Detecting Communities in Networks and Performance Prediction Based on Relation Strength Measurement

by

Soom Satyam Behera

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements

for the M.A.Sc degree in

Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

Complex networks is an interdisciplinary research area which focuses on the study of properties of complex systems that have many functional or structural subunits. Community detection algorithms are one of the major approaches to analyse complex networks with multilevel or overlapping community structures. This research work focuses on constructing a novel community detection approach for simplification of a given complex demographic network.

The general process of the abstraction from concrete problems as well as the general definition of communities have not been well defined and all the existing methods are derived from specific backgrounds, leaving the reliabilities in other fields open to question. This specificity of the existing methods reveals the need for a general approach for community definition and detection. Here, we devise a general procedure to find community structures in concrete problems by classifying the concrete networks into two basic types: Transmission networks and Similarity networks. The relation among nodes in transmission networks are constructed by material transmission and the ones in similarity network are constructed by the similarity in properties of the nodes. We show that both the types can be represented based upon an unified graph model. Based on the model, we propose a generic approach, Relation Strength Measurement (RSM), to define the communities.

We have demonstrated that the Effective Resistance Function (ERF), from the Klein and Randic's electrical network model, is applicable for quantifying the relation among nodes. We have also introduced a community threshold parameter (CP) based on which, the RSM algorithm categorizes the network nodes into communities. We have compared the performance of our algorithm with other well known community detection methods. The simulation results show that the algorithm accurately obtains the division of community structure both in real-world and synthetic networks.

Acknowledgements

This thesis has been a wonderful journey. I take this opportunity to thank Faculty of Graduate and Postdoctoral Studies, University of Ottawa for giving me this opportunity.

I am immensely thankful to Dr. Amiya Nayak, my thesis supervisor, for being kind and supportive. His constant guidance during the process has helped me think out of the box. His critical evaluation has always put me on the right tracks towards completion of this thesis. I value this association with him and consider myself lucky to have received his esteemed guidance.

I would like to express my deepest appreciation and sincere thanks to Haoye Lu for providing indispensable information, advice and support throughout this journey.

I convey my gratitude to colleagues of my former organization and Akanksha Singh for her continuous guidance and immense support, without whom this journey would not have started in the first place. I give my sincere regards to Mohammed Rehaan and Mayank Kumar for their words of encouragement and well wishes.

I would like to thank my father Mr. Sarat Chandra Behera for supporting me financially through the course, my mother Mrs. Purnima Behera for her blessings and emotional support. I appreciate their patience during my absence in the family.
A special thanks to my roommates Harsha and Hari for being my family away from home.

As in the words of Paulo Coelho, Roy shows me every day to be happy for no reason, to be curious and to strive tirelessly.

Contents

List of Tables

List of Figures

List Of Acronyms

CP	Community Detection Parameter
EEG	Effective Edges Graph
ERF	Effective Resistance Function
KR	Karate Club Network
NMI	Normal Mutual Information
OI	Omega Index
PFR	Pairwise Precision F-Score and Recall
P2P	Peer 2 Peer Network
RSMFSN	Relation Strength Measurement for Similarity Network
RSMFTN	Relation Strength Measurement for Transmission Network
RSM	Relation Strength Measurement
SDF	Shortest Distance Function
STT	Shortest Transmission Time
TTS	Theoretically Transmission speed

Chapter 1

Introduction

Detection of communities in real-world topologies such as large social networks, web graphs, and biological networks is a problem of considerable practical interest that has received a great deal of introspection [3][4][5]. A "network community" (also sometimes referred to as a module or cluster) is typically thought of as a group of nodes with more and/or better interactions amongst its members than between its members and the remainder of the network [5][6].

To find such cluster of nodes, one generally picks up a target function that captures the above instinct of a group as an arrangement of nodes with preferable inside integration over the connectivity with the outer network. At that point, following the goal is commonly NP-difficult to optimize exactly [5][7], one uses heuristics [8][9] or approximation algorithms [10] to discover such set of nodes that that more or less optimizes the target function and that can be comprehended or interpreted as real communities.

However, one might presume communities operationally to be the result of a community detection procedure, owing to the fact that they bear some relationship to the intuition as to what it means for a set of nodes to be a strong community. Although there is no such ubiquitous definition of what a community is, it can be viewed as a set of nodes that has a shared identity or a frequent interaction to each other than others.

Modules or communities are intermediate-level structures between the microscopic level of nodes and the macroscopic panorama of the whole network [11]. Once extracted, such clusters of nodes are often interpreted as organizational units in social networks, functional units in biochemical networks, ecological niches in food web networks, or scientific disciplines in citation and collaboration networks [12].

We stress upon the fact when we find the communities within a certain topology is the division of it in certain sets so that the edges appear within a confined set more often than that across other groups. But this boils down with an important argument that if the node belongs to the the clusters which has equal number of edges, then which cluster does it belong to?

Usually in the real-world scenario, we have a big picture of what a complex network is but we don't figure out the individual groups, modules or communities underlying it. The objective of the community detection algorithm is to find the modules or communities of a network using only the interactions between its members, which may be a (un)weighted and (un)directed graph. This again raises a question if the graph partition does the same then why the need for a community detection method? For the former method we are aware of the modules in advance we need to partition the graph into while for the latter, we are not aware of it. We need to find the number of modules and the partition of the network into modules at the same time. Of course, this leads to different applications. Graph partitioning is used to match a graph to external criteria such as a set of processors. Community detection is used to find the natural structure of a graph.

Our real world consists of elements and relations among them. In academia, we usually call the entity made by entities and relations among them as network. Real world networks are not random, as they display big inhomogeneities, revealing a high level of order and organization [5]. In particular, several elements have much stronger inner relations comparing to the ones with other nodes in the network. For example, people in

a company may have much stronger relations than the ones outside the company. This observation inspires people to group elements that have strong inner relations together and derives the definition of community structures. Communities have lots of concrete applications. Amazon tries to group customers buying similar books together and make more reasonable recommendations. Facebook groups users by relationships, hobbies, etc., in order to help users find the people they are interested in. UPS groups locations that have large volume of trade among them and assign proper number of transports. Because of the wide-spread application of community structures, researchers try to find proper algorithms to detect them. As of today, lots of algorithms have been derived from various backgrounds.

The traditional methods [13] include: graph partitioning [14][15] , hierarchical clustering [3] and spectral clustering [16, 17]. These methods are designed for different purposes and reveal many fundamental properties of complex networks. After that, many related algorithms are also derived (e.g. modularity-based methods [18–20], dynamic algorithms [21], methods based on statistical inference [22, 23], etc.).

We will start from the discussion of two main types of networks. One is based the material transmission and the other one is based on the nodes similarity. We show that both two types can be deduced to the same graph model. After that, we give a general definition of community structures based on the absolute relation strengths among nodes. Furthermore, we discuss some propositions about this definition and give an algorithm to detect the community structures. Finally, we give a complete demonstration to show how the algorithm works.

