Community Detection on Social Networks using Social Facets.

A Project Report Submitted in
Partial Fulfilment of the Requirements for the
Degree of Bachelor of Technology

by

Diksheet Agarwal 1912059
Aditya Agarwal 1912104
Mehul Dewangan 1912154

Under the Supervision of

Dr. Anupam Biswas
Assistant Professor

Computer Science & Engineering Department
NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

May, 2023

COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Declaration

Thesis Title: **Community Detection on Social Networks using Social Facets.**
Degree for which the Thesis is submitted: **Bachelor of Technology**

I declare that the presented thesis represents largely my own ideas and work in my own words. Where others ideas or words have been included, I have adequately cited and listed in the reference materials. The thesis has been prepared without resorting to plagiarism. I have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the thesis. I understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

Date : 15/05/2023

Diksheet Agarwal (1912059)

Aditya Agarwal (1912104)

Mehul Dewangan (1912154)

COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

It is certified that the work contained in this thesis entitled **"Community Detection on Social Networks using Social Facets."** submitted by **Diksheet Agarwal** (1912059), **Aditya Agarwal** (1912104), **Mehul Dewangan** (1912154) for the B.Tech. End Semester Project Examination May, 2023 is absolutely based on their own work carried out under my supervision.

Place: Silchar
Date: 15/5/2023

(Dr. Anupam Biswas)
Computer Science & Engineering
National Institute of Technology Silchar

Abstract

Community detection is an effective approach to unveil relationships among individuals in online social networks. In the literature, quite a few algorithms have been proposed to conduct community detection by exploiting the topology of social networks and the attributes of social actors.When analyzing different networks, it may be important to discover communities inside them. Community detection techniques are useful for social media algorithms to discover people with common interests and keep them tightly connected. Community detection can be used in machine learning to detect groups with similar properties and extract groups for various reasons. For example, this technique can be used to discover manipulative groups inside a social network or a stock market

Acknowledgements

We would like to acknowledge and provide our sincere thanks to our project supervisor Dr Anupam Biswas, Assistant Professor, CSE, NIT Silchar for his guidance and help throughout the project period. We would also like to express our gratitude to our mentor during the timeframe, Ms. Soumita Das for her immense support and mentorship in the partial completion of the project.

We would like to thank our parents for their constant support and encouragement. We would also like to thank our friends for being a strong pillar whenever needed to rest throughout this perilous journey.

We would also like to show our profound gratitude to the Department of Computer Science and Engineering, NIT Silchar for their meaningful contribution to this opportunity.

Diksheet Agarwal - 1912059
Aditya Agarwal - 1912104
Mehul Dewangan - 1912154

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Communities in social networks refer to groups of individuals who share common interests, beliefs, or behaviors and interact with each other more frequently than with other members of the network. Communities can be identified based on the patterns of social connections among individuals, which can reveal clusters or subgroups within the network. Communities are broadly classified into two types. Disjoint Community and Overlapping Communities.

Disjoint Communities in social networks refer to groups of individuals who are not connected with other groups. There are no nodes common to two groups of communities.

Overalapping Communitiesin social networks refer to groups of individuals who belong to multiple communities simultaneously. There can be overlapping nodes in these communities, which means that the nodes is a part of multiple communities.

1.1 Community Detection

The rapid growth of online social media has made the need of fast algorithms greater than ever for analyzing their information. Community Detection is one of the fundamental problems in network analysis, where the goal is to find groups of nodes that are, in some sense, more similar to each other than to the other nodes. Social media analysis has attracted special attention in the recent decade.In these networks, people are represented as nodes and their relations are shown as links or edges where most of the nodes are arranged in dense parts that are called communities or clusters. Each community is a group of nodes that have high

density relationships inside themselves, but they have low connections with the rest of the network.

Community Detection helps us uncover the underlying structure of the community and the important nodes in the network. It finds its use in a number of cases. For example, clustering of web clients who are geographically close to each other and share similar interests, can be served by the same server computer. Community detection also finds its place in marketing

Because of the importance of community detection as an essential tool for analyzing hidden information of networks, a wide range of algorithms with different features are studied. Recently, the main focus of researchers is on developing local methods, modularity-based methods, and non-negative matrix factorization-based method.

Community detection methods can be broadly categorized into two types; Agglomerative Methods and Divisive Methods. In Agglomerative methods, edges are added one by one to a graph which only contains nodes. Edges are added from the stronger edge to the weaker edge. Divisive methods follow the opposite of agglomerative methods. There, edges are removed one by one from a complete graph.

Community detection is very applicable in understanding and evaluating the structure of large and complex networks. This approach uses the properties of edges in graphs or networks and hence is more suitable for network analysis rather than a clustering approach. The clustering algorithms have a tendency to separate single peripheral nodes from the communities it should belong to. Many different algorithms have been proposed and implemented for network community detection. Each of these has various pros and cons depending on the nature of the network as well as the applying problem domain.

1.2 What is Information Diffusion

Diffusion is the process by which information is spread from one place to another through interactions. It is a field that encompasses techniques from a plethora of sciences and techniques from different fields such as sociology, epidemiology, and ethnography. Of course, everyone is interested in not getting infected by a contagious disease. The diffusion process involves three main elements as follows:

Sender: A sender (or a group of senders) is responsible for initiating the diffusion process.

Receiver: A receiver (or a group of receivers) receives the diffusion information from the sender. Commonly, the number of receivers is higher than the number of senders.

Medium: This is the channel through which the diffusion information is sent from the sender to the receiver. This can be TV, newspaper, social media (e.g., a tweet on Twitter), social ties, air (in the case of a disease spreading process), etc.

From a network point of view: how is the diffusion process handed over? In fact, social relations play a significant role. They are the channels by which social contagion and persuasion are done. Particularly, the structural positions of persons and their personal characteristics make some people more ready to adopt the innovation than others. Networks with different patterns of connection have different properties regarding how things are propagated, which have significant implications for interventions into, for example, rumor propagation. A diffusion starts with an adopter (or a few number of adopters) who spreads the innovation to others. Innovation typically represents newness, it is not the same thing as invention, it is both a process and an outcome, and it involves discontinuous change. Those who adopt early are often too innovative to be influential in a local network. They contaminate their contacts who in turn contaminate their contacts and so on. The more people a person is linked to, the greater the chances that that person will adopt the innovation. At a larger scale, and since communities are interlinked, it is very likely that an innovation jumps from one community to another via boundary spanners (or bridges) and starts over diffusing again. It is a characteristic of social networks. However, any diffusion process can be expedited, delayed, or even stopped if it is discovered that the product (e.g., a video, an audio, a book, etc.) is faulty, and it should be fixed and then released again. This process is called an intervention. Intervention can be achieved via several methods such as stopping the production of the product, limiting the distribution of the product, restricting the exposure to the product, reducing the interest in the product, or reducing interactions within the population. In any way, intervention processes can cause damage to the work of small companies as many customers will no longer trust the products that are produced by these companies.

