A multi objective Jaya optimization based approach for community detection in complex networks

Report submitted in fulfillment of the requirements for the B.Tech Project of

Fourth Year B.Tech.

by

Vempalli Mugenna Gari Madhava Reddy
Sai Kiran Anumalla

Under the guidance of

Dr.Amrita Chaturvedi

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
Dec 2020

Dedicated to

My parents, teachers,.....

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi

Date:1.12.2020

Vempalli Mugenna Gari Madhava Reddy
Sai Kiran Anumalla
B.Tech Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**A multi objective Jaya optimization based approach for community detection in complex networks**" being submitted by **Vempalli Mugenne Gari Madhava Reddy** and **Sai Kiran Anumalla** (**Roll No. 17075056,17075051**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of my supervision.*

Place: IIT (BHU) Varanasi
Date:1.12.2020

Dr.Amrita Chaturvedi
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and our institute. We would like to extend our sincere thanks to all of them. We are highly indebted to our project supervisor Dr. Amrita Chaturvedi for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project.

Place: IIT (BHU) Varanasi **Vempalli Mugenna Gari Madhava Reddy**

Date: 1.12.2020 **Sai Kiran Anumalla**

Abstract

Community detection in complex network has become a vital step to understand the structure and dynamics of networks in various fields. However, traditional node clustering and relatively new proposed link clustering methods have inherent drawbacks to discover overlapping communities. Node clustering is inadequate to capture the pervasive overlaps, while link clustering is often criticized due to the high computational cost and ambiguous definition of communities. So, community detection is still a formidable challenge. In this work, we implement Jaya optimisation based approach for community detection. We use the attributes of the nodes in graph to determine the community based on their gaussian similarity and the new jaya method for optimising the communities.

Contents

Chapter 1

Introduction

In order to have a better understanding of organizations and functions in real-world networked systems, the community structure in the graph is a primary feature that should be taken into consideration. As a result, community detection, which can extract specific structures from complex networks, has attracted considerable attention crossing many areas from physics, biology, and economics to sociology , where systems are often represented as graphs. Generally, a community in a network is a sub-graph whose nodes are densely connected within itself but sparsely connected with the rest of the network.

1.1 Need for multi objective approach

Recently, significant progress has been achieved in this research field and several popular algorithms for community detection have been presented. One of the most popular type of classical methods partitions networks by optimizing some criteria. Newman and Girvan[1] proposed a network modularity measure (usually denoted by Q) and several algorithms that try to maximize Q have been designed. But recent researches have found that the modularity based algorithms could not detect communities smaller than a certain size. This problem is famously known as the resolution limit. The single optimization criteria, i.e., modularity, may not be adequate to represent the structures in complex networks, thus we employ a new community detection process as a multi-objective optimization problem.

1.2 Similarity based approach

Another family of approaches considers hierarchical clustering techniques. It merges or splits clusters according to a topological measure of similarity between the nodes and tries to build a hierarchical tree of partitions[2]. Also there are some ways, such as spectral methods and signal process method, to map topological relationship of nodes in the graphs into geometrical structures of vectors in n-dimensional Euclidean space, where classical clustering methods like classical C-Means (CM), Fuzzy C-Means (FCM) or Evidential C-Means (ECM) could be evoked. However, there must be some loss of information during the mapping process. Besides, these prototype-based partition methods themselves are sensitive to the initial seeds.

For social networks with good community structures, the center of one group is likely to be one person, who plays the leader role in the community. That is to say, one of the members in the group is better to be selected as the seed, rather than the center of all the objects. To solve these problems, Jiang et al. proposed an efficient algorithm named K-rank which selects the node with the highest centrality value as the prototype. In our previous work, an evidential centrality measure is used to set one"most possible"object in the class to be the prototype. We believe that the characteristic on the prototype of each community is important for community detection. However, in some cases the way of using only one node to describe a community may not be sufficient enough,but we employ only a prototype or leader node for a network to work on with to get communities.

