Pen-ink Differentiate using Texture fetures for Hand-Written Document forensics

A dissertation submitted to the University of Hyderabad

in partial fulfillment of the degree of

Master of Technology

in

Information Technology

By

Kavali Sreenivasulu

17MCMb12

School of Computer and Information Sciences

University of Hyderabad

Prof. C. R. Rao Road, Gachibowli, Hyderabad - 500 046

CERTIFICATE

This is to certify that the dissertation entitled "**Pen-Ink differentiate using exture features for Hand-Written document forensics**" submitted by **Kavali Sreenivasulu**, bearing Reg. No. 17MCMB12, in partial fulfillment of the requirements for the award of Master of Technology in Information Technology is a bonafide work carried out by him under my supervision and guidance.

The dissertation has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Dr. Rajarshi Pal
Project Supervisor
School of CIS
University of Hyderabad

Dean
School of CIS
University of Hyderabad

DECLARATION

I, D Shiva Shankar hereby declare that this dissertation entitled "**Overlapping Community Discovery in Social Networks**" submitted by me under the guidance and supervision of **Dr. S. Durga Bhavani** is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date: (D Shiva Shankar)

Place: 12MCMI41

To,

My Parents

Acknowledgments

I would like to express my sincere gratitude to **Dr. S. Durga Bhavani**, my project supervisor, for valuable suggestions and keen interest through out the progress of my course of research.

I am grateful to Dean, SCIS for providing excellent computing facilities and a congenial atmosphere for progressing with my project.

I would like to thank **The University of Hyderabad** for providing all the necessary resources for the successful completion of my course work. At last, but not the least I thank my classmates and other students of SCIS for their physical and moral support.

I would also like to thank the **Open Source Community** who provided the free Software and documentation to work with.

With Sincere Regards,

D Shiva Shankar

Abstract

Social networks are modeled as graphs with nodes representing individuals and edges denoting interactions between the individuals. A community is defined as a subgraph with dense internal connections and sparse external connections. Many heuristic algorithms have been proposed in the literature for disjoint community detection. Nodes that belong to more than one community form overlapping regions between communities. In this thesis, we proposed two novel overlapping community discovery algorithms based on the idea of consensus clustering. The first algorithm defines core members of a community as nodes that will be co-clustered by all the algorithms as belonging to one community. The rest of the nodes, unless on the periphery are potentially in the overlapping regions of communities. The second algorithm tries to retrieve overlapping nodes directly based on connectivity consensus. The intuition behind this algorithm is that the neighbours of an overlapping node most probably belong to different communities. Here again the 'overlappingness' of a node is decided based on the consensus arrived at by majority of the community discovery algorithms. The two algorithms are implemented and tested on many benchmark data sets for community discovery. The algorithms successfully detected overlapping nodes that can be verified visually for small networks. Since overlapping node information is not available for benchmark networks, we designed two additional data sets joining friendship networks of two friends using their individual facebook data. It is shown that both the algorithms retrieve friends who have high degree of interactions with many other friends as overlapping nodes. These nodes are not merely those that belong to intersection of two communities but belong to common regions of many hidden communities that are discovered by these algorithms which could not have been inferred easily.

Contents

Chapter 1

Introduction

Social networks are modeled as graphs in which nodes represent the individuals and edges represent interactions between nodes. Many problems have been proposed on social networks regarding influence maximization [1], community discovery [2], influence spread [3] etc. These problems have been found to have many applications to real-world problems like finding the pathways of virus spread, to find influential customers in the online marketing scenario, to find groups of nodes that share common properties etc. In this thesis we are interested in exploring the community discovery algorithms with special focus on overlapping communities. It should be noted that community discovery problem is NP-hard [19].

1.1 Community

A community in a social network is loosely defined as a group of nodes which have dense connections within and sparse external connections with the rest of the graph.

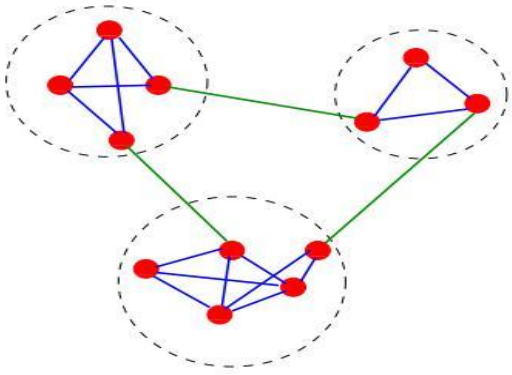

Figure 1.1: A network with community structure [17]

In the above figure we can observe that, there are three communities with dotted lines which are connected densely within the community than outside the community.

1.2 Overlapping Community

Overlapping community [4] can be defined as nodes which belong to more than one community falling in the overlapping regions between communities. To discover these overlapping nodes is the problem of overlapping community detection.

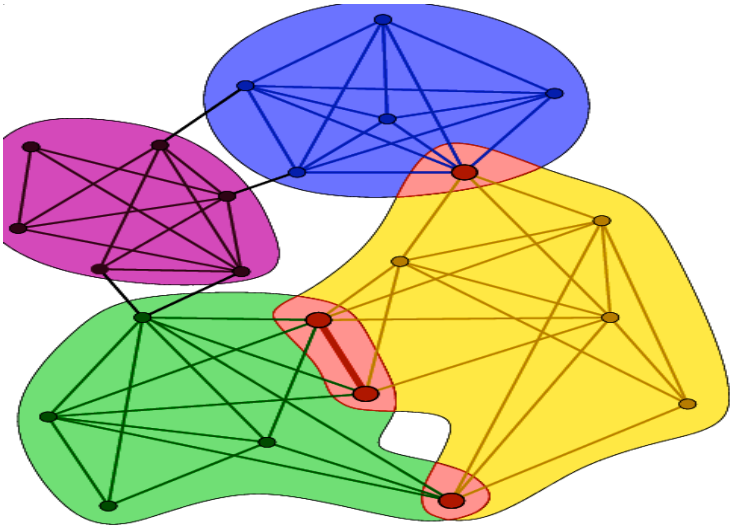

Figure 1.2: Network with communities and overlapping nodes

In the above figure we can observe that there are four communities exists in blue, green, pink and yellow colour boundaries. There are some nodes in red colour, they have approximately equal connections with other communities as well. That means they belong to both the communities, these nodes are considered as overlapping nodes.

1.3 Motivation

The existing algorithms for community discovery in the literature retrieves disjoint communities. On the other hand, in reality, many people (nodes) belong to more than one community. For example, in Social Networks there are groups of people or entities who share some common properties, that may be friendship, organization, research etc. and further who have membership in more than one community. Overlapping community discovery may be used

- to find groups of people having diverse interests.

- to find nodes with connections to different clusters in the network.
- to find groups of pages of World Wide Web which correspond to inter-disciplinary topics.

1.4 Overview of Project Report

In chapter 2, we discuss background and literature survey for community discovery and overlapping community discovery. We described about centrality measures, degree centrality and betweenness centrality, then we discuss measures like modularity, conductance, etc. These parameters are used in community discovery algorithms. We described some of the available community discovery algorithms. Finally we summarize three overlapping community discovery papers briefly.

In chapter 3, we propose two algorithms for overlapping community discovery based on the idea of consensus clustering. First algorithm tries to detect the cores of the communities based on consensus of existing community discovery algorithms in the literature and depicts the rest of the nodes as overlapping communities. The second algorithm proposes that usually overlapping nodes are connected to more than one community and hence analysis of the neighbourhood of a node will be useful in determining if it is an overlapping node. These two algorithms are given and explained using a small benchmark dataset called Zachary karate club network.

In chapter 4, implementation and results of the proposed algorithms are presented on various benchmark datasets and synthetic datasets. We test the algorithms on three benchmark datasets namely, Zachary, Dolphin and Collaboration networks and three synthetic datasets Synthetic dataset, Shiva-Gopi facebook data, Shiva-Swayam facebook network. The construction of these datasets and sources is explained briefly. Results of the algorithms in terms of the number of communities and modularity values are given. Overlapping communities detected by the two algorithms are compared and analyzed.