CHAPTER 2

Literature Survey

2.1 Disjoint Community Detection

2.1.1 Information diffusion based methods

COSINE: Community-Preserving Social Network Embedding from Information Diffusion Cascades [1]: This paper studies the problem of social network embedding without relying on network structures that are usually not observed in many cases. They address that the information diffusion process across networks naturally reflects rich proximity relationships between users. Meanwhile, social networks contain multiple communities regularizing communication pathways for information propagation. Based on the above observations, a probabilistic generative model, called COSINE, to learn community-preserving social network embeddings from the recurrent and time-stamped social contagion logs, namely information diffusion cascades. The learned embeddings therefore capture the high-order user proximities in social networks. Leveraging COSINE they are able to discover underlying social communities and predict temporal dynamics of social contagion. Experimental results on both synthetic and real-world datasets show that our proposed model significantly outperforms the existing approaches.

A Fast Local Balanced Label Diffusion Algorithm for Community Detection in Social Networks[2]: In this paper, a fast community detection algorithm based on local balanced label diffusion (LBLD) is proposed. The LBLD algorithm starts with assigning node importance scores to each node using a new local similarity measure. After that, top 5% important nodes are selected as initial rough cores to expand communities. In the first step, two neighbor nodes

with highest similarity than others receive the same label. In the second step, based on the selected rough cores, the proposed algorithm diffuses labels in a balanced approach from both core and border nodes to expand communities. Next, a label selection step is performed to ensure that each node is surrendered by the most appropriate label. Finally, by utilizing a fast merge step, final communities are discovered. Besides, the proposed method not only has a fast convergence speed, but also provides stable and accurate results. Moreover, there is no randomness as well as an adjustable parameter in the LBLD algorithm.

A fast community detection algorithm using a local and multi-level label diffusion method in social networks [3]: In recent years, many algorithms for community detection have been proposed. These algorithms can be classified into two categories: globally community detection algorithms and locally community detection algorithms. Global methods have higher time complexity in general and their performance is very poor in the case of large-scale networks. Local algorithms take into account the impact of the neighborhood only at the first level, while semi-local algorithms use the neighborhood effect at the second and the third level of the neighborhood. The global methods include graph partitioning-based methods, hierarchical clustering-based methods, traditional clustering methods, and the modularity optimization-based methods, to name a few. Local community detection algorithms include label propagation-based, diffusion-based, core-expanded-based, and other local similarity-based methods

A cascade information diffusion-based label propagation algorithm for community detection in dynamic social networks [4]: A variety of approaches exists for detecting communities in dynamic social networks, among which the label propagation algorithm (LPA) is the well-known approach. This approach has made remarkable performance, but still has several problems. One of the difficulties of this approach is the new nodes added to the social network graph in the current snapshot has a very slight chance of creating new communities. In fact, these nodes fall under the influence of existing communities. This drawback decreases the accuracy of community detection in dynamic social networks. We propose a new method based on label propagation approach and the cascade information diffusion model in order to solve this difficulty. Here, the newly proposed method, Speaker Listener Propagation Algorithm Dynamic (SLPAD), Dominant Label Propagation Algorithm Evolutionary (DLPAE) and Intrinsic Longitudinal Community Detection (ILCD) on real and synthetic networks are implemented. The findings indicate that the modularity and Normalized Mutual Information (NMI) and also F1AVG of this proposed method is considerably higher than the earlier available methods in most datasets. Therefore, it can be concluded that the proposed method improves the accuracy of community detection in comparison with other available methods.

Community Detection Using Diffusion Information [5]: This paper focuses on detecting communities by exploiting their diffusion information. To this end, we utilize the Conditional Random Fields (CRF) to discover the community structures. The proposed method, community diffusion (CoDi), does not require any prior knowledge about the network structure or specific properties of communities. Furthermore, in contrast to the structure-based community detection methods, this method is able to identify the hidden communities.

Information diffusion-aware likelihood maximization optimization for community detection [6]: When information diffusion occurs in the network, one can observe the cascade data in which nodes participate in the propagation process, which reflects the network's community structure to some extent. In this paper, a likelihood maximization model by utilizing the diffusion information and propose two different optimization algorithms to obtain community division of the network is proposed. A probability inferring model by utilizing the likelihood maximization method based on the information propagation with the integration of community structure, which can effectively capture the behavioral features of nodes' participation in information propagation at the community level of the network. Two different optimization algorithms based on the likelihood maximization model: the EM-CD algorithm adopts an iterative likelihood optimization strategy based on Expectation–Maximization, and the L-Louvain algorithm modifies the Louvain algorithm by replacing modularity optimization with likelihood maximization as a criterion and performs community repartition greedily to maximize the likelihood probability of propagation cascades.

2.1.2 Game theory based methods

Community detection in dynamic social networks- A game theoretic approach [7]: The community detection problem is empirically formulated from a game theoretic point of view and solved using a Crowding based Differential Evolution algorithm adapted for detecting Nash equilibria of noncooperative games. Numerical results indicate the potential of this approach. In this paper, we adopt the community definition proposed by Lancichinetti in [8]. A community is a subgraph identified by the maximization of the following fitness: $f^C = \frac{k^{in}(C)}{(k^{in}(C)+k^{out}(C))^\alpha}$ where $k^{in}(C)$ is the total internal degree of nodes in community C and equals the double of the number of internal links of that community. $k^{out}(C)$ is the total external degree of nodes in community C and can be computed as the number of links joining each member of the module with the rest of the graph. α a is a positive real-valued parameter, controlling the size of the communities.