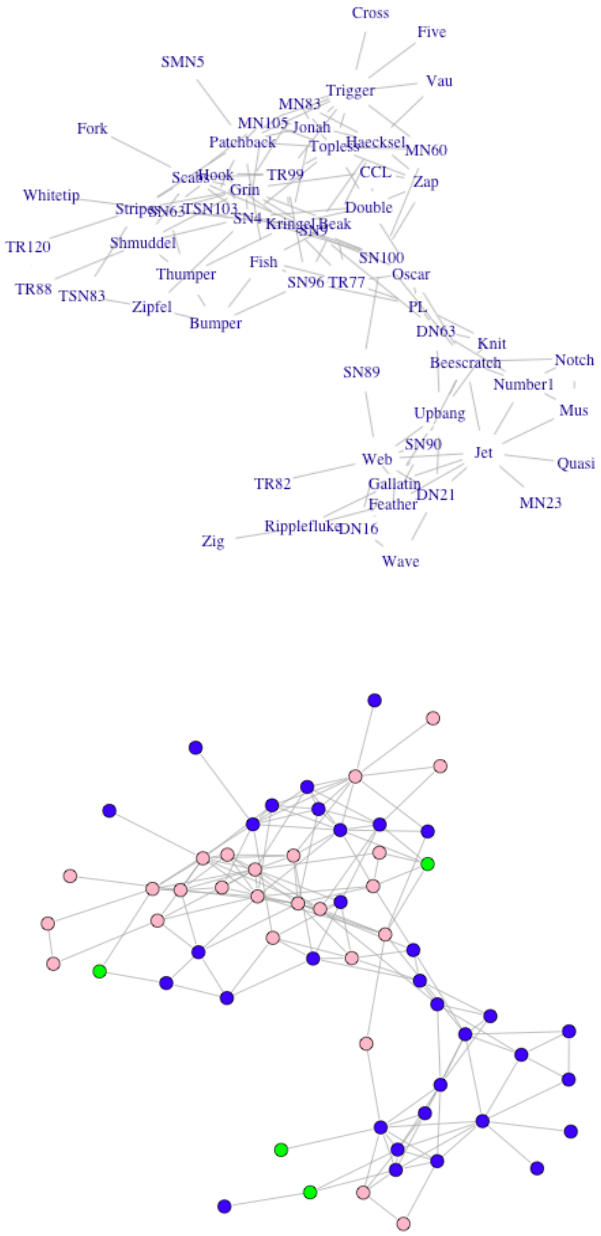


Figure 1.1: Social Network of bottlenose dolphins in Doubtful Sound, New Zealand [1]; each node Id represents an individual member of the network and each edge represents a pair that was observed in the same school more often than expected by chance. The sex determination observed were pink (female), blue (male), green (unknown). Most of the links around 70% connect dolphins of the same sex.

1.1 Motivation

Network analysis is a daunting task for a wide range of complex networks possessing different dimensional network topologies. Based on the topology, the properties of the network may vary. Although there has been substantial research in this field, the existing works have not dealt adequately with the problem of network classification based on their attributes. Particularly, the definition and the detection of communities within the complex networks needs to be focused upon since it's application areas range from fields like Epidemiology to Social networking.

Most community detection algorithms work pretty well within the constrained environments for which they are designed. But the reliabilities in other areas are still open to question. The specificity of algorithms reveals the demand for general community detection methods as well as the general community definitions. Besides, most algorithms exhibit ambiguity in the general process of the abstraction from concrete problems modelled based on real-world networks. The scope of this research includes seeking answers to the following questions:

1. Is there a common way to reduce a concrete problem into an unified mathematical model?

2. Based on the model, are there some common properties shared by most community definitions?

3. Can a generalized approach be adopted to detect community structures?

4. How does the generalized approach compare with other attribute-centric approaches?

These queries have been a key motivation for pursuing the current state of the art research carried out in community detection algorithms.

1.2 Thesis Objectives

The following objectives are set for this thesis:

1. Analyze complex networks and formulate an approach for defining communities.

2. Design and implement an uniform network generator using which community strength can be determined.

3. Propose an algorithm for detecting maximal communities and implement it on a network where no explicit community structure is available.

4. Evaluate the performance of the proposed algorithm and compare it to existing community detection methods. The means to achieve the goals of this thesis are presented in the rest of the chapters.

1.3 Contributions

The major contributions of this research are the following:

- Abstraction of a real-world network to a concrete problem and construction of a mathematical graph model.

- Formulation of a general community detection approach to measure the relationship among the individual members and identify the maximal community structures.

- Demonstration of the applicability of Klein and Randic's Electrical Network Model for relation measurement under the proposed approach for detecting communities.

- Benchmarking of the proposed algorithm and detailed analysis of the performance comparison with other community detection algorithms.

1.4 Organization

The organization of the thesis is as follows:

- Chapter 2 presents our survey of complex networks and the various community detection algorithms along with the current state of the art research in this area.
- Chapter 3 categorizes the complex networks based on their attributes and properties followed by important definitions and corollaries for defining a community. This chapter also includes the discussion on the proposed Relation Strength Measurement algorithm and the associated mathematical proofs.
- Chapter 4 includes a discussion on community evaluation using performance parameters. This chapter also involves the performance comparison and analysis of our algorithm with other community detection algorithms.
- Chapter 5 presents the summary and conclusions of this research work. The future possible extensions of the research have also been put forward.

Chapter 2

Background and Related Work

The community detection methods are always presented from an author's perspective which emphasizes on the speed, performance and the cost of the process. There is no such unique definition of what community is, instead the idea of communities is different and has been evolving depending on the field that defines it [24]. The reason being we have so many different algorithms existing. So under these scenario, comparing the performances of the different algorithms which has targeted different fields has less relevance.

Giving a complete literature review about complex networks, its metrics, measures and community detection algorithms is out of the scope of this study. We would like to refer the interested reader to the more comprehensive and popular reviews and books [13]. In this section, we are going to explain some important milestones of the community-detection algorithms from our subjective point of view and also elaborate how these selected algorithms constitute the base of our research plan.

The research of community detection starts from some concrete problems. For example, Kernighan-Lin algorithm [14] is designed for partitioning electronic circuits onto boards (the nodes contained in different boards need to be linked to each other with the least number of connections) and hierarchical clustering algorithm is derived from the

background that researchers try to find multilevel structure of the graph.

With the better understanding of network structures, researchers found that the concrete network in the real-world could always be abstracted into the graph model, which is a representation of a set of objects where some pairs of objects are connected by links.

The links here represents the relations among the objects. Although the meanings of them are various in different papers, we find that there are two main types: material transmissions among objects and object's property similarity.

The material here can represent concrete objects (like goods) or information (like data packages). One of the most typical example should be the transportation among cities. The cities that can easily communicate with each others are usually grouped in one community [25].

There are also many network structures constructed by node's property similarity. In protein-protein interaction networks, communities usually group proteins having the same specific function within the cell [26]. So the link here is the protein's function. In social network, the people show up in similar location and time might be considered as a community. Then the relation here is the similarity of people's locations and schedules. Moreover, in World Wide Web, the community may correspond to groups of pages dealing with the same or related topics [27, 28].

Based on the graph model, many community detection algorithms have come up which we will be discussing further.

2.1 Graph partitioning

Graph partitioning method divides the vertices in g groups of predefined size, such that the number of edges lying between the groups is minimal. Many algorithms belonging to this method can perfectly solve many concrete problems (For example, Kernighan-Lin

algorithm has good performance in minimising the number of connections among the nodes contained in different boards). However, the algorithms are not good for community detections, because it is necessary to provide as input the number of groups and in some cases even their sizes, about which in principle one knows nothing [13]. Also, the graph partitioning method is not derived from some explicit definition of communities. So, there is no guarantee that the vertex groups found by the method are communities following our intuitions.

