Community Detection Using Seed Set Expansion

Dipanjan Karmakar
Iowa State University
Department of Computer Science

Arjun Bhattacharya
Iowa State University
Department of Computer Science

Bijon Kumar Bose
Iowa State University
Department of Computer Science

ABSTRACT
One of the most popular and important characteristic of a complex network is community structure. A community within a network is defined as closely related nodes within the graph such that number of edges between the nodes within the community is more than with nodes outside. Detecting communities has been an area of interest for researchers to find an optimized algorithm and optimize that further. There have been various attempts made to find suitable and scalable algorithms to detect community structures using different techniques. In most of them finding the communities are exhaustive and they are in terms of partition of vertex set and does not allow overlapping of the communities. However overlapping of community structures is relevant in some practical scenarios such as social network. One of the most successful techniques for finding overlapping communities is based on local optimization and expansion of a community metric around a seed set of vertices. In this paper we propose a technique to find overlapping community structures in a network/graph using seed set expansion technique. Like any other seed set expansion technique our algorithm also has two phases – finding a set of good seeds and expanding the initial seed set to form a community. Specifically, we find a set of good seeds based on the degree of connectivity of a node within a graph and then recursively grow the seed sets until all the nodes are exhausted or no more growth or improvement is possible. The basic feature of this algorithm is its simplicity and at each iteration, it tries to grow the community and at the same time tries to improves the quality of the community as well. We also show the effectiveness of this algorithm in finding communities, using experimental results.

Categories and Subject Descriptors
[**Large Datasets**]: Network Analysis, Community Detection using seed set expansion

General Terms
Algorithms, Experimentation.

Keywords
Community Detection, Seed Set Expansion, Overlapping Communities, Clustering, Network Analysis, Large Datasets.

1. INTRODUCTION

In a social network, communities correspond to its social circles and generally nodes can belong to multiple social circles. Therefore it is important to study the problem of finding overlapping community structures in network analysis. A very profound strategy is the seed set expansion algorithms. In our project we have studied this overlapping nature of communities in networks. We have worked on finding a new algorithmic approach which will maintain a feasible performance as well as improve the quality of the communities detected. By improving quality we mean including as many nodes as possible inside the community structure which will form a dense cloud of closely related nodes. More specifically, we investigate how to select a better seed sets in a method for overlapping community detection for unweighted as well as weighted graphs that expands the communities around seed nodes. These local expansion methods are among the most successful strategies for finding overlapping communities in a network strucutre. There are many strategies to find seed sets which include exhaustively exploring all individual seeds or greedy methods that randomly pick a vertex, row, a cluster, and continue with any unassigned vertex. The goal of these traditional, exhaustive clustering methods is to determine a cluster for each and every data point or node. In our study we have focused not only on unweighted graphs but the weighted version as well, where weights on the edges can represent meaningful information like the reputation of a node, trustworthiness, distance, reliability etc. This representation can be very useful in terms of social network analysis, communication networks, marketing strategies etc. This is the reason why we have considered not only the degree of a node, but the weights on edges as well while selecting the seed nodes. The degree of a node represents the connectedness of a node and the weights might

represent the influence of this node to others. We have calculated the relative connectivity factor and the degree of reputation for a node, taking the whole node set into context and order them by the factor and then select the best nodes as seed set. These seeds are the centers of the communities that we need to find in the network. We have considered each individual seed and its adjacent nodes to be in the centre of the communities and the algorithm then recursively finding seed nodes from the rest of the graph and adding the next set of clusters to the appropriate community cores. While adding the next set of clusters we also check how the new clusters are improving the modularity of already defined community cores and accordingly add the newly formed clusters. The main contributions of our study are:

i) we have defined two new factors – degree of connectedness and degree of reputation,
ii) a new seeding strategy which works well for un-weighted graphs as well as weighted graphs,
iii) a new algorithm for seed set expansion phase,
iv) experimental results on real life data sets
v) analysis of the experimental results and study the behavior of our algorithm.

Our method scales very well for large datasets and we have tested it on datasets with nearly 5000 nodes as well.