2.2 Overlapping Community Detection

2.2.1 Information diffusion based methods

A Hierarchical Diffusion Algorithm for Community Detection in Social Networks [9]: This paper proposes a hierarchical diffusion method to detect the community structure. This algorithm is based on the idea that people in different communities usually share fewer common friends. It also makes use of the fact that people usually make decisions-based others 'choices, especially their friends'. This algorithm can distinguish between pseudo-communities and meaningful ones. Tests on both classical and synthetic benchmarks show that this algorithm is comparable to state-of-the-art community detection algorithms in both computational complexity and accuracy measured by the so-called normalized mutual information. In social network analysis, an edge(u, v) in graph G is a bridge, if deleting this edge would cause u and v to become disconnected. An edge(u,v) in a network is a local bridge if u and v have no friends in common. If an edge(u, v) with $W(u, v)$ functions as a bridge or a local bridge, $W(u, v)$ must be very small. As a result, after the thresholding step, the endpoints of a bridge are unlikely in the same community. Next is the diffusion of membership. Similar with the diffusion of innovation, the cascade of joining community m can be model as a natural model of direct-benefit effects in networks [10]. According to the assumption of direct-benefit model, people will benefit from directly copying others' decisions in that the cost for the people within the same group is usually lower than that for people in different groups. In this implementation, one node has only one community ID. If we relax this constraint by allowing every node to join as many communities as it needs, and the more communities it joins, the more taxes it pays. It's possible to generalize our algorithm to detect the overlapping communities.

TABLE 2.1: Literature Survey of the papers studied

Name	Author	Year	Key Features	Challenges
Community detection in dynamic social networks: A game-theoretic approach	Hamidreza Alvari, Alireza Hajibagheri, Gita Sukthankar	2014	Dynamic Game Theory method (D-GT) Treats the nodes of the network as rational agents	The algorithm is computationally intensive, especially for large networks, which can make it impractical for some applications The algorithm may not be scalable to networks with a large number of nodes or a high degree of connectivity
COSINE: Community-Preserving Social Network Embedding from Information Diffusion Cascades	Yuan Zhang, Tianshu Lyu, Yan Zhang	2017	Probabilistic generative model.Uses recurrent and time-stamped information diffusion cascades. The COSINE algorithm is designed specifically for social network embedding based on information diffusion cascades	The algorithm involves calculating pairwise cosine similarity between nodes, which can become computationally expensive for large graphs. Another disadvantage is that the algorithm assumes that information diffusion cascades are a reliable indicator of community membership
A Fast Local Balanced Label Diffusion Algorithm for Community Detection in Social Networks	Hamid Roghani, Asgarali Bouyer	2022	Communities are formed from both core and border nodes	Sensitive to the initial conditions, and the quality of the results may depend on the choice of initial communities. May not be able to detect small or overlapping communities, and the resolution of the detected communities may be limited
A cascade information diffusion based label propagation algorithm for community detection in dynamic social networks	Mohammad Sattari, kamran Zamanafir	2022	Cascade information diffusion model CIDLPA has been proposed	Sensitive to the initial conditions, and the quality of the results may depend on the choice of initial communities. The quality and quantity of data used to generate the information diffusion cascades can greatly affect the quality of the resulting

TABLE 2.1: Literature Survey of the papers studied

Name	Author	Year	Key Features	Challenges
A fast community detection algorithm using a local and multi-level label diffusion method in social networks	Bouyer, Asgarali; Azad, Khatereh; Rouhi, Alireza	2018	Influence on node neighbours has been considered	Assumes that all nodes in the network are equally important. The algorithm relies heavily on label diffusion, which can make it difficult to understand and interpret the results
A Hierarchical Diffusion Algorithm for Community Detection in Social Networks	Keyi Shen, Li Song, Xiaokang Yang, Wenjun Zhang	2010	Adopts a hierarchical framework and discloses the community structures at different scales	Relies heavily on the structure of the network and the diffusion process. If the diffusion process is not representative of the true social interactions in the network, the algorithm may produce inaccurate results.
Information diffusion-aware likelihood maximization optimization for community detection	Zheng Zhang, Jun Wan, Mingyang Zhou, Kezhong Lou, Guoliang Chen	2022	Likelihood maximization model by utilizing the diffusion information and propose two different optimization algorithms to obtain community division of the network	Is sensitive to noise in the network, which can lead to false positives or false negatives. IDALM does not always provide robust result and may be sensitive to the choice of initial conditions or the number of communities specified.
Community Detection Using Diffusion Information	Maryam Ramezani, Ali Khodadadi, Hamid R. Rabiee	2018	Utilize the Conditional Random Fields to discover the community structures. Community diffusion, do not require any prior knowledge about the network structure	Performance can depend on the choice of diffusion parameters, such as the diffusion time or the damping factor. Selecting appropriate parameters may require a significant amount of trial and error.

CHAPTER 3

Motivation

Community Detection is part of a much wider domain called as Social Network Analysis. Through this, researchers are able to find the core nodes which have greater influence in the graphs and now the information is passed on. There have been little research on the community detection for large datasets. Disjoint communities are good for understanding the working of different algorithms, but its the overlapping communities which gives us a real sense of how communities are formed in real life.

CHAPTER 4

Problem Statement

"Community detection on social networks using social facets."

The problem addressed in the statement is the need to identify communities or groups within social networks accurately and efficiently. Community detection is a crucial task in various fields, including social sciences, marketing, and cybersecurity. Traditional community detection methods based on network topology alone may not consider other social aspects that influence the formation and maintenance of communities. Therefore, there is a need to explore new approaches that leverage social facets to improve the detection of communities.

CHAPTER 5

Proposed Work

5.1 Disjoint Community Detection Algorithms

5.1.1 Implementation of LBLD

In this paper, a fast community detection algorithm based on local balanced label diffusion (LBLD [2]) is proposed. The LBLD algorithm starts with assigning node importance scores to each node using a new local similarity measure. After that, top 5% important nodes are selected as initial rough cores to expand communities. In the first step, two neighbor nodes with highest similarity than others receive the same label. In the second step, based on the selected rough cores, the proposed algorithm diffuses labels in a balanced approach from both core and border nodes to expand communities. Next, a label selection step is performed to ensure that each node is surrendered by the most appropriate label. Finally, by utilizing a fast merge step, final communities are discovered. Besides, the proposed method not only has a fast convergence speed, but also provides stable and accurate results. Moreover, there is no randomness as well as an adjustable parameter in the LBLD algorithm.

To demonstrate accuracy and efficiency of the proposed LBLD algorithm, it is tested on real-world and synthetic datasets with a wide variation range of nodes. The results proved superior performance and accuracy of the proposed method in large-scale networks as well as small networks. Running times showed that LBLD has much better running time than other methods due to its efficient data structure and lack of time-consuming operations.

5.1.2 Implementation of FSLD

In recent years, many algorithms for community detection have been proposed. These algorithms can be classified into two categories: globally community detection algorithms and locally community detection algorithms. Global methods have higher time complexity in general and their performance is very poor in the case of large-scale networks. Local algorithms take into account the impact of the neighborhood only at the first level, while semi-local algorithms use the neighborhood effect at the second and the third level of the neighborhood. FSLD [3] is one of the local methods. The global methods include graph partitioning-based methods, hierarchical clustering-based methods, traditional clustering methods, and the modularity optimization-based methods, to name a few. Local community detection algorithms include label propagation-based, diffusion-based, core-expanded-based, and other local similarity-based methods.