In Chapter 5, we conclude the thesis and discuss other applications of overlapping community discovery.

Chapter 2

Background and Related Literature

In this chapter we describe a few of the algorithms for community discovery whose knowledge we assume in the dissertation. The implementation of these algorithms is available in a software package called *RStudio* [22]. We also use the package *i*graph available with *RStudio* for carrying out the necessary statistical analysis. A few measures that are commonly used for community discovery algorithms are first defined.

A few algorithms that have been proposed in the literature for discovering overlapping communities are also discussed.

2.1 Centrality Measures

There are two centrality measures along with a measure called modularity that are mostly used for community discovery in social networks. These are described below. Let $G = (V, E)$ containing n vertices be the graph underlying the social network and let adjacency matrix of G be $A = (a_{ij})$.

2.1.1 Degree Centrality

Degree centrality is a measure that involves simply the number of neighbours of a node has or the number of links a node makes in the network. Degree centrality of a node x is denoted as $C_D(x)$, and can be used as an indicator of the measure of a network's interconnectedness.

$$C_D(x) = \sum_{i=1}^{n} (a_{ix})$$

Time complexity of degree centrality for n vertices is $\bigcirc(n)$.

2.1.2 Betweenness centrality

There are two types of betweenness centralities vertex betweenness centrality and edge betweenness centrality that compute the number of times a node or edge respectively act as a bridge along the shortest path between two other vertices. Most commonly used measure is edge betweenness centrality $C_B(xy)$ which is computed as follows:

- For each vertex $v \in V$, compute shortest paths to all the other vertices in V.
- Determine the fraction of shortest paths that pass through an intermediate edge $(u, v) \in E$.

$$C_B(xy) = \sum_{(u,v)\in E} \frac{\sigma_{uv}(xy)}{\sigma_{uv}}$$

where σ_{uv} is the total number of shortest paths from node u to v, and $\sigma_{uv}(xy)$ is the number of shortest paths that pass through the edge (x, y).

Time complexity for betweenness centrality for n vertices and m edges is $\bigcirc(mn)$.

2.2 Conductance and Modularity

Conductance and Modularity [7] are the measures used by most of the algorithms in community discovery. Based on these values a community structure can be decided, how densely or sparsely a community is formed.
Given a set of nodes S, we consider a function $f(S)$ that characterizes how community-like is the connectivity of nodes in S. Let $G(V, E)$ be an undirected graph with $n = |V|$ nodes and $m = |E|$ edges. Let S be the set of nodes, where n_s is the number of nodes in S, $n_s = |S|$; m_s the number of edges in S, $m_s = |(u, v) \in E : u \in S, v \in S|$; and c_s, the number of edges on the boundary of S, $c_s = |(u, v) \in E : u \in S, v \notin S|$; and $d(u)$ is the degree of node u.

2.2.1 Conductance

$$f(S) = \frac{c_s}{2m_s + c_s}$$

measures the fraction of total edge volume that points outside the cluster.

2.2.2 Modularity

Modularity [20] is the fraction of the edges that fall within the given groups minus expected such fraction if edges were distributed at random. The range of modularity value lies in [-1/2,1). If the number of edges within the groups exceeds the number expected on the basis of chance, then the modularity value will be positive.

$$f(S) = \frac{1}{4}(m_s - E(m_s))$$

It measures the difference between m_s, the number of edges between nodes in S and $E(m_s)$, the expected number of such edges in a random graph with identical degree sequence.

2.3 Algorithms for Community Discovery

There are nine community discovery algorithms available in a software package called *RStudio*. These are Edge betweenness, Fastgreedy, Infomap, Label propagation, Leading eigenvector, Multi level, Optimal, Spinglass, Walktrap algorithms. Among these we choose six algorithms for our implementation that are described below.

2.3.1 Algorithm of Girvan-Newman for community discovery

In this algorithm , the edges will be removed in the decreasing order of their edge betweenness scores [6].

Algorithm 1 Girvan-Newman approach ($G\,(V, E)$)

1: Calculate the score of betweenness edges in the network
2: Among all the edge betweenness scores find the highest score and remove that edge from the network
3: Recalculate the scores of betweenness for all remaining edges
4: Repeat the same procedure from step 2 until the graph is disconnected.

2.3.2 Fastgreedy Algorithm

Fastgreedy algorithm is a bottom-up approach. It will optimize a quality function called modularity (defined in section 2.2.2) in a greedy manner. In Fastgreedy algorithm initially each vertex belongs to a separate community, then communities are merged iteratively such that each merge is locally optimal (i.e. yields the largest increase in the current modularity value). The algorithm runs until it is not possible to increase modularity value any more, so it gives grouping as well as dendrogram. It suffers from a problem called resolution limit that means communities below a given size threshold (depending on the number of nodes and edges) will always be merged with neighbouring communities..

Dendrogram: A dendrogram [21] is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering. Dendrograms are often used in computational biology to illustrate the clustering of genes or samples.

2.3.3 Label Propagation Algorithm

Label Propagation algorithm is a simple approach in which every node is assigned one of k labels. The method then proceeds iteratively and re-assigns labels to nodes in a way that each node takes the most frequent label of its neighbors in a synchronous manner. The method stops when the label of each node is one of the most frequent labels in its neighborhood. This algorithm is very fast, but it gives different results based on the initial configuration (which is decided randomly). So that this method should run large number of times (may be hundreds of times) and then build a consensus labeling, which could be tedious.

2.3.4 Leading Eigenvector Algorithm

Leading Eigenvector algorithm is a top-down hierarchical approach that optimizes the modularity function. In each step, the graph is split into two parts in a way that the separation itself yields a significant increase in the modularity. The split is determined by evaluating the leading eigenvector of the so-called modularity matrix, and there is also a condition to stop which prevents tightly connected groups to be split further. Due to the eigenvector calculations involved, it might not work on degenerate graphs where the ARPACK eigenvector solver is unstable. On non-degenerate graphs, it is

likely to yield a higher modularity score than the fast greedy method.

2.3.5 Spinglass Algorithm

In this algorithm each entity (i.e. vertex) can be in one of c spin states, and the interactions (i.e. edges) specify which pairs of vertices would prefer to stay in the same spin state and which ones prefer to have different spin states. The model is then simulated for a given number of steps, and the spin states of the particles in the end define the communities. The steps are as follows,

1) There will never be more than c communities in the end, it is possible to set the c value as high as 200, which will be enough for a purpose.

2) There may be less than c communities at the end, because some of the spin states may become empty.

3) It is not guaranteed that nodes in completely disconnected parts of the networks have different spin states. This is more likely to be a problem for disconnected graphs only, so there is no problem in that.

2.3.6 Walktrap Algorithm

Walktrap algorithm is an approach based on random walks. The idea is that if you perform random walks on the graph, then the walks are more likely to stay within the same community because there are only a few edges that lead outside a given community. Walktrap algorithm runs short random walks of 3-4-5 steps it depends on one of its parameters and uses the results of these random walks to merge separate communities in a bottom-up manner like fastgreedy algorithm. Again the modularity score can be used to select where to cut the dendrogram (defined in section 2.3.2).

2.4 Overlapping Community Discovery

We discuss three recent papers on overlapping community discovery, namely Seed set expansion method, a Nonnegative matrix factorization approach and CONGA approach.

2.4.1 A Method of Seed Set Expansion

One of the recent approaches to finding overlapping communities is that of Wang et al. [8]. We describe this method in some detail here.
This method contains *four* phases, Filtering phase, Seeding phase, Seed set expansion phase and Propagation phase.

- **Filtering Phase:** The goal of the filtering phase is to identify regions of the graph where an algorithmic solution is required to identify the overlapping clusters. In this phase the graph (or network) is divided into clusters based on biconnectivity. A biconnected component with highest number of nodes is considered biconnected core.

- **Seeding Phase:** In seeding phase we find seeds by using two methods, Grauclus centers and Spread hubs. In Graclus centers the graph partitioning is done by high quality and efficient approach with fairly small conductance, that is to produce better boundaries between partitions. In Spread hubs method we find the seeds which are in boundaries with high connectivity with other clusters.