2.2 Hierarchical clustering

In some cases, the graph may have hierarchical structures. Namely, there are several levels of clusters of vertices. Hierarchical clustering is a method of community analysis which seeks to build a hierarchy of clusters. The strategies for hierarchical clustering generally fall into two types: agglomerative and divisive.

Agglomerative approach is "bottom up". Each observation starts in its own cluster, and pairs of cluster are merged as one moves up the hierarchy. Divisive approach is "top down".

1. *Agglomerative algorithms*, in which clusters are iteratively merged if their similarity is sufficiently high;

2. *Divisive algorithms*, in which clusters are iteratively split by removing edges connecting vertices with low similarity.

Hierarchical Clustering is one of the traditional methods used to find the communities. Being an agglomerative approach, it starts assuming each node as a different cluster and finds similarities between the every pair of nodes included in that network. Then the nodes are merged forming the communities ranging from highest to the lowest score [5]. It continues computing similarity with the newly formed group with the rest of the cluster of nodes. There are basically 3 main approaches to find the similarity between

nodes, nodes and clusters and between clusters.

1. The single-linked method uses the highest similarity score of two nodes between two groups.

2. The complete-linked method uses the lowest similarity score of two nodes.

3. The average-linked method takes the average of all similarities between groups [13].

Eventually, each of these approaches will give us a network of the nested set of communities. Cosine similarity is used to find the similarity between the nodes by not only finding the neighbours common between the two nodes i and j but also normalizing it n_{ij}. It produces values that ranges between 0 and 1. If both nodes have the same neighbors $\theta_{ij} = 1$, and if they have no common neighbors $\theta_{ij} = 0$ [29].

Although there is no best choice between one of the methods (single, average and complete) but it affects the quality of clustering the method is employed.

All observations starts in one cluster, and splits are performed recursively as one moves down the hierarchy. Any hierarchical algorithm starts from the definition of similarity function for any pair of vertices in a graph. In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), the algorithm also requires a measure of dissimilarity between sets of observations (linkage function). The method is applicable for all graphs. However, the results might makes no sense since the graph at hand may not have a hierarchical structure at all. Besides, vertices of a community may not be correctly classified, and in many cases some vertices are missed even if they have a central role in their clusters [30].

The major drawback of this type of clustering is that the similarity metrics give higher scores to the edges more central to the communities than the edges less central. It results that the node that are connected with a single link to the rest of the network are added later and thus result in the formation of the singleton communities [5][13].

2.3 Divisive Algorithms

The divisive algorithms are based on the idea of the between centrality, first proposed by Freeman [31]. Girvan & Newman proposed the popular edge-betweenness metric to find the edges on the boundaries [5]. The algorithms tries to find the edges that are at the boundaries of the communities instead which are at the core of the communities. The betweeness of an edge is measured by the number of the shortest paths that traverse through it. It first counts the number of edges that passes through each edge. It is quite likely that if there is an edge on the boundary of the communities there will be many shortest path routes along it. Thus they will have a higher betweenness score. If an edge is at the core of a community, it will have a low betweenness value as the shortest paths between pairs of vertices are likely to use other edges. This property distinguishes inter-community edges, which link many vertices in different communities and have high betweenness, from intra-community edges, whose betweenness is low.

Eventually the whole network which is considered as a single community at first results in dendrogram by the deletion of the edges one by one from the highest edge-betweenness value to the lowest one. The leaves of the dendrogram are individual nodes.

The drawback of those algorithms is that it is NP-complete and not suitable for the smaller networks with lesser number of nodes (~ 1000). Moreover no effective greedy algorithm has been developed for this method, which makes this algorithm impractical. Also everytime a link is removed, the centralities needs to be recalculated which requires a large amount of computer power and, for a network of size n with m links, the speed of calculating all link betweenness in one step still remains $O(m^2n)$ for unweighted networks [2].

2.4 Modularity

In an extension to the above shortest path centrality, Girvan and Newman presented two different approaches to calculate the betweenness centrality. In the first method, the network is looked upon as an electric circuit where each edge between two nodes are assigned an unit resistance and two nodes are selected that we define as unit voltage source and sink. We obtain a measure like that of centrality by calculating the current flow by the Kirchoff's laws. Those links with the lowest resistance (shortest path) carry the most current and, therefore, are the most central. The same approach is used by Wu-Huberman which is discussed further in this section. While the second approach employs random walks to calculate the betweenness centrality of the links. The network is used as a substrate for signals that perform a random walk between pairs of nodes. The link betweenness in this case is simply the rate of flow of random walkers through a particular link summed over all pairs of vertices. The drawback of these approaches are they require higher computation and don't improve the accuracy when compared to the earlier centrality method.

When a network is partitioned into different communities, the problem is looked upon as an evaluation of how good the partition is. Girvan and Newman [5] proposed a simple approach. Considering an arbitrary partition of a given network into N_c communities. We can define a $N_c \times N_c$, size matrix where the elements represent the fraction of total links starting at a node in partition i and ending at a node in partition j. Then, the sum of any row of $a_i = \Sigma_j e_{ij}$, corresponds to the fraction of links connected to i.

The expected value of the fraction of the links can be calculated by within partitions can be estimated. The probability that a link begins at a start of node i say a_i multiplied by the fraction of links that end at a node in j say a_j. The expected number of community links is given as $a_i \times a_j$.The real fraction of links exclusively within a partition is e_{ij}. Comparing the two and summing over all the partitions in the graph we get

$$Q = \sum_{i=1}^{c} (e_{ij} - a_{ij}) \tag{2.1}$$

This is known as modularity function. To illustrate with an example, if we have two partitions, corresponding exactly to the two components, modularity will have a value of 1. As we had discussed earlier the drawbacks of hierarchical clustering being NP-complete and no effective greedy algorithm has been found out yet. So Newman considered this approach as one of the standalone community detection method as it was able to limit those drawbacks [5].

A greedy algorithm is used to optimize the value of Q. Starting from a configuration where each node corresponds to one community, the authors computed all the changes in modularity obtained by joining any possible pair of nodes. The highest increment is selected and the two communities are joined, and the process is repeated until a maximum value of Q is obtained. This method used is really fast and the recalculation of the increments only uses local information. It can analyze a network in almost linear time. However, the accuracy achieved is the lowest of all the modularity optimizing methods. The resolution limit is also the drawback of this methods. It works well for the modules which are similar in size and degree [29].

2.5 Partitional clustering

Partitional clustering is another popular class of methods to find cluster in a set of data points. In this method, the number of clusters is given by the user, say k. The points are set in a metric space, so that each vertex is a point and a distance measure is defined between each pair of the points in the space. The distance is a measure of dissimilarity between vertices. The goal is to separate the points in k clusters such to maximize/minimize a given cost function based on distances between points and/or from

points to centroids (suitably defined positions in space). The limitation of partitional clustering is the same as that of the graph partitioning algorithms: the number of clusters must be specified at the beginning, but the method is not able to derive it [13].

2.6 Spectral clustering

Spectral clustering includes all methods and techniques that partition the set into clusters by using the eigenvectors of matrices. In particular, the objects could be points in some metric space, or the vertices of a graph. Spectral clustering consists of a transformation of the initial set of objects into a set of points in space, whose coordinates are elements of eigenvectors: the set of points is then clustered via standard techniques, like k-means clustering [13][32]. Although there are many successful applications in image segmentation and machine learning, researchers have already found several fundamental limitations of this method. For example, when confronted with clusters of different scales, corresponding to a multiscale landscape potential, standard spectral clustering which uses the first k eigenvectors to find k clusters will fail [33].