1.3 Evolutionary Algorithms

Evolutionary Algorithms In the last decade, a variety of community detection methods have been proposed. Among these algorithms, evolutionary algorithms (EAs) have shown the competitive performance in finding the communities. The first MOEA(Multi Objective EAs) for community detection in complex networks is MOGA-Net where two objectives, community score and community fitness, were simultaneously optimized. Experiments on synthetic and real-world networks demonstrated the superiority of MOGA-Net, when compared with single-objective EA commu- nity detection algorithms[3]. Since then, a number of MOEAs have been suggested for the community detection in complex net- work. For example a PESA-II based MOEA for community detection, termed MOCD. The MOEAs have demonstrated their competitive performance in solving the community detection problem. In this paper we try to implement Jaya Optmization algorithm to get maximum modularity for the communities.

Chapter 2

Preliminaries and Related Work

2.1 Community detection problem

We only consider undirected and unweighted network, which can be typically represented as a graph: G = (V, E), where V = v1, v2,. . ., vn is the set of nodes and E = (i,j)—vi in V, vj in V,i =/ j is the set of links.

In a complex network G, a community refers to the group of nodes such that links between nodes in the group are dense, whereas links between nodes in different groups are sparse. Let C = C1, C2, . . ., Ck be a set of all communities in G, and thus Ci,i(1 to k), needs to satisfy the following constraints: Ci is subset of V and Ci is not null. Ci != Cj, for all i != j and i,j 1, 2,. . ., k i=1 Ci = V[3] In this paper, we only focus on the non-overlapping community detection, which means each pair of communities satisfies: Ci intersection Cj = nullset, for all i != j and i,j 1, 2,. . ., k

2.2 Related Research

The first work on MOEA based community detection algorithms, was suggested by Pizzuti et al. [4] ,termed MOGA-Net. In this algorithm, the community detection was modeled as a bi-objective optimization problem and the fast elitist non-dominated sorting genetic algorithm (NSGA-II) was adopted to solve the problem. The two objectives used in MOGA-Net were community score (CS) and community fitness (CF), respectively.

The experimental results on the synthetic and real-world complex networks showed that when compared with single-objective EA based community detection algorithms, MOGA-Net can discover more accurate community structure. Due to the competitive performance of MOGA-Net, a variety of promising MOEAs have been developed to explore their potential in improving the quality of community detection.Shi et al. proposed a PESA-II[5] based multi-objective commu- nity detection algorithm,termed MOCD, where two-phase strategy was adopted. In the first phase, MOCD optimized two complemen- tary objectives (intra-link and inter-link), then returned a set of Pareto optimal solutions. In the second phase, two model selection methods were employed to select more preferable solution from the Pareto optimal set.

In recognizing the superiority of MOEA/D-Net, more decomposition based MOEAs for community detection have been suggested. For example, Gong et al. further presented a MOEA community detection algorithm based on discrete particle swarm optimization, termed MODPSO. The main difference between MOEA/D-Net and MODPSO lies in the fact that MOEA/D-Net used a genetic algorithm to optimize sub-problems, whileMODPSOmadeuse ofdiscretepar- ticle swarmoptimization.Recently,two other decompositionbased MOEAs, namely, MODTLBO/D [29] and MODBSA/D, were pro- posed for community detection, where a discrete teaching-learningbased optimization algorithm (DTLBO) and a discrete backtracking search optimization algorithm (DBSA) were used as the optimization algorithms.