- **Seed Set Expansion Phase:** In seed set expansion phase, we expand the seed sets using a personalized PageRank clustering scheme.

- **Propagation Phase:** Finally, in Propagation phase, we further expand the communities to the regions that were removed in the filtering phase.

2.4.2 A Symmetric Binary Matrix Factorization Approach

Zhang et al. [9] propose overlapping community detection in complex networks using Symmetric Binary Matrix Factorization (SBMF). They used SBMF model for community detection, where each node is assigned to community memberships. This model also distinguishes the outliers from overlapping nodes. The entire task is formulated as constrained non-linear programming model, by adding a penalty term to this optimization model.
In addition to this, they propose a *Partition density* measure to find the number of highly overlapping communities.

2.4.3 CONGA Approach

CONGA (Cluster-Overlap Newman Girvan Algorithm) is an overlapping community discovery algorithm given by Steve Gregory [10], by extending Girvan and Newman's community discovery algorithm based on betweenness centrality measure. In this algorithm network will be partitioned into any desired number of clusters, but allows them into overlap. To allow clusters to overlap, there should be some method to split an item so that it can be included in more than one cluster. In this method they propose a new concept called split betweenness. The notion of "split betweenness" is the key point of CONGA algorithm. This decide a) when to split a vertex instead of removing an edge, b) which vertex to split, c) how to split it. The main idea of the algorithm is to obtain communities by removing edge with maximum edge betweenness or split vertex with maximum split betweenness whichever is greater.

We propose two new overlapping community discovery algorithms based on the idea of consensus clustering, the details of which are given in the next chapter.

Chapter 3

Proposed Algorithms for Overlapping Community Discovery

We adopt the approach of Consensus Clustering to detect overlapping communities. In the context of clustering of nodes, different ways of clustering of nodes may be obtained by different algorithms. Also, as an internal parameter like K in K-Means algorithm is changed, different clusterings are obtained. In order to choose a single clustering that agrees most with the other clusterings, the idea of consensus clustering was proposed by [12].

Consensus clustering is an NP-hard problem. There are many approximation algorithms proposed in the literature like Best-of-K, Majority rule, Best One Element Move, average linkage etc [12].

3.1 Motivation

We adopt these ideas behind consensus clustering to the context of overlapping community discovery. Since existing community discovery algorithms attach a community label to each node, can we check if majority of algorithms agree that a pair of nodes belongs to a specific community then those two nodes belong to that community. The nodes (of degree > 1) for which the algorithms do not form a consensus are considered to be overlapping nodes.

3.2 Algorithm based on Consensus Clustering

A node in a social network either may be embedded within the core of a community or the nodes may be either on the periphery of a community or may belong to more than one community. In order to distinguish the location of a node, we propose the Algorithm 2, in which we define the core of a community as nodes that are co-clustered by all the algorithms as belonging to one single community. If algorithms do not have consensus on cluster labels of two nodes, then there exists some ambiguity regarding the membership of these nodes. These nodes are potentially in the overlapping regions if they are not isolated or on the periphery (of degree 1).

Algorithm 2 Overlapping community discovery using consensus clustering.

Input: Community labels obtained from k community discovery algorithms on a graph $G = (V, E)$, $|V| = n$, $|E| = m$

Output: overlapping nodes.

read the input file and save as $A(i, j)$ =community labels given to node i by the algorithm j
$M(i, j) = 0$ for each $i, j = 1, 2, ..., n$
for $l = 1, 2, ..., k$ **do**
  if $A(i, l) = A(j, l)$ **then**
    $count(i, j) + +$
  end if
  if $count = k$ **then**
    $M(i, j) = 1$
  end if
end for
for $i = 1, 2, ..., n$ **do**
  $G_i = \{i\}$
  for $j = 1, 2, ..., n$ **do**
    if $M(i, j) = 1$ **then**
      $G_i = G_i \cup \{j\}$
    end if
  end for
end for
for $i = 1, 2, ..., n$ **do**
  if $length(G_i) = 1$ && $degree(i) > 1$ **then**
    node i is overlapping node
  end if
end for

Time complexity:

In this algorithm it has to check labels of each node with all other nodes for all the given algorithms that takes nk time, and this checking should happen for all the remaining nodes also with all other nodes. Total time will be taken by this algorithm is $\bigcirc(n^2k)$.

3.2.1 Implementation on Zachary karate club

Zachary Karate club dataset is a graph with members of a university karate club as nodes and interactions between the members as edges and has been collected by Wayne Zachary. It has 34 nodes and 78 edges. We choose seven community discovery algorithms for computing the consensus on clustering. The algorithms of Edge betweenness, Fastgreedy, Infomap, Multi level, Leading eigenvector, Optimal, Walktrap algorithm available in a software package called *RStudio* are executed. We have not considered Label propagation algorithm and Spinglass algorithm since they are not found to be stable and community labels are getting changed in different executions. A snapshot of the community labels assigned to the nodes by these algorithms is given in Table 3.1.

Node	EBC	FG	INF	LEV	ML	OPT	WT
1	1	1	1	2	1	2	1
2	1	3	3	2	1	2	1
3	2	3	3	2	1	2	2
4	1	3	3	2	1	2	1
5	3	1	1	1	2	3	5
6	3	1	1	1	2	3	5
7	3	1	1	1	2	3	5
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
34	4	2	2	4	3	1	3

Table 3.1: Zachary nodes & its community labels

In the above table first column represents node numbers of Zachary network and remaining seven columns represent community labels of that algorithm.

EBC: Edge betweenness centrality algorithm

FG : Fastgreedy algorithm

INF: Infomap algorithm

LEV: Leading eigenvector algorithm

ML : Multi level algorithm

OPT: Optimal algorithm

WT : Walktrap algorithm

Actually there exists only two disjoint communities in the Zachary network in the ground truth [7].

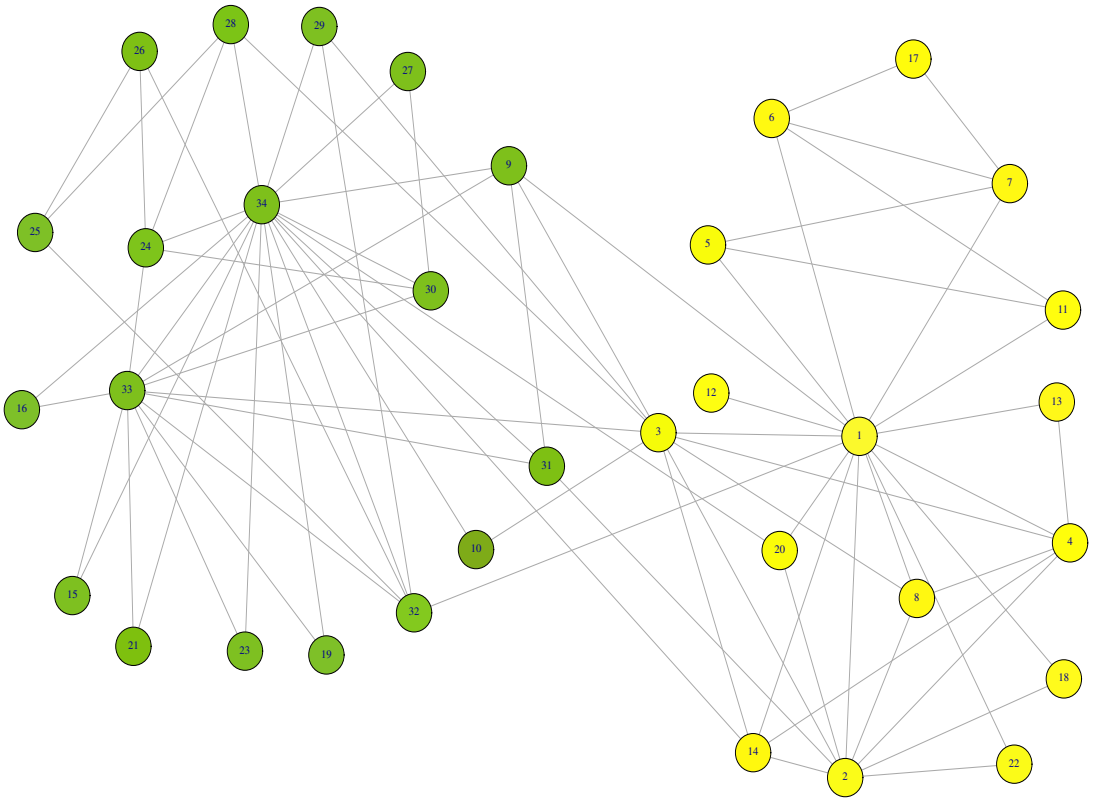

Figure 3.1: Zachary network with two communities represented in green colour and yellow colour [18]

In the above figure we can see two communities with green colour and yellow colour. There are no overlapping nodes as of now, but in some of the papers says that there are two nodes which are in overlapping they are, node 3 and node 10.