2. RELATED WORKS

We summarize a few closely related methods and studies in the area of community detection for network analysis.

- In the paper 'Communities Detection in large networks'[1], the authors introduce a technique for detecting communities of highly connected nodes within a network. The proposed method (Spectral Method) is related to the edge-betweenness technique but is claimed to have an efficient running time. This method assigns nodes to communities based on eigenvectors of matrices related to the adjacency matrix (viz. Laplacian and Normal matrices). Eigenvalues close to 1 in the Normal spectrum and those close to 0 in the Laplacian spectrum tend to expose the cluster structure in the original graph. The proposed technique is tested on a word association network which is derived by recording the response words of volunteers to a defined set of stimuli words. A network is built where words are nodes, and directed links are drawn from each stimulus to the corresponding responses. By applying the algorithm, words that are naturally associated by their meaning are clustered together. The disadvantage of such a community detection method lies in the fact that the computational complexity increases with the increase in the number of matrix vector multiplications. So, when applied to a very large data set, scaling up can be difficult.
- In 'Community Detection in Social Networks Based on Influential Nodes' [3], paper an attempt have been made to implement an idea of Influential nodes to extract community from a social network. Influential node is a node that has high influence on the network. Finding influential node in a community has high importance in various fields like marketing where influential nodes can be targeted by producers to promote their products. The main idea of the approach is to use random walk to find the influential nodes based on their degree and division of the nodes based on the influential nodes are done and checks are performed of smaller clusters can be merged into a big cluster. Even though the algorithm has many important aspects, it has certain drawbacks. It focuses its basis on influential node which may not be useful in cases where the degrees of nodes are almost similar. Modularity of the community can be improved by keeping the scope of the implementation bounded to certain areas where influence of the topic really matters the effects on others. Influences of the nodes are based on the degree of the nodes. However degree of the nodes may not alone be the correct measure of influence of a node in all cases.
- In 'Overlapping Community Detection using Bayesian Nonnegative Matrix Factorization' [4] Ioannis Psorakis et al proposed an overlapping community detection method which utilizes Bayesian nonnegative matrix factorization (NMF)

model to extract overlapping modules from a network. This method has the advantage of soft-partitioning solutions where communities are allowed to share members; soft membership distributions, which quantify 'how strongly' each individual participates in each group; excellent module identification capabilities; and the method does not suffer from the drawbacks of modularity optimization methods, such as the resolution limit. They had defined a posterior bases cost function which aim to maximize the model posterior given the observations and finally they find the inner rank K* denoting the inferred number of latent modules in the network. They have considered community detection as a generative model in a probabilistic framework. They assume a fully observed adjacency matrix which is not the case for many real life networks.

- In 'Community Extraction for Social Networks' [2] they have proposed a new Community Extraction framework using criterion that has a natural probabilistic interpretation. They have also proposed a hypothesis test to determine the number of communities in a network when prior information on true or desired number of communities to be extracted is not available. Unlike other partition methods, it allows for the rest of the network to include an arbitrary mixture of tight and weak communities or sparsely connected background nodes. Mathematically, the criterion matches the definition of community in a large class of probability models on networks which can be thought of as a generalization of the block model with some parameter constraints. They extracted the tightest community, by focusing on the edges within the candidate community and edges connecting it to the rest of the network, and ignoring edges within the rest of the network. The key feature of the extraction criterion is that it is not symmetric in the two sets into which the network is split. To maximize the extraction criteria, a local optimization technique is used based on label switching known as tabu search. . The algorithm is run for a prescribed number of iterations, and the best solution seen in the course of these iterations is taken to be the final solution. This approach approximates the criterion under null hypothesis and based on this value they decide whether to stop or apply extraction again. Their community detection method extracts one strong closely knit community one at a time from the given social network graph and leaves the remainder of the graph which are comparatively weakly bonded. But they didn't consider the fact that there can be overlapping communities as well. A person or a group of persons might be equally active and influential in a social network in a diverse range of fields. So overlapping communities in a social network can have noticeable impact while partitioning it into different communities. They have considered the graph to be undirected while the relations between nodes in social network should be directed. A person might follow other influential person but the opposite might not be true. So considering undirected graph might impact the quality of the resultant community.
- In 'Overlapping Community Detection Using Seed Set Expansion'[5] paper, they have introduced a seeding strategy based on using a multilevel weighted kernel k-means algorithm on the graph (the Graclus algorithm). They have used corresponding distance function to compute the "centroid vertex" of each cluster and then use the neighborhood set of that centroid vertex as the seed region for community detection. Then they grow a seed based on personalized PageRank random walks. The algorithm begins by filtering out regions of the graph that will not participate in an overlapping clustering. The seed finding algorithm is run and then seed expansion method is executed on the filtered graph. Then they post-process this output to assign communities for the vertices removed by filtering. One main disadvantage of this approach is that this works on undirected graphs, the edge attribute, weight, is not considered at all.