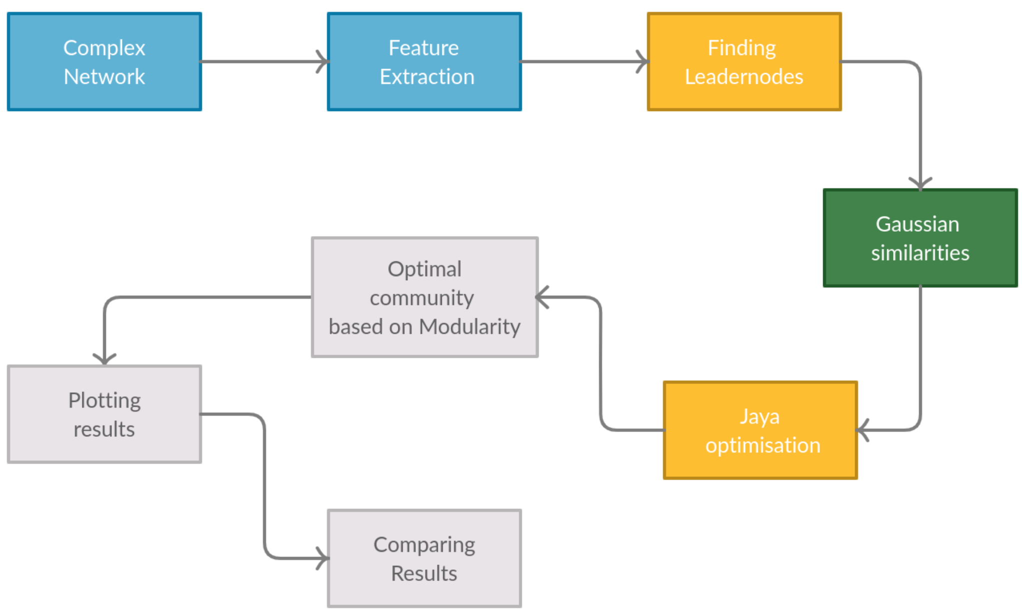
To improve the performance, a problem-specific population initialization method and a turbulence operator were also suggested in, which made them obtain better community structure than existing MOEAs for community detection.There also exist some other MOEA based community detec- tion algorithms, which were developed based on different swarm intelligence algorithms. For example,the modified harmony search algorithm with a chaotic local search for community detection (HSA-CLS), the community detection algorithm based on an enhanced firefly algorithm (EFA).In addition, there are some interesting MOEAs which focused on obtaining better community partition by optimizing new criteria, such as the bi-objective com- munity detection algorithm (BOCD) that optimized modularity Q and CS, the multi-objective immune algorithm based commu- nity detection algorithm (MICD) whose objectives were modified ratio association (MRA) and RA [3].

The MOEAs mentioned above have demonstrated their competitiveness in a variety of complex networks, and in this paper we will peek into this research line by implementing Jaya's EA for optimization.

Chapter 3

Methodology

3.1 Workflow



3.2 Feature Extraction

Initially we extract the following features for each node in our complex network and store these feature vectors in a matrix for further processing.

- Clustering
- Betweenness centrality
- No. of neighbours
- Avg neighbor degree
- Closeness centrality
- Degree centrality
- Load centrality
- Harmonic centrality
- Page rank

3.3 Node similarity

Here initially we find a leader node for a complex network which describes the most highly connected node. In order to acheive this, we find eigen-vector centrality of each and every node, and find the node with maximum eigen-vector centrality. we then consider this node as our leader node.

For a given graph G:=(V,E) with V vertices let A = (av,t) be the adjacency matrix, i.e. av,t = 1 if vertex v is linked to vertex t, and av,t = 0 otherwise. The relative centrality x, score of vertex v can be defined as:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$$

where M(v) is a set of the neighbors of v and lambda is a constant. With a small rearrangement this can be rewritten in vector notation as the eigen-vector equation

Ax=(lambda)x;

In general, there will be many different eigenvalues lambda for which a non-zero eigenvector solution exists. However, the additional requirement that all the entries in the eigenvector be non-negative implies (by the Perron–Frobenius theorem) that only the greatest eigenvalue results in the desired centrality measure

After finding our leader node for a network, we consider finding similarity for every node corresponding to the discovered leader node and store those results in a matrix file for further processing. Here we specially use gaussian similarity as ingredient for the similarity measure between the nodes of the network and the leader node.