3.2.2 Zachary network figures for various community discovery algorithms

In this section we look at seven figures [18] of Zachary network with different communities coloured differently. Each figure is an output of different community discovery algorithm that is mention in the figure caption.

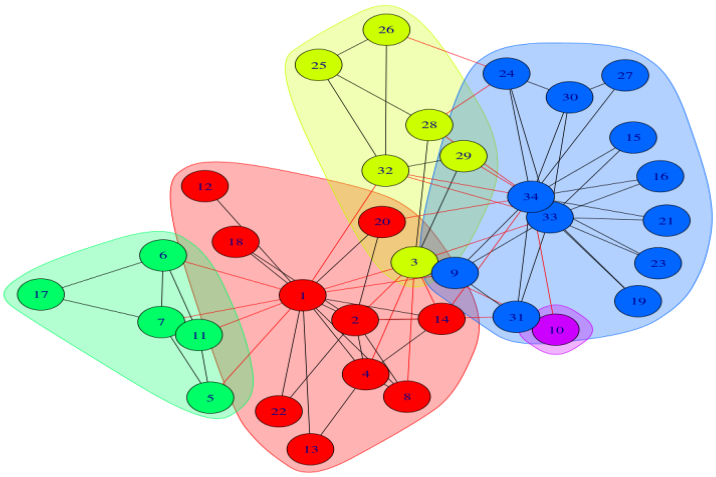

Figure 3.2: Zachary network with communities according to EBC

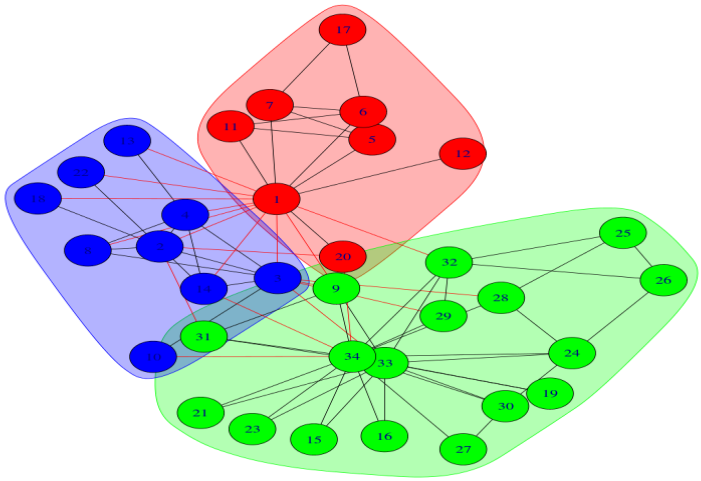

Figure 3.3: Zachary network with communities according to FG

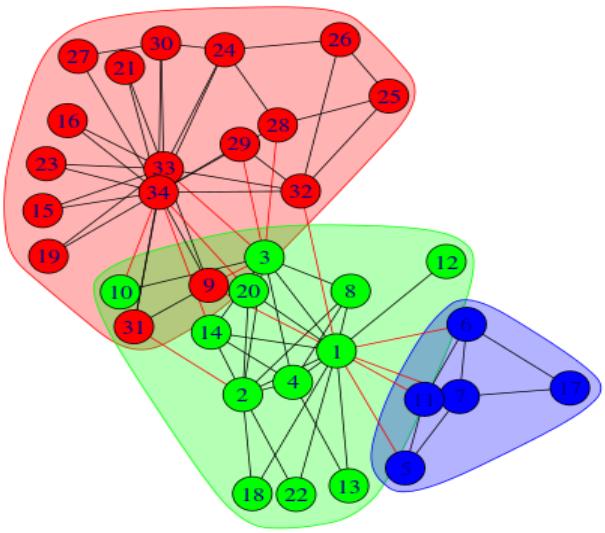

Figure 3.4: Zachary network with communities according to INF

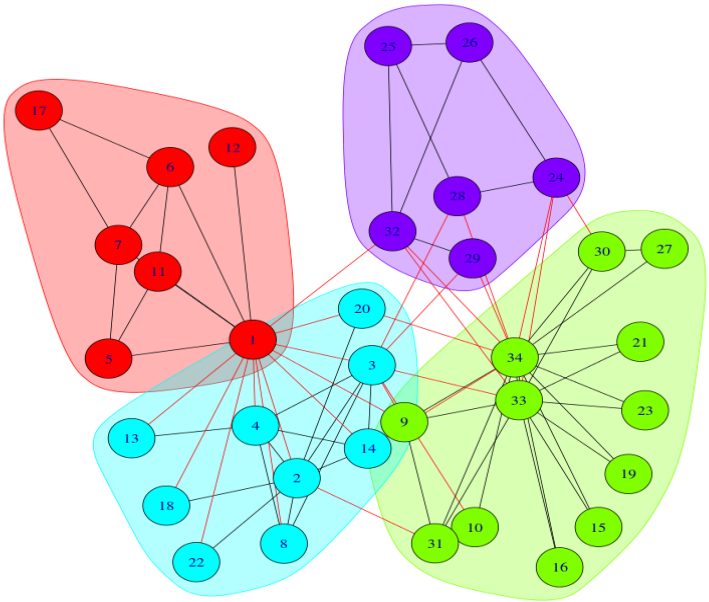

Figure 3.5: Zachary network with communities according to LEV

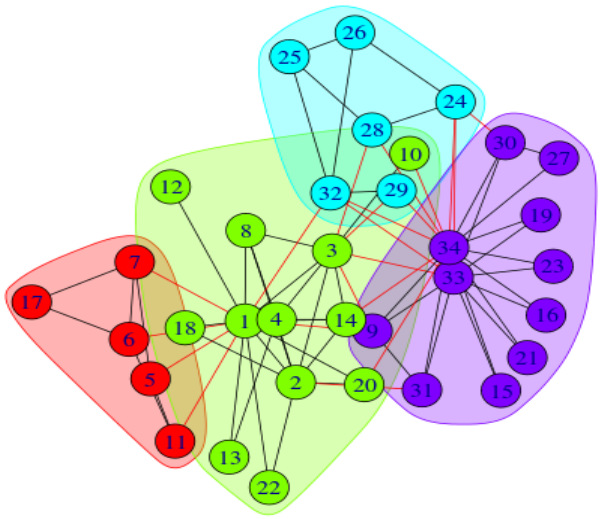

Figure 3.6: Zachary network with communities according to ML

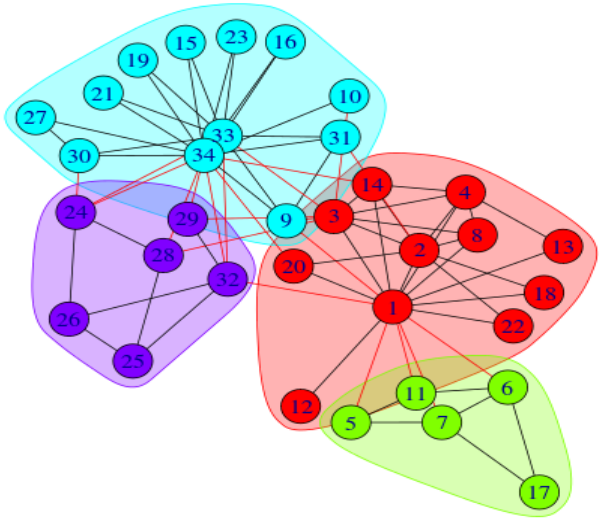

Figure 3.7: Zachary network with communities according to OPT

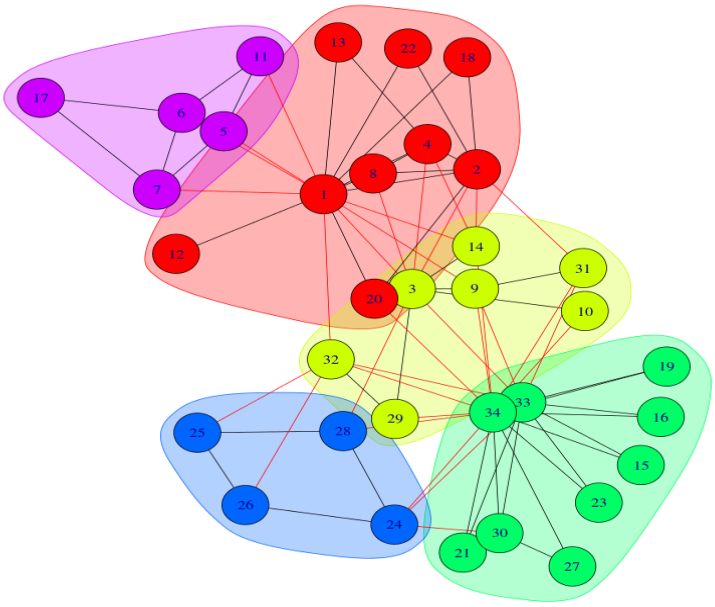

Figure 3.8: Zachary network with communities according to WT

As it is explained in the algorithm 2, the input file contains data in the format of *nodes* × *algorithms*. Consider nodes one and two, their labels in FG and INF is different and remaining all other algorithms are the same. It means nodes one and two are co-clustered in five algorithms out of seven algorithms. In the same way nodes one and three are co-clustered in three algorithms out of seven algorithms. Consider nodes five, six and seven whose labels in all the algorithms are same. This means that all these three nodes are co-clustered by all algorithms (i.e. 100% of the algorithms) and hence can be considered as 100% consensus. In the same way each pair of nodes considered and checked whether they are co-clustered in all algorithms or some percentage of algorithms (could be adjusted based on the requirement). All these co-clustered pairs will be written into a file and make groups of all the nodes which are co-clustered. If any node is not co-clustered with any other node and its degree is greater than one then this node will be considered as an overlapping node. This algorithm 2 is implemented on Zachary network and five nodes (3,10,14,20,24) emerge as overlapping nodes, because some of the algorithms divide this network into more than two communities that is shown in the section. Note that in Figure 3.9 these five nodes can be seen clearly to be overlapping nodes.