.

5.1.3 Implementation of Disjoint community detection using Cascades

DCC is disjoint community detection using cascades. This algorithm uses the concepts of tie-strength and cascades for detecting disjoin communities in the network. Tie strength between 2 nodes is the strength of the edge connecting those nodes. The tie strength is calculated by the number of common neighbors and the total number of neighbors of those nodes. Cascades are the paths that are being traced by tracking the increased tie-strength starting from the first node. The cascades that are generated are treated as initial communities. All the cascades whose path length is equal to 1 are disregarded and all the nodes in those cascades are considered to be unassigned. Then all the unassigned nodes are allotted a community based on preferential membership. Final communities are being derived from merging the initial communities by finding the common nodes in them and merging them.

5.2 Overlapping Community Detection Algorithms

5.2.1 Implementation of Adjacency Propagation Algorithm

In this paper, Adjacency Propagation Algorithm (APAL) [11], the adjacent vertices are taken into consideration, as they are best suited for finding the the overlapping communities in undirected, unweighted graphs. APAL leverages the idea of label propagation and propagation on adjacency matrices to identify overlapping communities in biological networks. The algorithm iteratively updates a label matrix based on the similarity between nodes and their neighbors, and then applies a clustering algorithm to identify overlapping communities. The algorithm uses a single parameter threshold and calculates the network intraconnectivity property. The algorithm is tested against the real world datsets and LFR benchmarks.

APAL defines a community C as a subgraph of a graph $G, C \subseteq G$ in which the intraconnectivity as defined by $\alpha = \frac{k}{n(n-1)}$ of the community is above a given threshold value, t, where k is the sum of internal degrees and n is the number of nodes. The running time complexity of APAL is $O(m^3/n^2)$ where m s the number of edges.

5.2.2 Implementation of Label Propagation Algorithm with Neighbor Node Influence

Its the updated version of the traditional LPA (Label Propagation Algorithm), with added Node Neighbours influence to detect the communities with greater precision. LPANNI [12] leverages the idea of label propagation to identify communities in complex networks. The algorithm iteratively updates a label matrix based on the similarity between nodes and their neighbors, and then applies a post-processing step to identify overlapping communities.

It consists of two phases. In the first phase, it computes the NI of all nodes and sorts the nodes in ascending order according to NI. In the second phase, it propagates labels by considering both the sequence of NI and NNI until the algorithm converges, then the overlapping community structure is detected. It effectively avoid the randomness and thus improve the stability of LPA algorithms.

5.2.3 Implementation of Overlapping Community Detection using Edge Reduction and Common Neighbor Selection (ERCNS)

In this algorithm to detect overlapping communities using edge betweenness and a post-processing step to convert disjoint communities to overlapping ones. The algorithm consists of two main steps:

Step 1: Edge Betweenness-Based Hierarchical Clustering

The first step in our algorithm is to use edge betweenness as a measure of the importance of each edge in the network. Edge betweenness measures the number of shortest paths that pass through a given edge in the network, and edges with high betweenness are likely to be bridges between communities. Therefore, we can use edge betweenness to sort all the edges in the network and remove the edges with the highest betweenness iteratively. The number of iterations is controlled by an input parameter called "iterations," which determines how many edges to remove from the network.

After each edge removal, we can observe the network's clustering structure, which consists of disjoint communities. This process continues until the desired number of iterations is reached, and we obtain a hierarchical clustering of the network.

Step 2: Disjoint to Overlapping Community Conversion

The second step in our algorithm is to convert the disjoint communities obtained in Step 1 to overlapping communities. This step is crucial as the hierarchical clustering process in Step 1 only produces disjoint communities.

To convert the disjoint communities to overlapping ones, we use a post-processing step. For each community in the disjoint list, we iterate over all the nodes in the community. For each node neighbor, we check the percentage of neighbors that lie in the current community. If this percentage is greater than a specified threshold, we add that neighbor to the current community. This process repeats until no more nodes can be added to the current community.

By iteratively adding nodes based on the threshold percentage, we can obtain overlapping communities. The threshold percentage can be adjusted based on the desired level of overlap in the communities. The final step in our algorithm is to output the overlapping communities obtained in Step 2. These communities can be represented as a list of nodes belonging to each community.

5.2.3.1 Pseudo Code

Algorithm 1 Disjoint communities formation

Require: *iters*, *neighbor_selection_threshold*, G
  edges ← number of edges in G
  num_iterations ← $iters * edges$
  while *num_iterations* - - **do**
    e ← edge with highest edge betweenness centrality
    G ← G - e
  end while
  disjoint_communities ← connected components of G

Algorithm 2 Conversion of disjoint communities to overlapping communities

Require: *disjoint_communities*
  communities ← [ ]
  for C in disjoint_communities **do**
    temp_list ← [ ]
    for n in C **do**
      temp_list ← temp_list + n
      for nn in neighbors of n **do**
        if nn in C **then**
          continue
        end if
        total_neighbor_count ← number of neighbors of n
        in_c_count ← 0
        for nnn in neighbors of nn **do**
          if nnn in C **then**
            in_c_count ← in_c_count + 1
          end if
        end for
        if $\frac{in\_c\_count}{total\_neighbor\_count} \geq$ threshold **then**

          temp_list ← temp_list + nn
        end if
        communities ← communities + temp_list
      end for
    end for
  end for

CHAPTER 6

Experimental Results and Discussions

6.1 Experimental setup

6.1.1 Datasets Used

We have ran our code on various real world datasets namely karate club, dolphin, football, polblogs, email, jazz, sawmill, egofacebook, bitcoin, CA-GrQc of these some are small datasets having nodes less than 1000, while others have nodes within 10000 as well.

6.1.2 Matrices used for evaluation

6.1.2.1 Quality Matrices

These matrices are used for evaluating the algorithm from a quality perspective. The matrices used are overlapping modularity, extended modularity, modularity overlap, community coverage.

Community Coverage: This metric counts the fraction of nodes that belong to at least one community of three or more nodes. A size of three is chosen since it constitutes the smallest non-trivial community structure.

6.1.2.2 Accuracy Matrices

These matrices are used for evaluating the algorithm from a precision and accuracy perspective. The matrices used are Onmi, Nmi_max, and nf1.

6.1.3 System Configuration

We have used Google Colab Notebook to run the python notebook (ipynb) files.

