaluate the results [13]. Let *A* and *B* be two partitions of a network and **N** be the confusion matrix, where the rows correspond to the partition *A* and the columns correspond to the partition *B*. The element of **N**, N_{ij}, is the number of nodes of the community *i* of the partition *A* that appear in the community *j* of the partition *B*. The normalized mutual information that is based on information theory is defined as follows:

$$I(A,B)=\frac{-2\sum_{i=1}^{cA}\sum_{j=1}^{cB}N_{ij}\log\left(\frac{N_{ij}N}{N_{i.}N_{.j}}\right)}{\sum_{i=1}^{cA}N_{i.}\log\left(\frac{N_{i.}}{N}\right)+\sum_{j=1}^{cB}N_{.j}\log\left(\frac{N_{.j}}{N}\right)} \qquad (11)$$

where *N* is the number of nodes, c_A is the number communities in the partition *A*, and c_B is the number communities in the partition *B*. $N_{i.}$ represents the sum over row *I*, and $N_{.j}$ expresses the sum over column *j* of matrix **N**. *A* is considered as actual partition and *B* as detected partition. The value of *I(A,B)* is between 0 to 1. If the detected communities and actual communities are equal, then *I(A,B)* takes the value 1. If the detected communities are entirely independent of the actual communities, then *I(A,B)* takes the value 0.

All of the experiments have been performed on a machine with Core2 Duo 1.83GHz processor and 2GB RAM. We use two synthetic and three real data sets with ground truth communities to assess the quality of our approach. Both modularity and community score (with *r* =1.5) are used in the experiments. The parameters of our algorithm are set as follows. The population size is 300, the probability of mutation $P_m = 0.02$, tournament size is 10, the number of selected members for computing probability vectors is 50, and the algorithm will stop if the best individual of the population has not changed during *g*=100 last generations. We denote our algorithm with modularity objective function by Ours-Q and with community score by Ours-CS. For each data set, we run our algorithm 10 times and the average NMI of these runs is reported.

A. Synthetic data set

We use Girvan-Newman[12] and LFR[14] data set in order to assess our approach. Girvan-Newman(GN) data set contains

[1] http://snap.stanford.edu/snap/index.html

128 nodes that fall into four communities, and each community contains 32 nodes and the expected degree of each node is 16. Because of the easiness of the GN benchmark, we also use LFR benchmark that uses power law distributions of degree and community size. We generate undirected and unweighted LFR benchmark graphs without overlapping communities. The parameters of the LFR graphs are set as follows. The number of nodes N = 1000, the average degree of nodes k = 20, the maximum degree of nodes $maxk$ = 50, the minus exponent for the degree sequence t_1 = 2 and the minus exponent for community size distribution t_2 = 1. We should determine a parameter named Mixing Coefficient (μ) for generating these data sets. This parameter specifies the proportion of the number of links of a node to other communities and the degree of the node. The larger the mixing coefficient the more difficult to detect communities on the GN or LFR benchmarks.

Fig. 4 shows the results of the algorithms on the GN data set. Our approach with Q objective function (Ours-Q) is better than the others when $\mu \leq 0.5$. Our approach with CS objective function (Ours-CS) does not show satisfactory results. The quality of Our-CS is sensitive to the r parameter of the community score. By increasing the value of r, the algorithm will tend to increase the number of communities, while there are only four communities on GN data set. The effect of r parameter is studied in [15].

As can be seen in Fig. 5, our algorithms have better NMI than CNM algorithm on the LFR data set. Since the number of communities is high (more than thirty), Ours-CS has satisfactory results.

B. Real data sets

We use American Football, Political Books and Dolphins data sets. The American Football data set contains the games between some colleges, as compiled by Girvan and Newman. The communities of the nodes indicate to which conferences they belong [12]. This network contains 115 nodes and 616 links in 12 communities. On the political books data set, nodes represent 105 books about US politics sold by the online bookseller Amazon.com. The Links represent frequent co-purchasing of the books by the same buyers. Nodes are divided into 3 communities that indicate whether books are liberal, neutral, or conservative. These divisions were assigned separately by Mark Newman [16]. The last real data set is Dolphins that contains a social network of frequent associations between 62 dolphins as compiled by Lusseau et al [17]. The nodes are divided into 2 communities.

Fig. 6 shows the results of applying different algorithms to detect communities on three real data sets. Ours-Q is better than other algorithms on Dolphins and Political Books and is very close to the best algorithm (here GN) on American Football data set. Except Football data set, Ours-CS does not gain satisfactory NMI on other two data sets. Since the r parameter in the community score objective function is set to 1.5, again Ours-CS has a tendency to increase the number of communities. On the Football data set, which includes 12 communities Ours-CS has an excellent result and on the Political Book with three communities and on the Dolphins with two communities, Ours-CS does not have satisfactory results.

V. Conclusion and Future Works

Community detection is an important problem in complex networks. Communities are basic units of complex networks and understanding of their structure help us to understand the structure of a network. For community detection, we use an UMDA and extend it to integer representation with both modularity and community score objective. We assessed our approach by real and synthetic networks and compared with some other approaches. The main advantage of our algorithm compared to the greedy approaches such as CNM is broader search of the space of all possible partitions of a network. Moreover, the rate of convergence of our algorithm is higher than GA-based algorithms. Since the problem variables in our algorithm are independent, our approach does not have the overhead of linkage learning. The experiments demonstrate the ability of our algorithm. In the future, we will attempt to detect overlapping and dynamic communities. Furthermore, community detection with other kinds of estimation of distribution algorithms will be our future research.