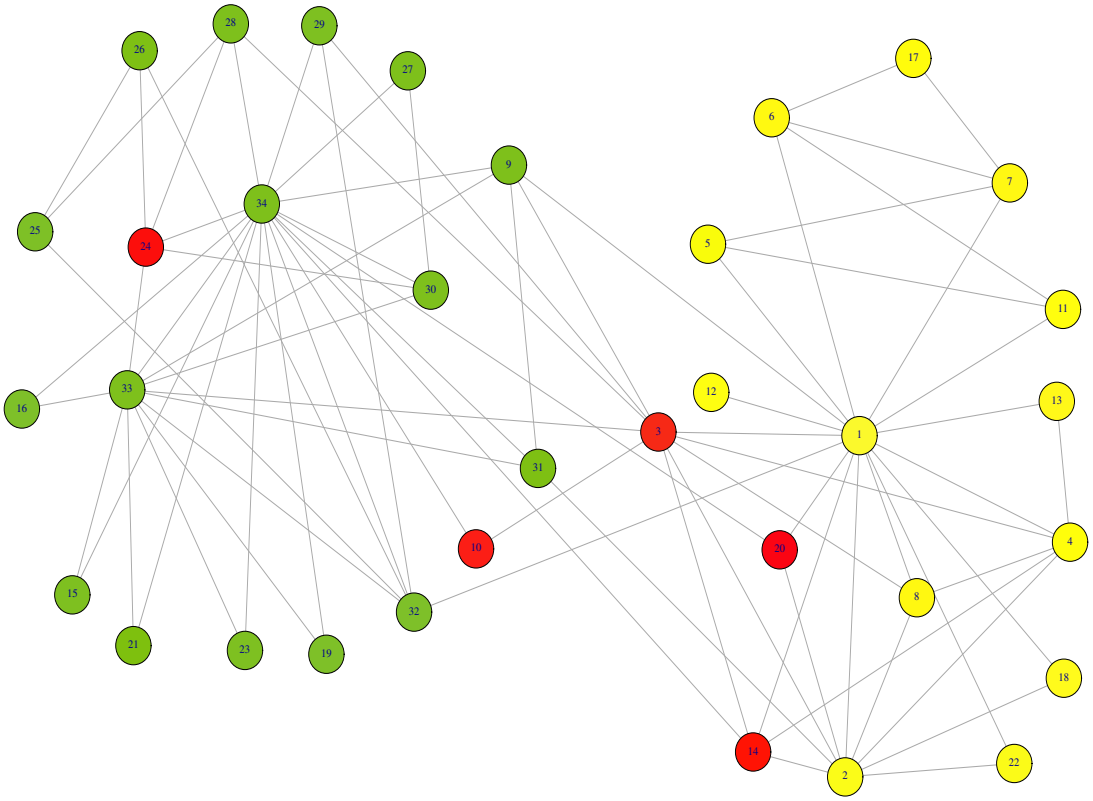

Figure 3.9: Zachary network with five overlapping nodes represented in red colour according to the above algorithm 2

In the above figure we can observe that there are five overlapping nodes (3,10,14,20,24) represented with red colour. The image source is [18]

3.3 Algorithm based on Connectivity Consensus

Unlike the previous algorithm which detects the cores of communities and annotates the rest as overlapping, here we detect directly the overlapping nodes. The intuition behind this algorithm is that an overlapping node is most probably connected to neighbours that belong to different communities. Here again the 'overlappingness' of a node is decided based on the consensus arrived at by majority of the algorithms. Zachary network nodes and its community labels are described in the section 3.2.1.

Step1: For a fixed algorithm for a node a if majority of its neighbours of node a belong to different communities.

Step2: If a node a and its neighbours do not belong to the same community by majority of the algorithms then it is to be considered that the node a is overlapping

node.

Algorithm 3 Overlapping community discovery using connectivity consensus.

Input: graph $G(V, E)$ edge list and a file consists community labels of each algorithm.
Output: overlapping nodes.

1: $|V|= n$, $|E|= m$, k = number of algorithms and $edge(i, j)$ = edge between nodes i and j.
2: calculate degree of each node.
3: **for** $i = 1$ to n **do**
4: $c = 0$, $count_1 = 0, count_2 = 0, ..., count_k = 0$.
5: **for** $j = 1$ to n **do**
6: **if** $edge(i, j) = TRUE$ **then**
7: **for** $l = 1$ to k **do**
8: **if** $label_l(i) != label_l(j)$ **then**
9: $count_l++$
10: **end if**
11: **end for**
12: **end if**
13: **end for**
14: **for** $l = 1$ to k **do**
15: **if** $degree(i) > 1 \&\& count_l >= \lceil degree(i)/2 \rceil$ **then**
16: $c++$
17: **end if**
18: **end for**
19: **if** $c >= \lceil k/2 \rceil$ **then**
20: print i.
21: **end if**
22: **end for**

Time complexity:

In this algorithm it has to check labels of all neighbours of one node. A node can have a maximum of $n - 1$ neighbours. To check one node's all neighbours takes $n - 1$ time, this checking should happen for all the nodes and also for all the algorithms. Therefore total time taken by this algorithm is $\bigcirc(n^2k)$.