2.7 Methods to find overlapping Communities

Most of the methods discussed in the previous sections aimed at detecting standard partitions, i.e. partitions in which each vertex is assigned to a single community. However, in real graphs vertices are often shared between communities and the issue of detecting overlapping communities has become quite popular in the last few years. We devote this section to the main techniques to detect overlapping communities.

The idea of overlapping communities says that a particular node can belong to various "thematic" communities (i.e. one can belong to a scientific group, a family, a sports team), which usually share a certain amount of nodes, a common scenario. The

methodology to find the overlapped communities is based on the concept of 'k-clique communities'. We discuss here about maximal cliques as we have used in our algorithm to find the maximal communities and would like the readers to know about it. A clique in an undirected graph $G = (V, E)$ is a subset of the vertex set C $\subseteq$ V, such that for every two vertices in C, there exists an edge connecting the two.

Maximum Clique is a clique of the largest possible size in a given graph and cannot be extended by including one more adjacent vertex. A k-clique is a group of nodes that is a complete subgraph, and a 'k-clique community' is the union of all k-clique that are adjacent (two k-cliques are adjacent if they share (k - 1 nodes) [2]. It has certainly interesting and useful applications, i.e. it can be used to observe the level of relationship between communities or to determine the communities where a certain node belongs.

2.7.1 Clique Percolation

The clique percolation method builds up the communities from k-cliques, which correspond to complete (fully connected) sub-graphs of k nodes. A community is defined as the maximal union of k-cliques that can be reached from each other through a series of adjacent k-cliques. Such communities can be best interpreted with the help of a k-clique template (an object isomorphic to a complete graph of k nodes). Such a template can be placed onto any k-clique in the graph, and rolled to an adjacent k-clique by relocating one of its nodes and keeping its other (k - 1 nodes) nodes fixed. Thus, the k-clique communities of a network are all those sub-graphs that can be fully explored by rolling a k-clique template in them, but cannot be left by this template.

This definition allows overlaps between the communities in a natural way, as illustrated in Fig. 2.1, showing four k-clique communities at k = 4. The communities are color-coded and the overlap between them is emphasized in red.

The definition above provided is a local one i.e. if a certain sub-graph completes the criteria to be considered as a community, then it will remain as a community independent of the changes in other parts of the network. On the contrary, when searching for the communities by optimizing with respect to a global quantity parameter, a change far away in the network can reshape the communities in the unperturbed regions as well. Furthermore, it has been shown that global methods can suffer from a resolution limit problem, where the size of the smallest community that can be extracted is dependent on the system size. A local community definition such as here circumvents this problem automatically.

Since even small networks can contain a vast number of k-cliques, the implementation of this approach is based on locating the maximal cliques rather than the individual k-cliques [34]. Thus, the complexity of this approach in practice is equivalent to that of the NP-complete maximal clique finding, in spite of the fact that finding k-cliques is polynomial. This suggests that although networks with few million nodes have already been analyzed successfully with this approach, [35] no prior estimate can be given for the runtime of the algorithm based simply on the system size. CFinder is a software tool for finding and visualizing overlapping dense groups of nodes in networks, based on the Clique Percolation Method (CPM). Despite their conceptual simplicity, one may argue that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structure in a specific network.

2.7.2 Line Graph and Link Partitioning

The idea of link partitions instead of nodes to discover community structure has also been investigated. A node in the original graph is called overlapping if links connected to it are assigned in more than one cluster.

In Ahn et al. [24], links are partitioned via hierarchical clustering of edge similarity.

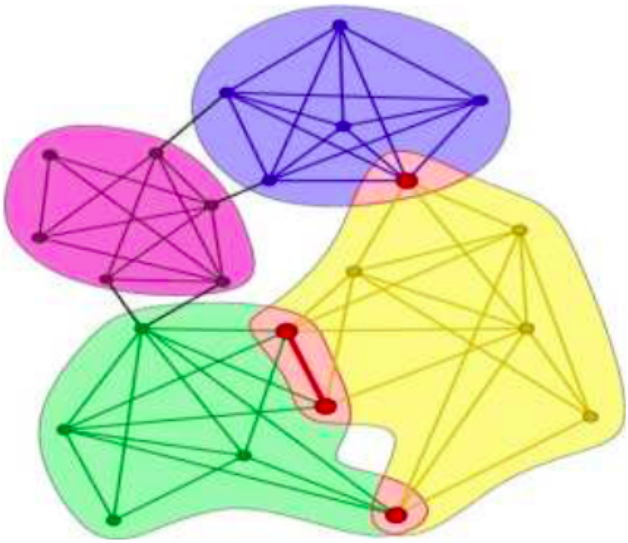


Figure 2.1: Clique Percolation Method. The example shows communities spanned by adjacent 4-cliques. Overlapping vertices are shown by the bigger dots[2]

.

Given a pair of links e_{ik} and e_{jk} incident on a node k, a similarity can be computed with the *Jaccard index* defined as

$$K(e_{ik}, e_{jk}) = \frac{N_i \cup N_j}{N_i \cap N_j} \tag{2.2}$$

where N_i is the neighborhood of node i including i. Single-linkage hierarchical clustering is then used to build a link dendrogram. Cutting this dendrogram at some threshold yields link communities. The time complexity is $\mathrm{O}(nk^2{}_{max})$, where k_{max} is the maximum node degree in the network.

Line graph has been extended to clique graph, where in cliques of a given order are represented as nodes in a weighted graph [36]. The strength of a node's membership i to community c is given by the fraction of cliques containing i which are assigned to c. Although the link partitioning for overlapping detection seems conceptually convincing, there is no guarantee that it provides higher-quality detection than node-based detection

does because these algorithms also rely on an ambiguous definition of community.

An extension to the link modularity was proposed by Nicosia et al. [37]. This measure is built on the belonging coefficients of links.

2.7.3 Expansion and Optimization

Algorithms that utilize the local expansion and optimization are based on growing a natural community Lancichinetti et al. [38] or a partial community. Most of them use a local benefit function that characterizes the quality of a densely connected group of nodes.

The Rank Removal Method proposed by Baumes et al. [39] is a 2-step process. The process iteratively removes highly ranked nodes until small, disjoint cluster cores are formed called as seeds which serves as the second step of the process, Iterative Scan (IS), that expands the cores or seeds by adding or removing nodes until a local density function cannot be improved. The density function can be defined as

$$f(c) = \frac{b_{in}^{c}}{b_{in}^{c} + b_{out}^{c}}, \tag{2.3}$$

where b_{in}^{c} and b_{out}^{c} are internal and external weights of the nodes of the community c. The Iterative Scan result in disconnected components in some cases. The reason being, a modified version called CIS by Kelly et al. [40] was introduced wherein the connectedness is checked after each iteration. In the case that the community is broken into more than one part, only the one with the largest density is. CIS also develops a new density function

$$f(c) = \frac{b_{in}^{c}}{b_{in}^{c} + b_{out}^{c}} + \beta \tag{2.4}$$

The parameter β optimizes the algorithm's performance in sparse areas of the network. It maintains a balance caused due to the addition of a node that results the change in the internal degree density and the change in edge density.

Lee et al. [41] proposed GCE which identifies maximum cliques as seed communities. It expands these seeds by greedily optimizing a local fitness function [41]. GCE also removes communities that are similar to previously discovered ones using distance between communities $c1$ and $c2$ defined as:

$$1 - \frac{|c1 \cap c2|}{min(|c1|, |c2|)} \tag{2.5}$$

If this distance is shorter than a certain parameter defined, the communities are similar. The time complexity for greedy expansion is $O(mh)$, where m is the number of edges and h is the number of cliques.