Google Colab Specifications:

- n1-highmem-2 instance
- 2vCPU @ 2.2GHz
- 13GB RAM
- 100GB Free Space
- idle cut-off 90 minutes
- maximum 12 hours

6.1.4 Parameters Settings

Algorithms:

1. ERCNS:
   - $iterations = 0.2$
   - $neighbor\_threshold = 0.8$
2. SLPA
   - $t = 21$ (maximum number of iterations)
   - $r = 0.1$ (threshold)
3. LPAM
   - $k = 2$ (number of clusters)

- $threshold = 0.4$ (merging threshold)
- $distance =$ "amp" (type of distance)

4. OCDID
   - $threshold = 0.001$
5. APAL
   - $t = 0.0005$ (threshold)

6.2 Results Of Disjoint Community Algorithm

Algorithm: Disjoint commmunity detection using cascades (DCC)

ARI Score: Adjusted Rand Index (ARI) value lies between 0 and 1. The index value is equal to 1 only if a partition is completely identical to the intrinsic structure and close to 0 for a random partition.

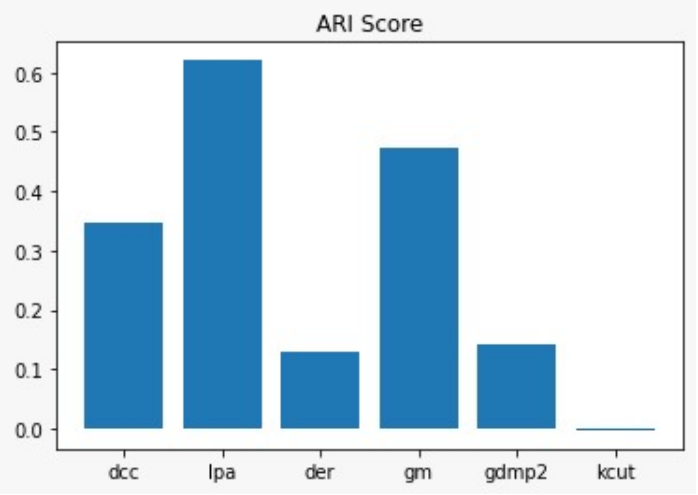

FIGURE 6.1: ARI Score of DCC in comparision with other algorithms.

NMI Score: Normalized Mutual Information (NMI) is a normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).

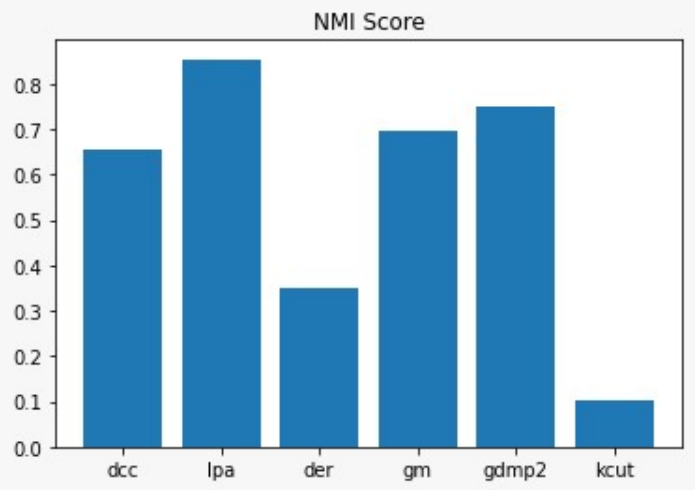

FIGURE 6.2: NMI Score of DCC in comparision with other algorithms.

Cut Ratio: Cut ratio is the fraction of existing edges (out of all possible edges) leaving the community.

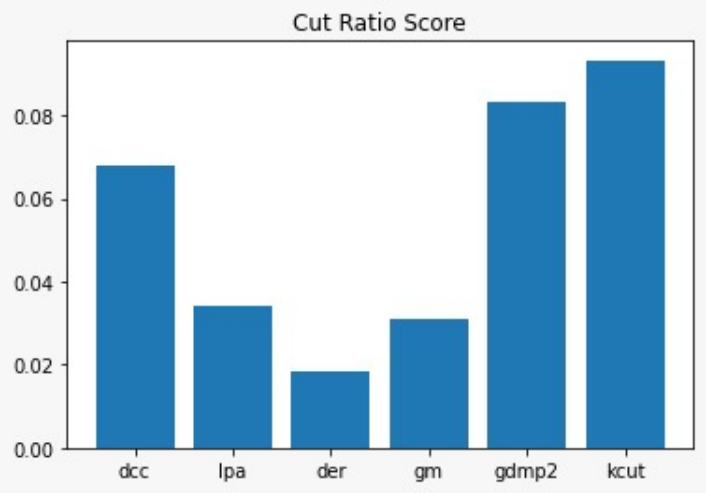

FIGURE 6.3: Cut Ratio of DCC in comparision with other algorithms.

6.3 Results Of Overlapping Community Algorithm

Algorithms such as SLPA [13], LPANNI [12], UMSTMO [14], OCDID [15], APAL [11], LPAM [16], Walkscan [17], Core Expansion [18] and PERCOMVC [19] were ran against many datasets such as karate club, dolphin, football, sawmill, polblogs, CA-GrQc, Jazz and facebook. The results of these are shown in below tables and graphs

TABLE 6.1: Quality Metrics Results on Karate Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	1.0	0.267	0.287	0.211
SLPA	1.0	0.05	0.262	0.278
LPAM	1.0	0.414	0.321	0.356
Core Expansion	0.559	0.23	0.16	0.519
Walkscan	0.941	0.02	0.111	0.021
PERCOMVC	0.941	0.299	0.299	0.123
UMSTMO	0.706	0.068	0.193	0.031
LPANNI	0.1	0.449	0.317	0.532
OCDID	0.941	0.426	0.292	0.518
APAL	0.882	0.38	0.281	0.366

TABLE 6.2: Accuracy Matrices Results on Karate Club Dataset

Algorithm	NF1	ONMI	NMI Max
ERCNS	0.537	0.567	0.611
SLPA	0.345	0.000	0.000
LPAM	0.915	0.866	0.866
Core Expansion	0.333	0.238	0.234
Walkscan	0.585	0.142	0.116
PERCOMVC	0.340	0.457	0.381
UMSTMO	0.031	0.178	0.180
LPANNI	0.509	0.678	0.628
OCDID	0.493	0.586	0.546
APAL	0.431	0.449	0.433