3. PROPOSED WORK

The main motivation behind our work is that we wanted to consider the weights on the edges as well

while considering the set of seed nodes and a greedy method for the seed set expansion phase. Seed nodes are considered to be the centers of the communities to be detected. Degree of a node alone cannot be a good measure to find the seed nodes for a weighted graph. So we have defined few properties for the nodes in a graph – i) Degree of connectedness and ii) Degree of reputation and iii) Quality of a seed node. We measure these properties relative to the whole graph. Instead if we have considered the degree of a node only while selecting the seed set, our approach would have been biased towards high degree nodes. And that would have not been very much practical as a node can have high degree, but the weights or sum of weights on its incident edges might be very less. So it would fail to reveal the true quality of a node. While on the other hand, one single edge incident to a node might have a large weight, may be few order larger than rest of the edges of the graph, but its degree might be very very less even possibly be one. In that case again, considering edge weights only would fail to give us a proper set of seed nodes.

Degree of Connectedness: This property is defined as the fraction of number of edges incident to a node divided by the total no of edges in the graph.

Degree of Connectedness : Degree of a node /(Total number of nodes in a graph)

Degree of Reputation: This property is defined as the fraction of sum of weights on the edges incident to a node divided by the sum of weights on all the edges in the graph.

Degree of Reputation: (Sum of weights on all edges incident to a node)/(Sum of total weights on all the edges in the graph)

Quality of a Seed node: This factor is used to determine one node as seed over other and represents the measure for a node to be selected as candidate for seed set. This factor is calculated as the multiplication of degree of connectedness and degree of reputation. Due to large number of nodes and edges in large datasets the value of degree of connectedness and degree of reputation is very very less. So we multiply quality of a seed node factor for all the nodes with a very large value, say 10^5, to scale up this value for easy calculation. As this factor is multiplied to all the nodes, it does not alter the ranking or any other meaning in the algorithm.

Now, we explain the overall algorithm for our approach. There are basically two steps for our algorithm – selecting good seed set and then growing the seed set to form communities. Actually these two processes are interleaved in our algorithm. We first take the adjacency matrix as input and calculate degree of connectedness, degree of reputation and quality for each node in the graph and then sort them based on their quality. We assume we need to find k number of communities. So, first we select k nodes from the sorted list of quality, which are not directly connected with each other in the first iteration. We take each of these k seed nodes and take adjacent nodes for each of them. Each of these seeds nodes along with their adjacent nodes form the core of the k community structures. After that at each step we select k seed nodes again from the remaining nodes which are not already part of the k community cores. We take this next set of seed nodes and their corresponding adjacent nodes to form k clouds. We determine to which core community these clouds should be connected by determining the links between seed nodes from previous iteration and current iteration. Next we check if connecting a cloud to a community core or multiple cores increase or optimize the modularity of the graph partition. If link exits between multiple cores and a seed cloud and merging the cloud with those cores increases the modularity, then we add the cloud to all of the cores. This step leads to overlapping of the community structure. We repeat these steps until all the nodes of the graph are exhausted or there are less than k number of nodes remaining unvisited. In case there are less than k numbers of nodes unvisited, we only form clouds with the remaining nodes.