We have considered those approaches and methodologies to compare with other our algorithm as they are used to find overlapping communities.

2.8 Other Approaches

Approximate resistance networks although a similar concept founded by Newman but Wu et al. [42] presented an extension of this resistance approach method to reduce the time complexity of it. The algorithm identifies communities based on the properties of resistor networks. It is a method for partitioning graphs in two sections, similar to that of spectral bisection, where it partitions in an arbitrary number of communities, obtained by repeated iterations. The graph is transformed into a resistor network where each edge has an unit resistance. A unit potential difference is set between two randomly chosen vertices named as source and sink.

The underlying idea is that if there is a visible gap between voltage values for vertices at the borders between the clusters, then it makes two separate communities. The voltages are calculated by solving Kirchhoff's equations, an exact solution would be too time consuming, but it is possible to find a reasonably good approximation in a linear time for a sparse graph with a distinct community structure, so the more time consuming

part of the algorithm is the sorting of the voltage values, which takes time O(n log n) [42]. Any possible vertex pair can be chosen to set the initial potential difference, so the procedure should be repeated for all possible vertex pairs. The authors showed that this is not essential, and that a certain number of sampling pairs is sufficient to get desired results, so the algorithm scales as O(n log n) and is very fast.

An interesting feature of the method is that it can quickly find the natural community of any vertex, without determining the complete partition of the graph. For that, one uses the vertex as source voltage and places the sink at an arbitrary vertex. The same feature is present in an older algorithm by Flake et al. [43], where one uses max-flow instead of current flow.

The algorithm proposed by Orponen and Schaeler [44] is based on the same principle, but it does not need the specification of target sources as it is based on diffusion in an unbounded medium. The limitation of such methods is the fact that one has to input the number of clusters, which the user is not aware of before. The accuracy of this method is dependent on how many times the iterative step is repeated and we are not sure of the maximum limit the step should be followed. It is also dependent on having a good idea of the sizes of communities, which make it difficult to use it in large networks. However, this is one of the few methods that is able to identify the community around one node in linear time.

Many researchers found a general network was similar to an electrical network one after another. So a general network expressed by a connected graph can be transformed into an electrical network. In general, let an undirected weighted graph G = (V, E, W) be a network, where V is a set of n vertices, E is a set of m edges between vertices, and W is a set of weight values on the edges and usually the edges are weighted by 1. According to the knowledge of electrical network, a complex network can be simplified as an electrical network. With the above conceived notion, we observe that the properties of

ERF is a RSM. We further use it to detect communities in both synthetic and real-world networks.

The motivations of the algorithms that we discussed are mainly some concrete problems [14, 45]. Also, the complexities of the algorithms are optimised for dynamic network community detection [21, 46, 47].

2.9 Properties of the social networks

First, to be able to detect communities in a static and a dynamic network, it is essential to analyze the properties of these networks or graphs. In this section various properties observed in social networks are discussed that are commonly used to characterize them. An overview of social networks described in Section 2.9.1. In this section, also the sizes of social networks in Section 2.9.2 and degree distribution in Section 2.9.3 will be introduced, because they influence the way that the network can be analyzed. Finally community structures are described in Section 2.9.4.

2.9.1 Procedures to obtain social networks

A social network is a social structure made up of individuals (or organizations) called nodes, which are tied (connected) by one or more types of interdependencies, such as friendship, kinship, common interest, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige with links. Social networks could be defined by their way of gathering, traditional social networks and online social networks (OSN).

Social networks are gathered manually by social researchers which follow a group of people over a period of time to investigate their social interaction. This type of network will be called traditional social networks because of the way they are obtained.

Famous examples of traditional social networks are the Zachary's karate club and the football network of Girvan et al. [5]. In the Karate club network, the nodes represent members of a karate club and the links are the social interactions between them. In the football network, the nodes represent football teams and a link exists if they played a match against each other. More details about the detection of the communities in these network are discussed in the later sections.

2.9.2 Sizes

The size of a network which is commonly defined by the number of nodes, determines the way a network can be obtained and analyzed. While small social networks can be gathered and interpreted by hand, for larger networks that becomes infeasible and automated methods have to be used. For large networks (Peer to Peer [48]), even automated methods to analyze networks can be infeasible due to their runtime. Therefore, the size is an important property of a network. The number of nodes in a social network heavily depends on the type. Traditional social networks (Karate Club [49], Dolphin [1], High-School [50]) are gathered through field studies by researchers and therefore have a limited number up to hundreds of nodes. Depending upon the size and density of the network, we see the variation of our algorithm's performance. One of the largest networks that is fully available for researchers is a Twitter dataset from 2009 which contains 50 million users and 1.5 billion links and was made available by Kwak et al. [51].

2.9.3 Degree and Distribution

Besides the size of the network, another important property of social networks is how the links are distributed among the nodes, in other words how the distribution of friends is among the users of a social networks. This property is not only relevant for understanding the relationships between entities, it also influences the order in which nodes will be

obtained in certain community detection methods. The degree is defined as the number of adjacent neighbors of a node. The degree distribution *P(k)* is the probability for a fraction of nodes in a network having degree k. A frequently observed property in many online social networks is their power-law degree distribution. A power-law degree distribution is described by the following relation:$P(k) \sim k^{-\gamma}$.

2.9.4 Community Structure

Detection of communities mainly depends upon the community structure which is the main focus of this thesis and will be addressed in the later section. A general definition of community structure introduced by Girvan and Newman [5] is used, namely "The division of network nodes into groups within which the network connections are dense, but between which are sparser". An example of a network with community structure is depicted in Fig. 2.2. Nodes in a community should share more connections with each other than with nodes in other communities.

In the example, nodes within a community are completely connected meaning that all possible links within the community exists, while there are few links between nodes of different communities.

2.10 General Definitions about Community and Community Structure

Community: A community (or cluster or group) can be defined in several ways, and there is no universally accepted definition. However it can be intuitively understood as a set of nodes that are densely connected to each other, and relatively sparsely connected to the rest of the graph.

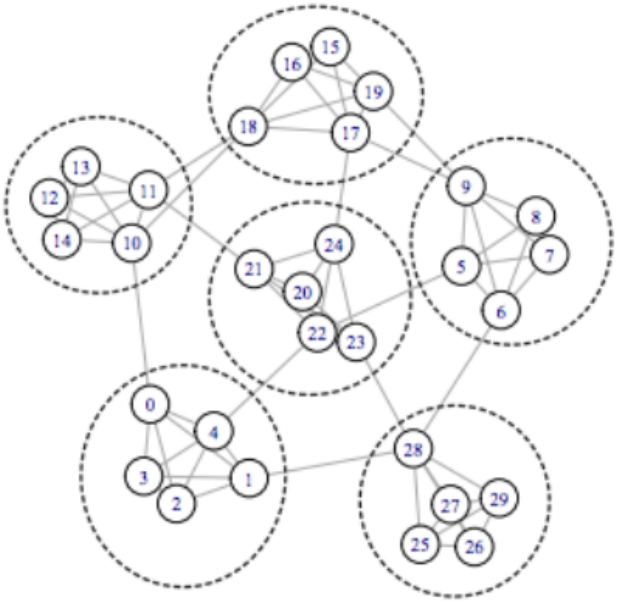


Figure 2.2: Example of a network with 6 communities, represented by the dashed circles

Community Structure: The community structure of a graph is the set of communities in a graph, it can be represented as a partition-where each node belongs to only one community. The discussion of overlapping communities, where a cover of the graph is desired is discussed in our research.