3.3.1 Implementation on Zachary karate club

As explained in the algorithm, input file contains data in the format $nodes \times algorithms$. Take every individual node and check the community labels of all its neighbours for a fixed algorithm. If a minimum of 50% of its neighbours have different community labels (i.e. not co-clustered) then this node is potentially an overlapping node. Now check for all the remaining algorithms for consensus. Finally, among these, only the nodes whose degree is greater than one are considered to be overlapping nodes.

This approach when implemented on Zachary network gave again the five nodes (3,10,24,28,32) as overlapping nodes. Note that in Figure 3.10 these five nodes can be seen clearly to be overlapping nodes.

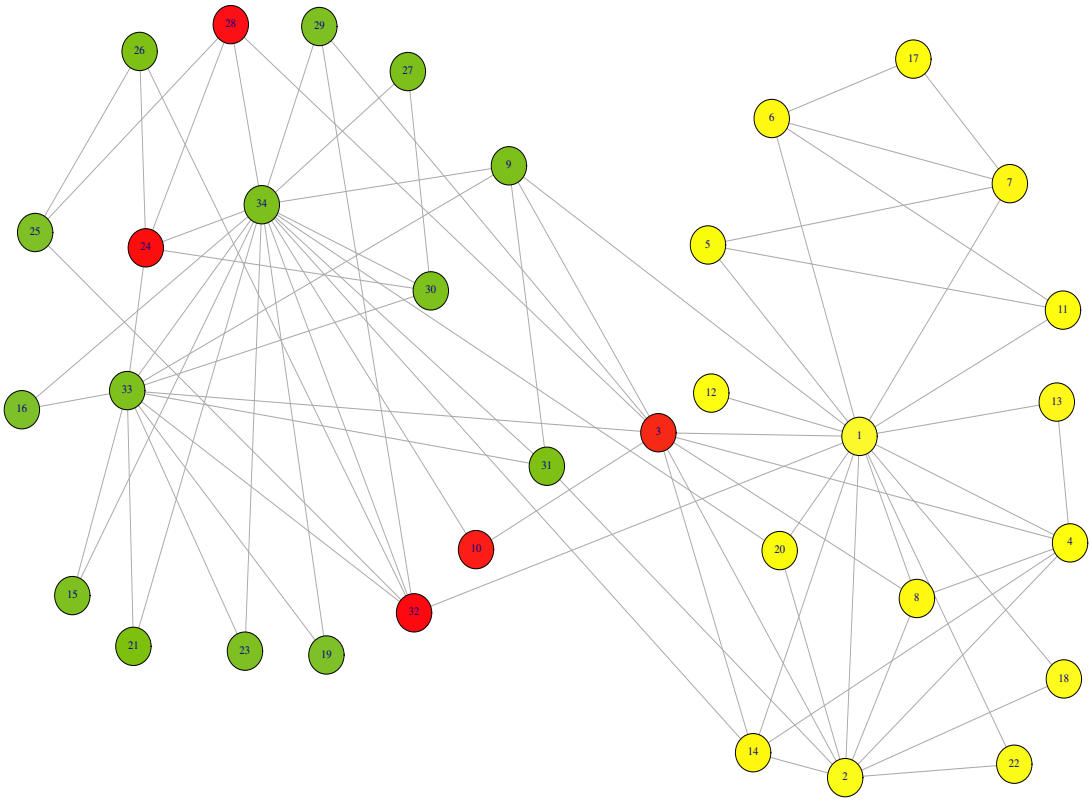

Figure 3.10: Zachary network with five overlapping nodes represented in red colour according to the above algorithm [18]

In the above figure we can observe that there are five overlapping nodes (3,10,24,28,32) represented with red colour.

3.3.2 Conclusion

Two novel algorithms for overlapping community discovery have been proposed in this chapter based on the idea of consensus clustering. As a preliminary checking, the algorithms have been implemented and tested on a small but popular benchmark data set called Zachary karate club which is known to have two communities. It should be noted that there are no benchmark data sets available for checking overlapping communities. Visually we can see that at least two nodes 3 and 10 may be considered to be overlapping for Zachary karate club. We would like to check this hypothesis with our algorithms.

The Algorithm 2 gave (3,10,14,20,24) and Algorithm 3 gave (3,10,24,28,32) as overlapping nodes of the communities. Both the algorithms detect 3 and 10 as overlapping nodes satisfying the prediction. Newman's edgebetweenness algorithm outputs four communities as seen in figure 3.2, with all the seven nodes 3, 10, 14, 20, 24, 28 and 32 detected by the two algorithms in the overlapping regions of a pair of communities. 3 and 10 are clearly between the major communities but 24 seems to be in the overlapping region of communities coloured by blue and yellow. If we observe in the figures 3.2-3.8, the communities of LEV, ML, OPT and WT node number 24 is in the boundary of that community and sharing majority of its connection with other communities as shown by four out of seven algorithms. From this explanation we can conclude that three nodes (3, 10, 24) can certainly be designated as overlapping.
The algorithms do give satisfactory results. We need to implement these algorithms on larger bench mark data sets for validation which is carried out in the next chapter.

In the next chapter we implement these algorithms on three benchmark datasets and three synthetic datasets.

Chapter 4

Implementation and Results

Implementation of the proposed overlapping community discovery algorithms, Algorithm 2 & algorithm 3 is carried out on three benchmark social network datasets namely Zachary, Collaboration and Dolphin. Though analysis of Zachary network has been done in the previous chapter, we place it here with more details and for completeness sake. Since overlapping information is not available for any of these data sets, we construct two data sets from individual facebook accounts where we know who are common friends using the author's facebook network data along with two of his friends' facebook network data. One synthetic dataset named as Synthetic dataset is also being used. These total six datasets are used for the analysis of the proposed algorithms given in the section 3.2 & section 3.3

4.1 Datasets

The datasets being considered for this study are summarized in Table 4.1 which are described in some detail in the following section.

Dataset	Nodes	Edges
Zachary network	34	78
Collaboration network	5242	14496
Dolphin	63	157
Synthetic dataset	350	6215
Shiva-Gopi facebook data	919	24279
Shiva-Swayam facebook data	1603	40203

Table 4.1: Benchmark & Synthetic datasets considered

4.1.1 Benchmark Datasets

- **Dataset1: Zachary network** [13]

  Description: Zachary dataset contains information regarding interaction among members of a university karate club given as two 34×34 matrices, as a binary matrix and a weighted matrix. Binary matrix represents the presence of ties among the members of the club. Valued matrix represents the relative strength of the associations that is number of situations in and outside the club in which interactions occurred. In the year 1977 Zachary used these data and an information flow of network conflict resolution to explain the split-up of this group followin disputes among the members.

- **Dataset2: Collaboration network** [14]

  Description: Collaboration network contains author collaborations with regard to General Relativity and Quantum Cosmology Arxiv GR-QC covering papers submitted submitted to General Relativity and Quantum Cosmology category in the period from January 1993 to April 2003 (124 months). If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes.

- **Dataset3: Dolphin** [15]

  Description: Dolphin is an undirected graph depicting frequent associations between 62 (as per reference paper) dolphins in a community living off Doubtful Sound, New Zealand.

4.1.2 Synthetic Datasets

- **Dataset4: Synthetic dataset [16]**

  Description: Synthetic dataset has been designed [16] in such a way that the network contains a community structure. This graph has four types of edges. We do not consider the type and hence remove all parallel edges. This pruned network has 350 nodes and 6215 edges.

- **Dataset5: Shiva-Gopi facebook data.**

- **Dataset6: Shiva-Swayam facebook data.**

  Description: Recently Facebook has made it possible for the data to be downloaded using a facebook app called *netvizz*. Datasets 5 and 6 are collected by combining data of two facebook user accounts of the author Shiva Shankar Danthuri with data of his friends Gopi Tadaka and Swayam Prakash respectively. The downloaded facebook data is in file.gdf format. Its data contains two parts, one is friends list that is in the form of

  $nodedef >$ *name* VARCHAR, *label* VARCHAR, *sex* VARCHAR,
  locale VARCHAR, *agerank* INT

  and another is edgelist that is in the form of

  $edgedef >$ *node*1 VARCHAR, *node*2 VARCHAR.

4.2 Results on Benchmark Datasets

The algorithms available on *RStudio* are executed for all the datasets which give modularity coefficient and number of communities as the output.