Algorithm 1:

FindSeedSet()

Input: Adjacency Matrix of weighted(non negative weights) or unweighted Graph G

Output: A sorted list of nodes in the increasing order of quality

Steps:

1. Calculate degree of connectedness for each node.
2. Calculate degree of reputation for each node.
3. Calculate quality for each node.
4. Return a sorted array which contains the nodes sorted in the increasing order of their quality.

Algorithm 2:

FindCommunities()

Input: Adjacency Matrix of a weighted (non negative weights) or unweighted Graph G, a number k

Output: k number of communities detected their adjacency matrix and modularity of the graph after final partition.

Steps:

1. Repeat until all nodes are not exhausted or number of nodes remaining < k
    a. Take k node from the sorted list of nodes returned in Algorithm 1 which are not immediately connected with each other.
    b. Form k community cores/clouds.
    c. Check if there is any link for each cloud with each of the community cores.
    d. Check if adding the cloud to each core to which it has some path-link.
    e. Calculate the modularity considering the cores only.
    f. Check if adding the cloud to cores increases modularity of the graph.
    g. If there are multiple such cores existing then add the cloud to each community core.
2. Calculate the modularity after the final clustering.

For unweighted graph we follow the same algorithms as mentioned above and we assume each edge $e \in G(E)$ has equal weight of 1 throughout the graph. We calculate the different properties of the nodes in the same way and then find the communities. The main feature of our algorithm is that it does not consider weighted and unweighted graphs in the same way. It considers the weights of the graph edges as well as degrees of the nodes and calculates a relative factor to determine the set of seed nodes. Then recursively we find seed set and clouds at each iteration and this approach reduces the search space drastically after each iteration. Thereby it reduces the runtime as well. Before merging the clouds with community cores we also aim to optimize the modularity of the graph so as to maintain the quality of the communities detected.

4. PERFORMANCE EVALUATION

4.1. Performance Metrics

For performance evaluation we have used different datasets to study and analyze the behavior of our algorithm. We have also used modularity to check for the quality of communities detected. We have compared graph coverage by our algorithm and compared the same with other algorithms. We have run this algorithm for a number of different types of network graphs from different categories and sizes and we have mentioned a few in this report.

4.2. Datasets Used

We have taken a number of datasets like – modified version of Zachary Karate Club with edge weights(34 nodes undirected graph), Copperfield Word Adjacencies dataset(112 nodes Undirected unweighted graph), PowerGrid Data Set(4941 nodes) - An undirected, unweighted network representing the topology of the Western States Power Grid of the United States. Data compiled by D. Watts and S. Strogatz, Coauthorships in network science (1589 nodes Undirected Unweighted dataset) - Coauthorship network of scientists working on network theory and experiment, as compiled by M. Newman in May 2006, Facebook Circles Dataset(4039 nodes, 88234 edges) - this dataset consists of 'circles' (or 'friends lists') from Facebook. Facebook data was collected from survey participants using this Facebook app etc.

4.3. Evaluation

Our algorithm largely depends on the value of k i.e. the number of communities to be detected and the number of nodes present in the graph. Also it depends on the interconnectivity between the nodes in the graph. So we have studied the dataset for various sizes of nodes and for each dataset we have run the algorithm for different values of k. We have also checked the modularity after each iteration and for each run of the algorithm. Please refer to Table 1. for few of the experimental results.

4.4. Analysis

From the experimental results we came to the conclusion that our algorithm depends on the connectivity of the nodes inside the community as well as outside. The denser the network is, our algorithm runs faster and with increase in the value of k or increase in the number of nodes the runtime does not change drastically. For example, for the facebook dataset we can see that although the value of k increases, but the runtime does not change that much even if value of k increase by one order. Looking at the modularities for different values of k we also conclude that optimum number of communities for the dataset lies in between 5 and 10 which gives the optimum modularity. We have also run our algorithm for a modified version of Zachary Karate Club dataset

which included weights on the edges. Although with lack in available large weighted dataset we could not perform much experiment for the weighted version.