Community detection algorithm looks to identify a partition $P = C1...Ck$ of the graph such that the nodes of each community are densely connected to each other, and relatively sparsely connected to the rest of the graph.

However, researchers hardly focus on the general definition of communities and the general community detection algorithms. The lack of the understanding of the relations among various algorithms might pose some obstacles to the optimisation of them so that it is hard to use one algorithm to perfect the other.

Some comparisons have been made among the different definitions and the corresponding algorithms [52]. Besides, some researcher believes that the definition often depends on the specific system at hand and/or application one has in mind [13].

2.11 Summary

In this chapter we reviewed the approaches followed by various community detection algorithms. We discussed on the properties of the networks and the limitations of the detection algorithms. We also discussed on the general definitions of community, structure and the community detection algorithms.

Chapter 3

Relation Strength Measurement

Our real-world consists of elements and relations among them. We usually call the entity made by entities and relations among them as network. These networks are not random, as they display big inhomogeneities, revealing a high level of order and formation. In particular, several elements have much stronger inner relations comparing to the ones with other nodes in the network and thus considered as a "community". Thus the strength of the relation (RSM) among those elements/nodes needs to be defined for detecting communities in the networks.

3.1 Concrete problem reduction

Most people believe that, a community is some set of objects that the relation strengths among them are strong. However, there are lots of arguments on the definitions of relations[1] and the ways to measure them. Generally, we derive the relations in two ways.

One is based on the *material* transmission. The materials here can represent both material substance (like goods) or just information (like data). Intuitively, the objects that can easily communicate with each other should have strong relations among them

[1]The relations to derive communities rather than the ones in mathematics

and then, those objects can somehow be considered as a community.

The other one is based on the similarity (for example, people buy the similar kind of books might be considered as a community). And in usual cases, we believe the objects in the same community should share some other similar properties. Then we can do some reasonable predictions on the community level (For example, Amazon uses this trick to recommend new books).

Although both two community derivations come from different backgrounds, we show that both of them can be reduced to the same graph model.

3.1.1 Transmission Network

3.1.1.1 Transmission Relation Characteristics

In order to make the problem easier to discuss, only one material will be considered. Moreover, we need the following definitions and assumptions.

Assumption 1. *For any material, there is a minimal unit can be transferred. And we call this minimal unit a point.*

Definition 1 (Node)**.** *The object that receives and sends points is node.*

Definition 2 (Medium)**.** *The object that propagates points is medium.*

Assumption 2. *The transmission relations are constructed in nodes, media and points, which will also determine the properties of the relations.*

Because of assumption 2, the entity we need to consider consists of nodes, points and media. And we name it *transmission network*.

Two most important quantities in a transmission network are *the number of points transferred* and *the time used in this process*. The quantity relating them is *speed*.

Definition 3 (Speed)**.** *Speed is the number of points transferred in an unit time length.*

The transmission capability can be described by the speed function of time. Currently, only the pair of nodes connected by media directly will be considered. By assumption 2, this function's behaviour depends on the properties of the network's nodes, media and points. This means a specific analytic expression of the speed function cannot be given unless all the properties have been designated. However, some common characteristics can be expected e.g. a pair of nodes (u, v) which are connected by media directly as shown in Fig. 3.1.

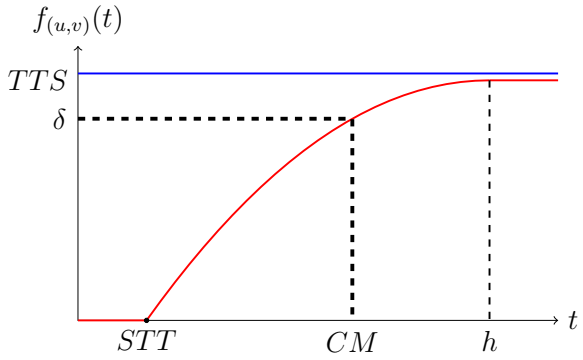


Figure 3.1: Transmission Capability Graph

There is an interval at the beginning that the function's value is 0, which is due to the consideration of the time for node receiving and sending points as well as the one for points propagating in media. We call the first moment that the function value becomes nonzero the *shortest transmission time* (STT)[2]. Besides, there is always a limit of speed for some certain point and medium. Thus, there should be some upper bound. And we call the least upper bound *theoretically transmission speed* (TTS).

However, in the real-world, TTS may never be reached, which leads to the meaninglessness of it. So instead, we can choose some proper *threshold* $\delta \in \mathbb{R}^*$ which is the

[2]Theoretically, STT = sending time + transmission time + receiving time.

lowest speed we can accept. Moreover, we call the time to reach this threshold the *critical moment* (CM).

Remark 1. *Suppose there are two towns A and B near a river. A is upstream of B. Consider the goods transportation on the river. Because of the water stream, $f_{(A,B)}(t) \geq f_{(B,A)}(t)$. Thus, in general cases, $f_{(u,v)}(t) \neq f_{(v,u)}(t)$. Typically 'f' is considered to be the transmission function.*

3.1.1.2 Special Relation Strength Measurement For Transmission Network

The transmission speed functions of time can describe almost all the important properties of relations in a transmission network. So, the special relation strength measurement (SRSM) can be derived from this function according to the problem confronting us. Here are several examples.

1. The shorter STT is, the stronger the corresponding relation is.
2. The shorter CM is, the stronger the corresponding relation is.
3. The shorter the time to transfer certain number of points is, the stronger the corresponding relation is.

In fact, a reasonable SRSM cannot be constructed until a concrete problem is given. However, several key properties should be shared by all SRSMs.

Basically, in a transmission network, relations should not cancel out each other. So the SRSM is non-negative. Besides, the strongest relation in a transmission network should be the one that the node relates to itself because there is nothing to be transferred and the speed can be considered as infinity forever. Intuitively, all SRSMs should have the same value in this case. So the most special element 0 in $\mathbb{R}^*$ is used. In the other direction, the relation strength will get weaker if the function value increases. Therefore, we have the following definition.

Definition 4 (SRSM)**.** *Suppose there is some network. Let N be the set of nodes in it and $P \subseteq N \times N$ be the set of pairs of nodes that are connected by media directly. Moreover, let $s : P \to \mathbb{R}^*$, which satisfies the following properties for all $(u, v) \in P$:*

1. $s(u, v) \geq 0$.
2. $s(u, v) = 0$ *if and only if* $u = v$.

Then we say s is a special relation strength measurement (SRSM).

Remark 2. *Because transmission function might be different due to the order in the pair of nodes, so is SRSM.*

With the help of SRSM, the transmission network can be reduced to a graph model. In the graph, all the nodes become the vertices. Two nodes are connected by a weighted edge if they are connected by media directly. Moreover, the edge's weight is assigned by some specific SRSM constructed according to the problem considered. In general cases, the graph should be directed. If $s(u, v) = s(v, u)$ for all applicable pairs of nodes (u, v) in a network, the graph can also be considered as an undirected one.

The graph model in Section 3.1.1.3, will be used to define more general relation strength measurement. Besides, we would use the terms *node* and *vertex* interchangeably. Similarly the terms *relation* and *edge* are interpreted the same.