- **Zachary network**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.4012985	5
Fastgreedy	0.3806706	3
Leading Eigenvector	0.3934089	4
Multi Level	0.4188034	4
Optimal	0.4107143	4
Infomap	0.4020381	3
Walktrap	0.3532216	5

Table 4.2: Zachary network: number of communities & modularity values

On Zachary network Edge betweenness and Walktrap algorithms output five communities as shown in Figures 3.2.2, remaining all other algorithms give three

or four communities. The node 10 is considered as a separate community by Edge betweenness algorithm. In Walktrap algorithm there are three communities formed with $8, 6, 9$ nodes, these nodes are forming into two communities in other algorithms. These nodes are forming into two communities of size $10, 12$ by adding one more node that is node 10 which is considered a separate community in the Edge betweenness algorithm.

Algorithm	Number of overlapping nodes
Consensus Clustering	5 (3,10,14,20,24)
Connectivity Consensus	5 (3,10,24,28,32)

Table 4.3: Overlapping nodes for Zachary network

It is observed that in the above table there are five nodes in overlapping region in both the algorithms. Among these five nodes three nodes named $3, 10, 24$ are common in both the algorithms, so we can say these three nodes satisfy consensus over consensus. The overlapping nodes have good number of connections to other communities as well.

- **Collaboration network**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.8490437	433
Fastgreedy	0.8137622	415
Leading Eigenvector	0.7840748	406
Multi Level	0.860752	392
Infomap	0.7943545	717
Walktrap	0.7823643	815

Table 4.4: Collaboration network: communities & modularity values

Since it is a big collaboration network with 5242 nodes and 14496 edges, it takes more time to execute in *RStudio*. Table 4.4 shows the different number of communities obtained by the different algorithms. Walktrap algorithm gives highest number of communities 815 and ML gives 392 communities with highest modularity coefficient. All the algorithms give modularity value of approximately 0.80.

Algorithm	Number of overlapping nodes
Consensus Clustering	389
Connectivity Consensus	335

Table 4.5: Overlapping nodes for Collaboration network

In the above table consensus clustering algorithm gives 389 overlapping nodes by considering 100% consensus (i.e all the algorithms). This value changes according to the consensus percentage. In case of connectivity consensus there are 335 overlapping nodes, this output is according to 50% of an individual node connectivity with consensus of 50% of algorithms. This value could be changed when we change the consensus either in connectivity or in algorithms.

There are 193 common nodes among these 389 and 335 nodes given by two algorithms 2 & 3. F rom this we can conclude that there are 193 nodes which are completely overlapping which satisfy the properties of both the algorithms.

- **Dolphin dataset**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.4392876	15
Fastgreedy	0.4657187	4
Leading Eigenvector	0.4986815	5
Multi Level	0.5212382	5
Infomap	0.5152745	6
Optimal	0.5252546	5
Walktrap	0.5002637	7

Table 4.6: Dolphin network: communities & modularity values

From Table 4.6, we can observe that Edge betweenness algorithm gives highest number of communities 15, other algorithms gives not more than 7. Edge betweenness algorithm outputs four communities with single node they are nodes numbers 5, 12, 13, 36 and three communities with two nodes they are nodes (33, 61), (47, 50) and (54, 62). Walktrap algorithm gives two communities having two nodes each and these are (33, 61) and (47, 50). Except for Edge betweenness and Walktrap algorithms no other algorithm forms communities with one node

or two nodes. These one node or two node communities are merged with other communities by the other algorithms. We could observe that all the modularity values are approximately 0.50.

Algorithm	Number of overlapping nodes
Consensus Clustering	9 (1,3,21,29,31,37,40,53,55)
Connectivity Consensus	10 (1,8,9,20,21,29,37,40,41,51)

Table 4.7: Overlapping nodes for Dolphin network

From Table 4.7, we can see that there are 9 nodes overlapping according to Consensus clustering with 100% consensus. This value changes when consensus percentage changes. According to the algorithm of Connectivity consensus there are 10 nodes that are overlapping. Both the algorithms agree on five common nodes 1, 21, 29, 37, 40 to be overlapping. According to the results of our algorithms we can conclude these five nodes to be certainly in the overlapping region.

4.3 Results on Synthetic Datasets

- **Synthetic dataset**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.3943921	4
Fastgreedy	0.3953974	3
Leading Eigenvector	0.3925157	3
Multi Level	0.3953667	3
Infomap	0.3953974	3
Walktrap	0.3953974	3

Table 4.8: Synthetic dataset: communities & modularity values

Edge betweenness algorithm gives four communities and remaining all the other algorithms are giving three communities for this Synthetic data set which is designed to have 3 communities confirming the definition of the data set in Section 4.1.2. Edge betweenness algorithm has a single node 310 as a separate

community. We can observe that modularity value in all algorithms is stable at approximately 0.3953.

Algorithm	Number of overlapping nodes
Consensus Clustering	1 (i.e. 310)
Connectivity Consensus	0

Table 4.9: Overlapping nodes for Synthetic dataset

It is highly interesting to note that for the synthetic data set designed to have 3 well-formed communities, the proposed algorithms also detects almost no overlapping nodes. In Table 4.9 we can observe that there is only one overlapping node according to Consensus clustering approach (with 100% consensus) because Edge betweenness algorithm considers the node 310 as a separate community. There is no overlapping node according to Connectivity c onsensus approach since the data set is a synthetic data set designed to have three disjoint communities.

From this we can observe that the Synthetic dataset which is designed to have three well-formed communities, has very little chance to have any overlapping nodes. In fact this shows that the Algorithms 2 and 3 are both working very well on the synthetic data set.

- **Shiva-Gopi facebook data**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.311986	92
Fastgreedy	0.3895095	7
Leading Eigenvector	0.414189	13
Multi Level	0.4264074	7
Infomap	0.3270177	21
Walktrap	0.3892167	56

Table 4.10: Shiva-Gopi facebook network: communities & modularity values

Shiva-Gopi facebook data set has 919 nodes and 24279 edges as given in Table 4.1. The author Shiva Shankar Danthuri has 301 friends and his friend Gopi Tadaka has 630 friends. There should be a total of 931 (i.e. 301 + 630) nodes

by combining these two user networks, but we mentioned 919 nodes in the table 4.1 because there are 12 common friends in both accounts.

Among all algorithms, Edge betweenness algorithm gives highest number of communities (i.e. 92), since very high degree nodes exist, for example, 4 nodes having degrees 630, 403, 334, 330; 18 nodes whose degrees are more than 200 and there are 123 nodes whose degrees are 100 and more. We actually combined two networks with 301 nodes and 630 nodes, but the community discovery algorithms extract more than two communities because there are some nodes whose degrees exceed the number of nodes of one network.

Algorithm	Number of overlapping nodes
Consensus Clustering	74
Connectivity Consensus	69

Table 4.11: Overlapping nodes for Shiva-Gopi facebook network

In Table 4.11, we can see that there are 74 overlapping nodes by using Consensus clustering approach. In this table, Consensus clustering approach gives that there are 74 overlapping nodes because community discovery algorithms are dividing the whole network into a large number of communities instead of dividing into only two communities. This happens because there are some nodes which are densely connected in the network, their degrees are more than 50% of nodes in one a ccount data, these nodes are also forming communities. That is the reason there are large number of communities formed in the network leading to large number of overlapping nodes. In case of Connectivity approach there are 69 overlapping nodes.

Out of these nodes, we see that the number of common nodes is 39 which can be considered to be truly overlapping agreed by both the algorithms 2 & 3. These 39 nodes share their connections approximately equally with the other communities as well as within the community. When we analyzed further, we find that not all of the 12 common friends belong to this overlapping region. Only two of these friends are found to be overlapping nodes by the first algorithm. We find that the reason for this fact is that common friends unless they are of high degree, that is have interactions with many other common friends do not come out in the

overlapping region. Hence these algorithms do retrieve non-trivial nodes which satisfy the intuition that friends having connections to many communities as overlapping.