TABLE 6.3: Quality Metrics Results on Dolphin Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	1.0	0.404	0.322	0.133
SLPA	1.0	0.455	0.295	0.222
LPAM	1.0	0.388	0.337	0.295
Core Expansion	0.774	0.281	0.177	0.212
Walkscan	0.403	0.003	0.038	0.3
PERCOMVC	0.661	0.397	0.251	0.326
UMSTMO	0.581	0.139	0.181	0.008
LPANNI	0.935	0.529	0.326	0.163
OCDID	0.887	0.421	0.281	0.012
APAL	0.452	0.254	0.186	0.516

TABLE 6.4: Accuracy Metrices Results on Dolphin Dataset

Algorithm	NF1	ONMI	NMI Max
ERCNS	0.384	0.507	0.499
SLPA	0.453	0.605	0.538
LPAM	0.407	0.476	0.378
Core Expansion	0.138	0.289	0.226
Walkscan	0.145	0.0	0.0
PERCOMVC	0.715	0.473	0.407
UMSTMO	0.019	0.036	0.046
LPANNI	0.428	0.677	0.663
OCDID	0.192	0.398	0.435
APAL	0.292	0.188	0.115

TABLE 6.5: Quality Metrics Results on Football Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	1.0	0.26688	0.28717	0.21066
SLPA	1.0	0.26688	0.28717	0.21066
LPAM	1.0	0.41440	0.32061	0.35560
Core Expansion	0.55882	0.23044	0.16026	0.51880
Walkscan	0.52941	0.06060	0.07514	0.21353
PERCOMVC	0.94117	0.29933	0.29933	0.12278
UMSTMO	0.70588	0.06771	0.19321	0.03053
LPANNI	1.0	0.44888	0.31734	0.53200
OCDID	1.0	0.26688	0.28717	0.21066
APAL	1.0	0.26688	0.28717	0.21066

TABLE 6.6: Accuracy Metrices Results on Football Dataset

Algorithm	NF1	ONMI	NMI Max
ERCNS	0.075	0.314	0.264
SLPA	0.509	0.495	0.479
LPAM	0.062	0.147	0.113
Core Expansion	0.257	0.479	0.39
Walkscan	0.013	0.033	0.025
PERCOMVC	0.6	0.687	0.687
UMSTMO	0.038	0.023	0.027
LPANNI	0.787	0.743	0.755
OCDID	0.304	0.479	0.375
APAL	0.274	0.427	0.345

TABLE 6.7: Quality Metrics Results on Sawmill Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	1.0	0.511	0.333	0.547
SLPA	1.0	0.396	0.317	0.273
LPAM	1.0	0.373	0.319	0.291
Core Expansion	0.833	0.304	0.197	0.358
Walkscan	0.222	0.082	0.051	0.195
UMSTMO	0.917	0.036	0.239	0.05
LPANNI	0.917	0.501	0.311	0.616
OCDID	0.861	0.4	0.281	0.417
APAL	0.667	0.224	0.213	0.337

TABLE 6.8: Accuracy Metrices Results on Sawmill Dataset

Algorithm	NF1	ONMI	NMI Max
ERCNS	0.546	0.552	0.419
SLPA	0.062	0.208	0.419
LPAM	0.093	0.107	0.094
Core Expansion	0.099	0.107	094
Walkscan	0.073	0.253	0.157
UMSTMO	0.104	0.049	0.034
LPANNI	0.206	0.447	0.316
OCDID	0.245	0.35	0.281
APAL	0.323	0.062	0.044

TABLE 6.9: Quality Metrics Results on Polblogs Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	0.754	0.147	0.201	0.185
SLPA	1.0	0.422	0.337	0.149
Core Expansion	0.689	0.057	0.039	0.042
Walkscan	0.098	0.018	0.018	0.106
UMSTMO	0.845	0.003	0.237	0.002
LPANNI	0.98	0.428	0.338	0.238
OCDID	0.883	0.004	0.246	0.348
APAL	0.788	0.098	0.197	0.416

TABLE 6.10: Quality Metrics Results on CA-GrQc Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	0.383	0.288	0.151	0.453
SLPA	0.931	0.745	0.388	0.113
Core Expansion	0.781	0.519	0.278	0.068
UMSTMO	0.086	0.057	0.036	0.001
LPANNI	0.885	0.701	0.361	0.994

TABLE 6.11: Quality Metrics Results on Jazz Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	0.995	0.011	0.251	0.08
SLPA	1.0	0.447	0.31	0.418
Core Expansion	0.96	0.145	0.11	0.135
Walkscan	0.535	0.094	0.161	0.364
PERCOMVC	0.955	0.012	0.251	0.288
UMSTMO	0.747	0.015	0.187	0.005
LPANNI	1	0.438	0.307	0.563
OCDID	0.99	0.007	0.251	0.286
APAL	0.965	0.225	0.279	0.353

TABLE 6.12: Quality Metrics Results on facebook Dataset

Algorithm	Community Coverage	Extended Modularity	Overlapping Modularity	Modularity Overlap
ERCNS	0.840	0.457	0.309	0.213
SLPA	1.0	0.766	0.434	0.21
Core Expansion	0.961	0.486	0.292	0.198
Walkscan	0.298	0.215	0.141	0.082
UMSTMO	0.948	0.022	0.249	0.178
LPANNI	0.994	0.797	0.43	0.4