3.1.1.3 Relation Strength Measurement for transmission network

In this section, we will discuss on the graph model based on which various community detection algorithms are derived. So some notations concerning graphs need to be introduced at first.

The classical notations will be used. Let $G = (V_G, E_G)$ be a weighted graph. V_G and E_G are the sets of vertices and edges of Graph G respectively. For any edge $e \in E_G$, the weight of e is denoted by $|e|$.

In the previous section, SRSM measures the direct relations between any pair of nodes. That is, the retransmission function of nodes is not taken into account. Neither is the parallel transmission on various paths. However, in the real-world, the retransmission and parallel transmission of points happen over and over again[3], which reveals the necessity to derive a more general relation strength measurement function. We name this new measurement **relation strength measurement for transmission network (RSMFTN)**.

For the same reason in derivation of SRSM, an analytic expression of RSMFTN cannot be given unless a concrete problem is designated. But several key properties must be held for a reasonable RSMFTN.

First of all, all the relations between any pair of nodes should be measurable. So the domain of RSMFTN should be $V_G \times V_G$. The properties of SRSM should be inherited. Then RSMFTN is also non-negative and $RSMFTN(u, v) = 0$ if and only if $u = v$. Besides, there cannot be any relation between two nodes belonging to two disconnected components respectively. Comparing with the coincidence of two nodes, this is the other extreme. So it is reasonable to define the value of $RSMFTN$ to be infinity(∞), which is an element greater than any real number. In general, the relation of some vertices u and v will get stronger if the function value $RSMFTN(u, v) \to 0$.

Consider a linear graph Fig. 3.2(a). All the points sent by u and received by v will be retransmitted by w. So the difficulty to transfer points from u to v is greater or equal to the "sum" of the ones from u to w and from w to v. Notice that the difficulty for points to be received and sent by some node has been taken into account by SRSM and thus, is contained by the edge's weight. Hence, the equality should hold. In other words, $RSMFTN(u, v) = RSMFTN(u, w) + RSMFTN(w, v)$ if w is an intermediate connecting node on the path from u to v. To add on, the transmission difficulty does not

[3]Like express service and data transmission on the internet

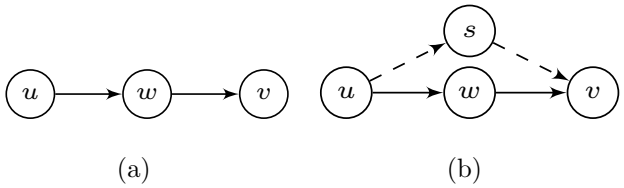


Figure 3.2: Linear Transmission Graph

increase if we add some other retransmission node s (See Fig.3.2(b)). If we generalise this idea, we have $RSMFTN(u, v) \leq RSMFTN(u, w) + RSMFTN(w, v)$.

If we sum all ideas up, we can define the RSMFTN as follows:

Definition 5 (Relation strength measurement for transmission network)**.** *Suppose G is some directed graph and $g : V_G \times V_G \to \mathbb{R} \cup \{\infty\}$ is an abstract function, which satisfies the following properties:*
For all vertices u, v and w in V_G

1. $g(u, v) \geq 0$. *(non-negativity)*
2. $g(u, v) = 0$ *if and only if u and v coincide. (coincidence axiom)*
3. $g(u, v) = \infty$ *if and only if there is no path between u and v.*
4. $g(u, v) \leq g(u, w) + g(w, v)$. *Moreover, the equality holds if the two components that contains u and v are connected by the intermediate connecting node w.*
5. *Suppose G' is a graph which is same as G except that the edges' weights in G' are all α times greater than the ones in G. Then for the corresponding vertices u' and v' in G', $g(u', v') = \alpha g(u, v)$*

Besides, if G is undirected,

6. $g(u, v) = g(v, u)$ *(Symmetry)*

Then we say g is a relation strength measurement for transmission network (RSMFTN).

Remark 3. *Actually, many measurements derived by other researchers are RSMFTNs. A well-known one should be shortest distance function(SDF), which evaluates the shortest distance of some pair of nodes in a graph.*

Proof. Suppose u, v and w are arbitrary vertices in some directed graph G. g is SDF. We prove the proposition when the graph G is weighted. The unweighted graph is considered as the weighted graph whose edges' weights are 1.

Since the $g(u, v)$ returns the sum of weights of the shortest path between u and v, $g(u, v) \geq 0$. The Property 1 holds. The Property 2 and 3 hold by the definition of SDF.

For Property 4, assume $g(u, v) > g(u, w) + g(w, v)$. Consider the path p consisting of the shortest path from u to w and the one from w to v. It is easy to see that the length of the path $l(p)$ is $g(u, w) + g(w, v)$, which is shorter than $g(u, v)$. So $g(u, v)$ cannot be the length of the shortest path. We get a contradiction. Moreover, if there exists a intermediate connecting node w connecting the components containing u and v respectively, the shortest path between u and v can be split into the one from u to w and the one from w to v. So $g(u, v) = g(u, w) + g(w, v)$. Therefore, the Property 4 holds.

For Property 5, suppose path p is some shortest path from u to v in graph G. Let p' denote the corresponding path in G. Since all the edges' weights in G' are α times the ones in G, so is the length of path p'. That is, $l(p') = \alpha l(p)$. Since it is obvious that path p' connects u' and v' in G', then we have $g(u', v') \leq l(p')$. In other words, $g(u', v') \leq \alpha g(u, v)$. Similarly, consider the reverse transformation from G' to G. That is, all the weights of edges in G' is $\frac{1}{\alpha}$ times in the ones in G. So we have $\frac{1}{\alpha} f(u', v') \geq g(u, v)$, which is equivalent to $g(u', v') \geq \alpha g(u, v)$. Combining with $g(u', v') \leq \alpha g(u, v)$, we conclude that $g(u', v') = \alpha g(u, v)$. So Property 5 holds.

Suppose G is an undirected graph, then by the commutativity and associativity of the addition operator, $g(u, v) = g(v, u)$. In other words, the order to add the weights of the edges compounding the shortest path does not change the final result.

To sum up, SDF is an RSMFTN.

□

3.1.2 Similarity Network

3.1.2.1 Similarity Function

In order to make the problem easier to discuss, we need to give some fundamental definitions at first. We name the objects needed to measure the similarity among the nodes. Each nodes should have some various properties. Moreover, there should be some possible options for each property (for example, red, blue, yellow are possible options for property color), and we name these options cases.

The similarity network concerns the nodes' properties similarity. We assume that, for a certain problem, the set of properties is fixed and for each property, there exists a similarity function that maps some pair of cases to real numbers. Intuitively, the function value makes no sense if it is negative. Thus, we define similarity function as a nonnegative one. Besides, just following the similar convention when defining SRSM in Section 3.1.1.2, the function value should increase if the similarity decreases. Moreover, for some objects A and B, if A is similar to B, then B is also similar to A. Then we make this preliminary idea formal,

Definition 6. *Assume P is some set of possible cases for some property of a node, then there exists a* ***similarity function*** $sim_P : P \times P \to \mathbb{R}$ *such that,*

1. $sim(p_1, p_2) \geq 0$ *(non-negativity)*
2. $sim(p_1, p_2) = 0$ *if and only if* $p_1 = p_2$ *(coincidence axiom)*
3. $sim(p_1, p_2) = Sim(p_2, p_1)$ *(symmetry)*

Since for a certain problem, there should be certain couple of properties $P_1, P_2 \cdots$ needed to consider, there also exists the corresponding similarity functions $sim_{P_1}, sim_{P_2}, \cdots$. For convenience, we can write them in matrix form. Namely, $[P_1\ P_2 \cdots]$ and $[sim_{P_1}\ sim_{P_2} \cdots]$.