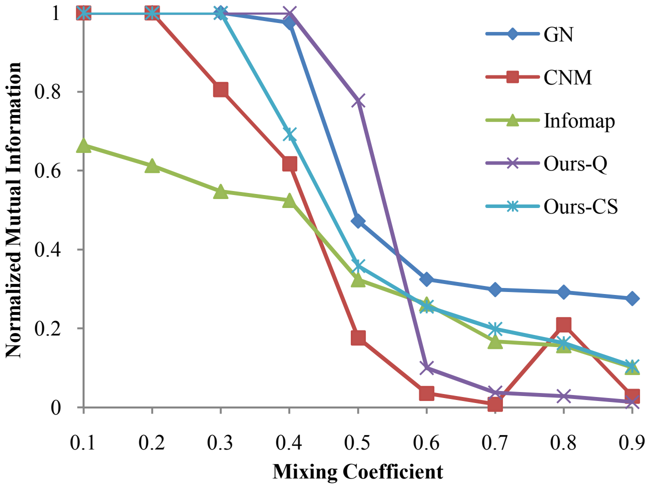

Fig. 4. The results on GN benchmark

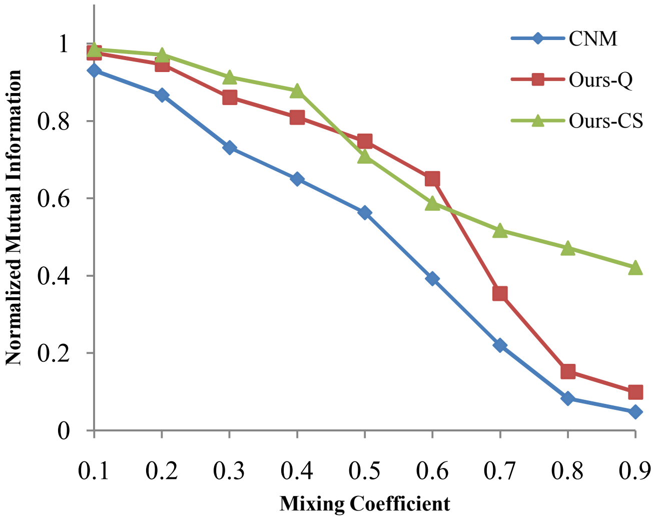

Fig. 5. The results on LFR Bechmark

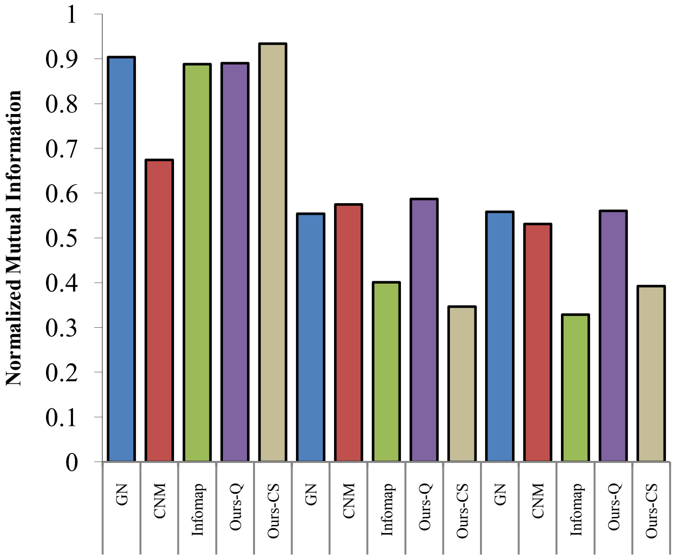

Fig. 6. The results on three real benchmarks

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics Reports,* vol. 486, pp. 75-174, 2010.

[2] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E,* vol. 69, p. 026113, 2004.

[3] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences,* vol. 105, pp. 1118-1123, 2008.

[4] A. Clauset, *et al.*, "Finding community structure in very large networks," *Physical review E,* vol. 70, p. 066111, 2004.

[5] V. D. Blondel, *et al.*, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2008, p. P10008, 2008.

[6] C. Pizzuti, "GA-Net: A Genetic Algorithm for Community Detection in Social Networks," in *Parallel Problem Solving from Nature – PPSN X.* vol. 5199, G. Rudolph, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 1081-1090.

[7] J. Li and Y. Song, "Community detection in complex networks using extended compact genetic algorithm," *Soft Computing,* vol. 17, pp. 925-937, 2013/06/01 2013.

[8] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation,* vol. 1, pp. 111-128, 2011.

[9] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature — PPSN IV.* vol. 1141, H.-M. Voigt, *et al.*, Eds., ed: Springer Berlin Heidelberg, 1996, pp. 178-187.

[10] B. Karrer, *et al.*, "Robustness of community structure in networks," *Physical review E,* vol. 77, p. 046119, 2008.

[11] Y. Park and M. Song, "A genetic algorithm for clustering problems," in *Proceedings of the Third Annual Conference on Genetic Programming*, 1998, pp. 568-575.

[12] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences,* vol. 99, pp. 7821-7826, 2002.

[13] L. Danon, *et al.*, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment,* vol. 2005, p. P09008, 2005.

[14] A. Lancichinetti, *et al.*, "Benchmark graphs for testing community detection algorithms," *Physical review E,* vol. 78, p. 046110, 2008.

[15] C. Pizzuti, "Mesoscopic analysis of networks with genetic algorithms," *World Wide Web,* vol. 16, pp. 545-565, 2013/11/01 2013.

[16] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences,* vol. 103, pp. 8577-8582, 2006.

[17] D. Lusseau, *et al.*, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology,* vol. 54, pp. 396-405, 2003.