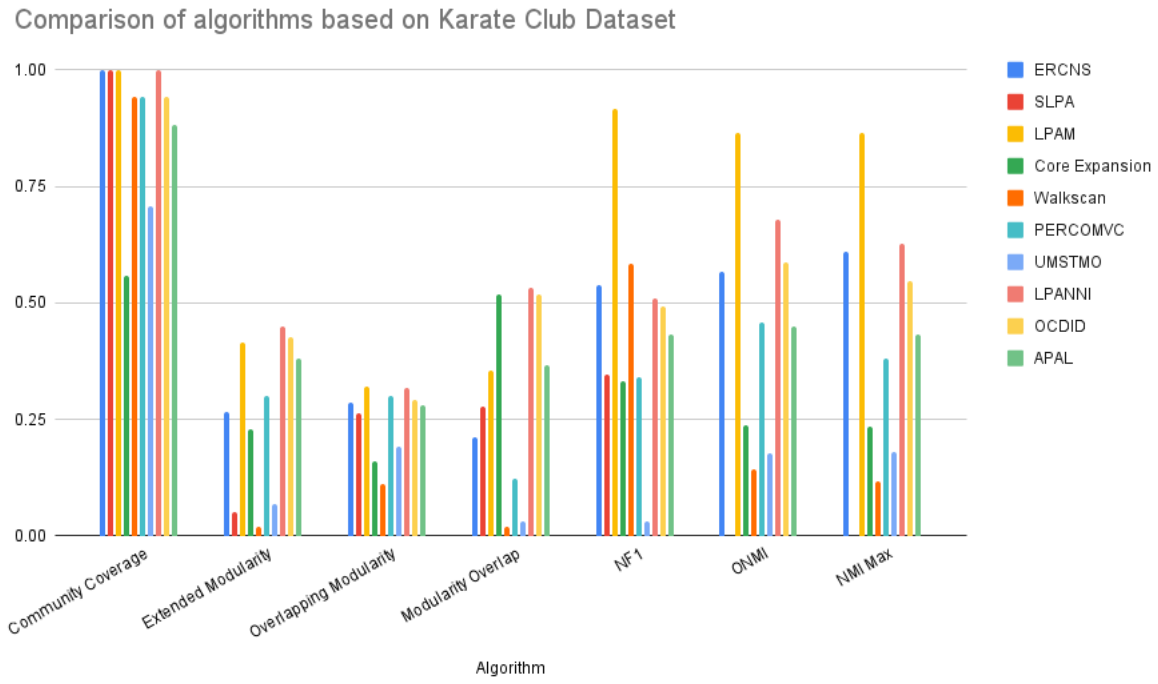

FIGURE 6.4: Comparision of algorithms on Karate Club dataset.

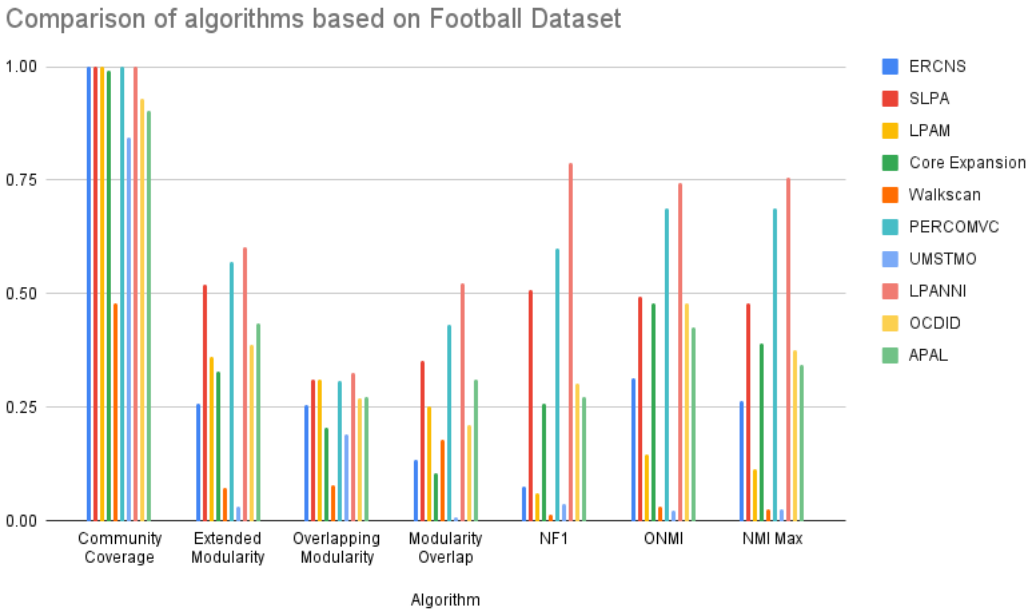

FIGURE 6.5: Comparision of algorithms on Football dataset.

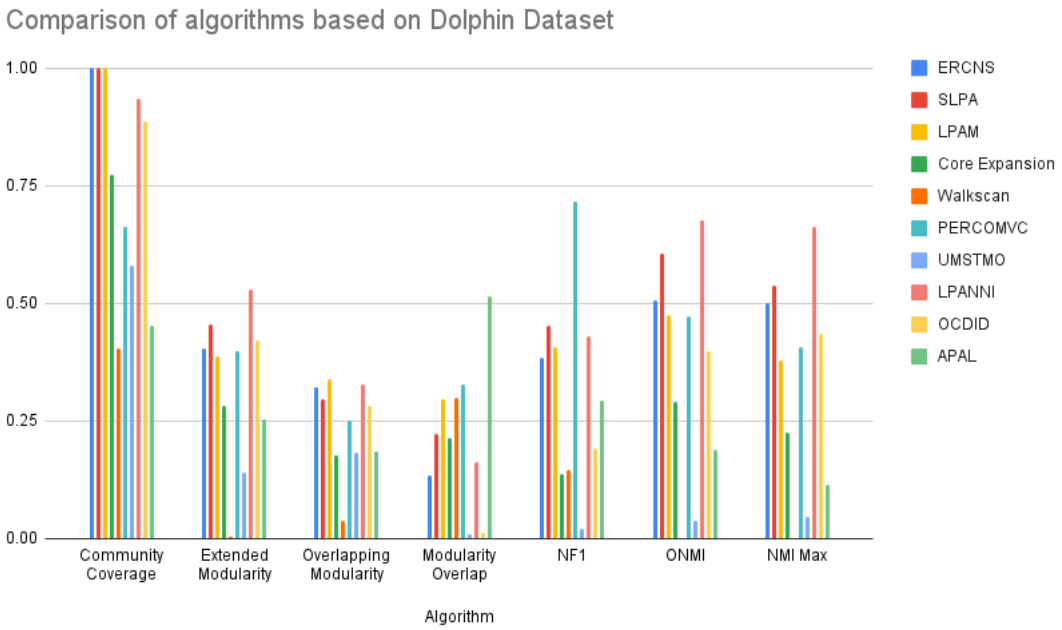

FIGURE 6.6: Comparision of algorithms on Dolphin dataset.

6.4 Result Analysis

6.4.1 Result Discussion of DCC

DCC algorithm was ran against Karate Club Dataset, and was compared with baselines algorithms like lpa, der, gm, gdmp2 and kcut. The table and graphs were shown above.

In Figure 6.1 all the algorithms were ran on Karate Club dataset and the results are shown in the figure for ARI. The results shows that the DCC algorithm gives decent results as compared to other algorithms such as gdmp2 and gm.

Similarly in Figure 6.2, Figure 6.3 all the algorithms were ran on Karate Club dataset and the results are shown in the table for NMI Score and Cut Ratio Score. The results shows that DCC gives decent results as compared to other algorithms such as gm, gdmp2, kcut.

6.4.2 Result Discussion of ERCNS

ERCNS algorithm was ran against many datasets such as karate club, dolphin, football, sawmill, polblogs, CA-GrQc, Jazz and facebook. For the comparative analysis, we chose some baseline algorithms, quality metrics and accuracy metrics. The table and graphs were shown above.