5. Conclusion and Future Work

There have been numerous attempts made to find the community structures within a network. One of the efficient ones is the seed set base expansion approach. But the existing algorithms do not consider the case for weighted graph or do not consider the edge weights. The edge weight can represent valuable information with respect to network analysis, like distance, popularity, influence, trustworthiness of the nodes. We have made an attempt to consider the weights of the edges as well while selecting the seed set. Not only we find a set of k communities, but we also find k most central nodes in the graph in terms of above mentioned attributes as represented by the edge weight. From our analysis we have seen that our algorithm works very fast for smaller values of k and gives an efficient runtime. One other note to mention in this context is that runtime of our algorithm depends on the number of nodes in the graph, but it does not depend on the number of edges. So no matter how many edges are there, it would give a similar run time for the same set of nodes. From the experimental results we have seen our algorithm works best for dense networks and we can the efficiency for sparse graphs as well and try to optimize the performance for that category of graphs. Although for dense graphs the runtime is not affected much but from the results we have seen that, for sparse graphs the runtime increases highly with the order of increase in the value of k or number of nodes. The reason behind this is the number of comparisons made while the chances of cloud getting attached to a core is determined; the number of comparisons increases exponentially with increase in the value of k and the number of nodes in each cloud gets drastically reduced at each iteration. We have also noted the fact that for sparse graph overlapping increases and this increases the complexity of the algorithm. So our assumption of trying to cover a large number of nodes per clouds is not valid in case of sparse graphs. We can improve our algorithm in this perspective by using a better data structure or expansion method where we don't need so much comparisons. There is scope for future work to improve the logic for the expansion phase.

6. REFERENCES

[1] Andrea Capocci, Vito D.P. Servedio, Guido Caldarelli and Francesca Colaiori, *Communities Detection in Large Networks, Physica A: Statistical Mechanics and its Applications*, 2005, vol. 352, issue 2, pages 669-676.

DOI= 10.1016/j.physa.2004.12.050.

[2] Yunpeng Zhao, Elizaveta Levina1, and Ji Zhu. *Community Extraction for Social Networks.* Department of Statistics, University of Michigan, Ann Arbor, MI 48109. Proc Natl Acad Sci U S A. 2011 May 3; 108(18): 7321–7326. Published online 2011 Apr 18.

DOI = 10.1073/pnas.1006642108

[3] Chang Su, Yukun Wang, Lan Zhang. *Community Detection in Social Networks Based on Influential Nodes,* Journal of Software, Vol 9, No 9 (2014), 2409-2416, Sep 2014.

DOI= 10.4304/jsw.9.9.2409-2416.

[4] Ioannis Psorakis, Stephen Roberts, and Mark Ebden, Pattern Analysis and Machine Learning Research Group, Department of Engineering Science, University of Oxford, *Overlapping Community Detection using Bayesian Nonnegative Matrix Factorization.*

[5] Joyce Jiyoung Whang, David F. Gleich, Inderjit S. Dhillon.

Overlapping Community Detection Using Seed Set Expansion.

[6] Mingming Chen, Tommy Nguyen, Boleslaw K. Szymanski. *On Measuring the Quality of a Network Community Structure.*

[7] Mingming Chen, Konstantin Kuzmin, Student Member, IEEE, and Boleslaw K. Szymanski, Fellow, IEEE. *Community Detection via Maximization of Modularity and Its Variants.*

[8] Stephen Kelley, Mark Goldberg,Malik Magdon-Ismail, Konstantin Mertsalov, and Al Wallace. *Defining and Discovering Communities in Social Networks.*

Dataset	k = 5		k = 10		k = 50		k = 100		k = 200	
	Time	m	Time	m	Time	m	Time	m	Time	m
ZacharyKarateGraph	1	0.40	1	0.48	NA	NA	NA	NA	NA	NA
CoAuth	6	0.48	6	0.55	8	0.67	12	0.78	17	0.88

PowerGrid	37	0.17	115	0.40	869	~1	1270	~1	NA	NA
Facebook	118	0.71	169	0.75	213	0.67	116	0.65	40	0.64

Table 1: All the times mentioned above are in seconds and in all cases we have achieved 100 percent coverage of the graphs which is much more than using tools like Demon, or BigClam.