3.4 Jaya Optimisation

This algorithm always tries to get closer to success (i.e. reaching the best solution) and tries to avoid failure (i.e. moving away from the worst solution). The algorithm strives to become victorious by reaching the best solution and hence it is named as Jaya (a Sanskrit word meaning victory). This proposed method is illustrated by means of an unconstrained benchmark function known as Sphere function. Initially we determine the population size and variable from the stored files of above section and we consider iteration size of minimum 1000.

for each and every iteration of this algorithm, we find the minimum and maximum values of the gaussian similarity values. by using these values we modify our original attributes of the nodes based on the best and worst solutions of similarites of the nodes, which are max and min values respectively. After this, we update our gaussian similarities and store the objective function value given by the respective communities formed as per intervals in our resulted gaussian similarities.

Let f(x) is the objective function to be minimized (or maximized). At any iteration i, assume that there are 'm' number of design variables (i.e. j=1,2,...,m), 'n' number of candidate solutions (i.e. population size, k=1,2,...,n). Let the best candidate obtains the best value of f(x) (i.e. f(x)best) in the entire candidate solutions and the worst candidate worst obtains the worst value of f(x) (i.e. f(x)worst) in the entire candidate solutions. If Xj,k,i is the value of the jth variable for the kth candidate during the ith iteration, then this value is modified as per the following Eq [6]

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - \left|X_{j,k,i}\right|\right) - r_{2,j,i}\left(X_{j,worst,i} - \left|X_{j,k,i}\right|\right)$$

where, Xj,best,i is the value of the variable j for the best candidate and Xj,worst,i is the value of the variable j for the worst candidate. X'j,k,i is the updated value of Xj,k,i and r1,j,i and r2,j,i are the two random numbers for the j th variable during the i th iteration in the range [0, 1]

After successful execution of the above process we now gather the objective function values for each iteration and observe only the one with maximum value. for that particular time quantum we get the similarities of the nodes and also the respective communities. We now say that these are the most optimized communities for the given network using Jaya Optimization approach.

Chapter 4

Conclusion

4.1 Progress

In this work we propose a multi objective Jaya optimization based approach for community detection in complex networks. Here initially feature vectors of the nodes are extracted and then they are send to calculate the similarities of nodes in complex network and then we use eigen vector centrality to determine the leader node. Then we use this leader node as a reference to find the gaussian similarities of all nodes of the complex network.we can then based on the similarities classify the nodes to a certain communities. In order to further optimize these communities we perform 1000 iterations which updates our feature vectors for all nodes and also the gaussian similarities which modifies or updates our community.

Here we calculate the objective function values for each iteration's communities and find the step where the communities yield maximum objective function value. This resulted communities we got are the final optimized communities ,which borrows idea of Jaya Optimization Algorithm.

The implementation of the objective function and its optimization can be done in the near future which we are working on.

4.2 Scope for Future work

These Communities will be tested for results, and they will be plotted to get the actual statistical analysis report of the project and will be compared to various research results to get the performance idea of current project expecting to achieve State-of-art results.

Bibliography

[1] M. Girvan and M. Newman, "J. 2002," *Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2016.

[2] K. Zhou, A. Martin, and Q. Pan, "A similarity-based community detection method with multiple prototype representation," *Physica A: Statistical Mechanics and its Applications*, vol. 438, pp. 519–531, 2015.

[3] F. Cheng, T. Cui, Y. Su, Y. Niu, and X. Zhang, "A local information based multi-objective evolutionary algorithm for community detection in complex networks," *Applied Soft Computing*, vol. 69, pp. 357–367, 2018.

[4] C. Pizzuti, "A multi-objective genetic algorithm for community detection in networks," in *2009 21st IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2009, pp. 379–386.

[5] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[6] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.