In Table 6.1 all the algorithms were ran on Karate Club dataset and the results are shown in the table for different quality metrics such as community coverage, extended modularity, overlapping modularity, and modularity overlap. The results shows that our novel algorithm (ERCNS) gives decent results as compared to other algorithms such as UMSTMO.

In Table 6.2 all the algorithms were ran on Karate Club dataset and the results are shown in the table for different accuracy metrics such as NF1, ONMI, NMI Max. The results shows that our novel algorithm (ERCNS) gives decent results as compared to other algorithms such as PERCOMVC and APAL.

Similarly in the Table 6.3 and Table 6.4 shows the run on Dolphin dataset. Table 6.5 and Table 6.6 shows the run on Football dataset. Table 6.7 and Table 6.8 shows the run on Sawmill dataset. Table 6.9 shows the run on Polblogs dataset and only shows the quality metrics, because the ground truth was not available. Table 6.10 shows the run on CA-GrQc dataset for quality metrics. Table 6.10 shows the run on Jazz dataset for quality metrics. Table 6.10 shows the run on facebook dataset for quality metrics.

Figure 6.4, Figure 6.5, Figure 6.6 demonstrates the graphical representation of algorithms on various quality and accuracy metrics. They shows run on Karate Club, Football, and Dolphin datasets respectively.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

After implementing the 3 disjoint community detection papers, we got a clear idea of how communities behave in a given list of nodes, and the importance of finding the core nodes in the communities through information diffusion. Time complexity was a major challenge in the Disjoint Community detection using Cascades algorithm. Both LBLD and FSLD algorithm were based on published research paper and prepared us for the improvements needed in the algorithms.

After working on disjoint communities we moved on to overlapping communities, as it demonstrated how the real world communities will behave in actual environments. Through APAL and LPANNI we were able to come up with our algorithm, Overlapping community Detection using Edge Reduction via Common Neighbor Reduction.

The time complexity of the proposed algorithm is $\theta(e * \frac{n*(n-1)}{2} * I + n * M)$, where, e is number of edges, n is number of nodes, I is the number of iterations (parameter), and M is the average degree of the nodes. We evaluated our algorithm on several real-world networks. Our algorithm achieved good accuracy on these networks and was able to identify meaningful communities that aligned well with domain knowledge.

Overall, our algorithm presents a promising approach to community detection that can be applied to a wide range of networks.

7.2 Future Work

Although our community detection algorithm has shown promising results on several networks, there are still many avenues for future work and improvement. Further works can explore ways to improve and extend our algorithm, such as by handling directed and weighted networks or scaling to larger networks. Here, we outline some potential areas for future research:

- While our algorithm works well on networks with up to a few thousand nodes, it may not be scalable for larger networks comprising of nodes greater than hundred thousands nodes. Future work could investigate ways to parallelize or optimize our algorithm to improve its scalability.
- Improvements can be done in the quality matrices. The ERCNS algorithms can be optimised for other quality matrices
- Our algorithm currently only works on undirected and unweighted networks. Future work could explore ways to extend our algorithm to handle directed and weighted networks.
- Although we have measured only the accuracy and quality for overlapping communities, we can use this algorithm to detect just the disjoint communities by disabling steps 2 and 3.

References

[1] Y. Zhang, T. Lyu, and Y. Zhang, "Cosine: Community-preserving social network embedding from information diffusion cascades," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[2] H. Roghani and A. Bouyer, "A fast local balanced label diffusion algorithm for community detection in social networks," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[3] A. Bouyer, K. Azad, and A. Rouhi, "A fast community detection algorithm using a local and multi-level label diffusion method in social networks," *International Journal of General Systems*, vol. 51, no. 4, pp. 352–385, 2022.

[4] M. Sattari and K. Zamanifar, "A cascade information diffusion based label propagation algorithm for community detection in dynamic social networks," *Journal of Computational Science*, vol. 25, pp. 122–133, 2018.

[5] M. Ramezani, A. Khodadadi, and H. R. Rabiee, "Community detection using diffusion information," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 2, pp. 1–22, 2018.

[6] Z. Zhang, J. Wan, M. Zhou, K. Lu, G. Chen, and H. Liao, "Information diffusion-aware likelihood maximization optimization for community detection," *Information Sciences*, vol. 602, pp. 86–105, 2022.

[7] H. Alvari, A. Hajibagheri, and G. Sukthankar, "Community detection in dynamic social networks: A game-theoretic approach," in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pp. 101–107, IEEE, 2014.

[8] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New journal of physics*, vol. 11, no. 3, p. 033015, 2009.

[9] K. Shen, L. Song, X. Yang, and W. Zhang, "A hierarchical diffusion algorithm for community detection in social networks," in *2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 276–283, IEEE, 2010.

[10] M. Jackson and Y. Zenou, *Economic analyses of social networks volume I: Theory*. Cheltenham, UK: Edward Elgar Publishing, 2013.

[11] O. Doluca and K. Oğuz, "Apal: Adjacency propagation algorithm for overlapping community detection in biological networks," *Information Sciences*, vol. 579, pp. 574–590, 2021.

[12] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, "Lpanni: Overlapping community detection using label propagation in large-scale complex networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 9, pp. 1736–1749, 2018.

[13] J. Xie, B. K. Szymanski, and X. Liu, "Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *2011 ieee 11th international conference on data mining workshops*, pp. 344–349, IEEE, 2011.

[14] K. Asmi, D. Lotfi, and M. El Marraki, "Overlapping community detection based on the union of all maximum spanning trees," *Library Hi Tech*, vol. 38, no. 2, pp. 276–292, 2020.

[15] Z. Sun, B. Wang, J. Sheng, Z. Yu, and J. Shao, "Overlapping community detection based on information dynamics," *IEEE Access*, vol. 6, pp. 70919–70934, 2018.

[16] A. Ponomarenko, L. Pitsoulis, and M. Shamshetdinov, "Overlapping community detection in networks based on link partitioning and partitioning around medoids," *Plos one*, vol. 16, no. 8, p. e0255717, 2021.

[17] A. Hollocou, T. Bonald, and M. Lelarge, "Improving pagerank for local community detection," *arXiv preprint arXiv:1610.08722*, 2016.

[18] A. Choumane, A. Awada, and A. Harkous, "Core expansion: a new community detection algorithm based on neighborhood overlap," *Social Network Analysis and Mining*, vol. 10, pp. 1–11, 2020.

[19] N. Kasoro, S. Kasereka, E. Mayogha, H. T. Vinh, and J. Kinganga, "Percomcv: A hybrid approach of community detection in social networks," *Procedia Computer Science*, vol. 151, pp. 45–52, 2019.