- **Shiva-Swayam facebook data**

Algorithm	Modularity (best split)	Number of Communities
Edge Betweenness	0.6523698	198
Fastgreedy	0.6293061	10
Leading Eigenvector	0.653447	8
Multi Level	0.6712994	7
Infomap	0.6464744	47
Walktrap	0.6668866	35

Table 4.12: Shiva-Swayam facebook network: communities & modularity

In Shiva-Swayam facebook data there are 1603 nodes and 40203 edges. The user Shiva Shankar Danthuri has 301 friends and the other user Swayam Prakash has 1384 friends having 82 common friends.

Among all algorithms, Edge betweenness algorithm gives highest number of communities (i.e. 198), because there are 254 nodes whose degrees are 100 and more than 100. These nodes form new communities with its neighbourhood nodes, so that community discovery algorithms give large number of communities instead of two communities.

Algorithm	Number of overlapping nodes
Consensus Clustering	107
Connectivity Consensus	100

Table 4.13: Overlapping nodes for Shiva-Swayam facebook network

In Table 4.13 we can see that there are 107 overlapping nodes by using Consensus clustering approach and 100 nodes using the second algorithm. There are 41 common nodes among these 107 and 100 nodes given by two algorithms 2 & 3.

These 41 nodes share connections approximately equally with the other communities as well as within the community. In fact out of the 82 common friends only

one node is obtained as overlapping by both the algorithms. Similar to analysis carried out earlier, we see that the overlapping nodes obtained by our algorithms are of very high degree having many connections with friends from different communities and hence truly belong to the overlapping regions of different communities.

4.4 Conclusion

The two algorithms on overlapping community discovery tackle the problem from two different points of view: one considering nodes outside core communities as overlapping; the other detecting nodes that have connections with more than one community as overlapping. It is expected that the results are different and also it is interesting to see that there is a non-trivial intersection between the predictions. That is there is a set of overlapping nodes that is arrive d at on consensus by both the algorithms. The common nodes given by both the algorithms can be considered to be in the overlapping region with a high certainty since they are sharing approximately equal connections with the same community and other communities (i.e. approximately half of their connections are with the outside of the community).

In the case of benchmark datasets of Zachary karate club and Dolphin network, we can observe that the overlapping nodes commonly given by both the algorithms 2 & 3 do belong to the overlapping region as can be seen visually since these networks are comparatively smaller than the other datasets.

In Collaboration network and three synthetic datasets, the interesting result of obtaining no overlapping nodes for Synthetic dataset is very satisfactory as the data set is designed to have three disjoint communities. The facebook results of Shiva-Gopi facebook dataset and Shiva-Swayam facebook datasets conform to the common friendships as well as nodes that may be friend-of-friend having high degree of interconnections.

It is of course necessary to validate the algorithms further on any bench mark data sets that are designed and constructed for overlapping community discovery which are not yet available to the best of our knowledge.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

Overlapping community detection in social networks finds several applications: to find the people from different places who are gathered at a common venue; in social network analysis; in discovering the total structure of the communities etc.

In this thesis, we proposed two novel overlapping community discovery algorithms based on the idea of consensus clustering. Different clustering algorithms give different ways of partitioning the same data set and taking consensus of these algorithms would give a new clustering of the data that maximizes on the clustering quality of different algorithms. In social networks, community discovery is an important problem and many algorithms have been proposed in the literature. Most of these algorithms assume disjoint communities and divide the data forcefully putting some points into one of the communities. On the other hand, individuals of diverse interests and web pages corresponding to inter-disciplinary areas belong to more than one community and detecting these overlapping nodes is an interesting problem.

We approach the problem of identifying these nodes from two points of view. If we discover the nodes that lie within a core of a community, then the remaining nodes will either belong to more than one community or simply on the periphery of a community, a kind of a recluse loosely belonging to a community, not interacting with any one in particular. Based on this idea we discover core members of communities based on the consensus given by the existing community discovery algorithms. Another way of approach is to directly detect nodes that are interacting with many nodes of more than

one community. Again apply consensus to see if these nodes are considered overlapping by majority of the algorithms.

In order to implement our algorithms, we need to run the existing community discovery algorithms from the literature. For this purpose, we choose the implementations available in the software tool called *RStudio* (igraph package) of the seven community discovery algorithms, namely Edge betweenness algorithm, Fastgreedy algorithm, Infomap algorithm, Leading eigenvector algorithm, Multi level algorithm, Optimal algorithm and Walktrap algorithm.

Our proposed algorithms are implemented on several bench mark datasets. On the small data sets, the overlapping nodes are visually seen and detected correctly by our algorithms. As a negative compliance, we tested on a synthetic data set that is designed to have only disjoint communities and our algorithms do give a result of nearly empty overlapping region. Finally, we design two data sets using the App provided by Facebook in order to investigate the efficacy of our algorithms. We join two facebook accounts of two friends to form a friendship network and expect two communities to be detected. But the presence of many high degree nodes which are individuals having high interactions with many different people make the network to be having many small communities. Hence guessing at overlapping nodes is not straightforward. The two networks Shiva-Gopi and Shiva-Swayam turn out to be very interesting giving out 39 and 41 nodes to be overlapping by both the algorithms 2 and 3. When we analyzed these nodes, we see that these correspond to people having high degree and connections with many communities justifying them to be overlapping nodes.

5.2 Future Work

Overlapping community discovery is crucial to understanding the network structure. Hence this study needs to be integrated with the community discovery as a whole. The entire presentation can be made rigorous by giving definitions for overlapping nodes and proving that the algorithms actually detect the overlapping nodes as per the definition. The algorithms proposed have a limitation in terms of the time-complexity. The algorithms crucially depend on implementing several community discovery algorithms instead of tackling the problem directly. On the other hand, this approach seems to be very robust and may be used to build bench mark data sets

for overlapping community discovery where ground truth information is not available. Imposing 100% consensus seems to be too hard a constraint. Experimentation with different threshold values of consensus gives varying results. These issues need to be investigated further.

Bibliography

[1] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2009.

[2] M. E. J.Newman, Detecting community structure in networks. *European Physical Journal* B 38: 321-330(2004).

[3] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.

[4] G. Palla, I. Dereényi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," Nature, http://dx.doi.org/10.1038/nature 03607, vol. 435, no. 7043, pp. 814-818, 2005.

[5] M. Girvan and M. E. J. Newman, Community structure in social and biological networks. *Proc. Natl. Acad. Sci.USA* 99, 7821–7826 (2002).

[6] U. Brandes, A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* 25, 163–177(2001).

[7] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *ICDM*, (2012).

[8] J. J. Whang, D. F. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *CIKM*, (2013).

[9] Zhang Z-Y, Wang Y, Ahn Y-Y (2013) Overlapping community detection in complex networks using symmetric binary matrix factorization. Physical Review E 87: 062803. doi: 10.1103/physreve.87.062803

[10] S. Gregory. An algorithm to find overlapping community structure in networks. In PKDD, 2007.

[11] B. Yan and S. Gregory, Detecting community structure in networks using edge prediction methods,*Journal of Statistical and Mechanical*, P09008 (2012).

[12] A. Goder and V. Filkov. Consensus clustering algorithms: Comparison and refinement. In ALENEX '08: Proceedings of the Workshop on Algorithm Engineering and Experiments. SIAM, 2008.

[13] Zachary W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33, 452-473.

[14] J. Leskovec, J. Kleinberg and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.

[15] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten and S. M. Dawson, Behavioural ecology and sociobiology 54, 396-405 (2003).

[16] L. Tang, X. Wang, and H. Liu. Community detection via heterogeneous interaction analysis. Data Mining and Knowledge Discovery, 25:1–33, 2012.

[17] Images are taken from google images.

[18] Images are drawn in a software package called *RStudio*.

[19] `http://www.dcs.fmph.uniba.sk/~fduris/VKTI/3color_HamCycle.pdf`

[20] `http://en.wikipedia.org/wiki/Modularity_%28networks%29#cite_note-config-3`

[21] `http://en.wikipedia.org/wiki/Dendrogram`

[22] `http://www.rstudio.com/`

[23] `http://en.wikipedia.org/wiki/Deep_learning`