Although we have defined the similarity function for properties, we cannot compute similarity between objects because objects may have many various properties. Moreover,

for a certain scenario, the importance of different properties may not coincide. Hence, we should introduce some new function to handle this problem. In the tradition, we usually name the function that manipulates the importance of different factors *weight function*. In our paper, we also follow this convention. So we have the following definition:

Definition 7 (Weight function)**.** *Suppose N is the set of nodes needed to discuss in some problem. P_1 P_2 $\cdots$ are the properties needed to consider. Besides, $[sim_{P_1}\ sim_{P_2} \cdots]$ is a list of the corresponding similarity functions. Then $w : [sim_{P_1}\ sim_{P_2} \cdots] \mapsto f$ is a weight function where $f : N \times N \to \mathbb{R}$ is a non-negative function.*

Remark 4. *The choice of the weight function should depend on the problem confronting us. The most common weight function should be a list of weights. In more details, suppose $[sim_{P_1}\ sim_{P_2} \cdots sim_{P_n}]$ is a list of the corresponding similarity functions. Then $w = [\alpha_1\ \alpha_2 \cdots\ \alpha_n]^T$ could be some weight function. So f here becomes $[\alpha_1\ \alpha_2 \cdots\ \alpha_n]^T \cdot [sim_{P_1}\ sim_{P_2} \cdots sim_{P_n}] = \sum_{i=1}^{n} \alpha_i \cdot sim_{P_i}$, which is a non-negative function.*

3.1.2.2 Relation Strength Measurement For Similarity Network

The weight function can generate a function f to measure the similarity of a pair of nodes. However, the weight function here has no guarantee that f will always give the measurement following our intuition. In particular, we hope f should have the following properties:

Suppose N is the set of nodes needed to discuss. P is the set of properties needed to consider. Then for any nodes u, v and w in N, we have,

1. $f(u, v) \geq 0$
2. $f(u, v) = 0$ if and only if u and v have the exactly same cases for all properties in P.
3. $f(u, v) \leq f(u, w) + f(w, v)$

4. $f(u, v) = f(v, u)$

The first two properties are inherited from the similarity function. Since the similarity relation should be symmetric (i.e. if A is similar to B, then B is also similar to A), then we have Property 4. Besides, Property 3 shows that the direct measurement of any pair of nodes is at least not greater than the sum of the ones with an intermediate point. Since this function is defined for similarity measurement, we name it the **relation strength measurement for similarity network (RSMFSN)**.

Remark 5. *In other words, f should be a distance function. In fact, the example we give in Remark 4 is an RSMFSN.*

3.1.3 Connections between similarity network and transmission network

In many cases, there are very strong relations between similarity networks and transmission networks. A typical example is the pathogen infection among some species. If we consider the DNA similarity among organisms. It is easier for some certain pathogen to infect organisms that have similar DNAs. Or in other words, the easiness of pathogen transmission will increase if the DNA similarity among the organisms increases. Therefore, the relative relation strength among the organisms should be similar no matter which relation type we consider here.

Since both RSMFTN and RSMFSN are used to measure the relations among the nodes in networks and the propositions in rest paper will not be affected by the difference of their definitions, we call both two relation measurements **relation strength measurement (RSM)** in the later discussion.

3.2 Definition of Community

After defining RSM, the definition of communities can be derived. Before formally defining community, an important problem needs to be discussed. That is the community relation's transitivity. In other words, if A and B are in one community and so are B and C, can we also say A and C are in one community? In general, this implication is not true. A typical counterexample is "Your friends' friends may not be your friends". So all relations strength between any pair of nodes should be considered when we define a community. Moreover, a relation strength threshold (community threshold parameter) is needed to be designated since only those nodes having strong enough relations can be considered as a community.

Definition 8 (Community)**.** *Suppose $W^G \subseteq V_G$ for some directed graph G, $\epsilon \in \mathbb{R}^*$, and g is some RSM. Then W^G is a community with respect to RSM g and constant ϵ if and only if $\forall (u, v) \in W^G \times W^G$. $g(u, v) \leq \epsilon$. Moreover, we say ϵ is the community parameter(CP) of W^G with respect to g. If there is no ambiguity of the choice of RSM and CP, we will briefly say W_G is a community.*

Since CP gives a threshold of the relations strength, no matter which pair of nodes we choose in a community, the relation strength of the pair cannot be weaker than the ones the CP represents. So for those problems considering the worst cases, the CP can be designated according to some CM with some certain threshold (in Section 3.1.1.1). Then the inner structure of the community can be ignored in the later discussion since the poorest performance of the community satisfies the requirement. In other words, a community can be considered as a relatively independent entity, and the CP is some global property of it.

Remark 6. *Notice that the definition of the community is based on the set of vertices instead of the subgraph used in many other papers. Besides, it is worthy to emphasize*

that the choice of community will usually consider the whole graph's topology rather than the local one (this shows that the community is some higher level structure based on the original graph). Since the results might be different for various choices of graph topology, the superscripts are used to make the description clear. (For example, W^G means W is a vertices set and graph G is the working topology).

3.3 Some propositions and Detection Algorithm for absolute communities

Based on the definition of RSM, an adjoint complete graph can be derived for recording all the relation strengths. More accurately, the weights of edges in the adjoint graph will be assigned by corresponding RSM values.

Definition 9 (Adjoint complete digraph)**.** *Suppose G is some directed graph and g is some RSM. Let $E = V_G \times V_G$ be a new set of edges where the weights are assigned by g. Then the adjoint complete digraph $adj(G, g) := \{V_G, E\}$.*

The definition of communities uses CP to give a threshold of the relation strength. That means, if the relation strength is not strong enough, then this relation will not be considered during the community detection. Moreover, for any pair of nodes, the definition of communities requires the enough strengths of the relations in both two directions. Hence, we can remove the relations unsatisfying the requirement to simplify our graph without changing the result of community detection. Based on this idea, we have the following transformation.

Definition 10 (Refinement transformation)**.** *Suppose G is a directed weighted graph, $D_G \subseteq E_G$, and $\epsilon \in \mathbb{R}^*$ is some CP. The refinement transformation is defined like this.*

$$R_\epsilon(D_G) := \{(u, v) \in D_G : |(u, v)| \leq \epsilon \wedge |(v, u)| \leq \epsilon\}$$

Besides, all the edges' weights are set to 1 after applying the transformation.

Remark 7. *For convenience, the relations* (u, v) *and* (v, u) *are together denoted by* $u \leftrightarrow v$*. In this case the weight is not applicable.*

The definition of refinement transformation shows that if some edges (u, v) is in $R_\epsilon(D_G)$, then so is (v, u). Moreover, the edges' weights become unnecessary since they all equal 1. Therefore, there is no need for us to consider the edges' directions and weights in $R_\epsilon(D_G)$ anymore. So in the later discuss, $R_\epsilon(D_G)$ will be thought of a set of undirected unweighted edges. Moreover, if $(u, v) \in R_\epsilon(D_G)$, then we say the relation between u and v is reserved. Or briefly, $u \leftrightarrow v$ is reserved.

In fact, the refinement transformation is a higher order function that applies a Boolean function to each relation in the set of edges. The Boolean function here judges whether the given relation is strong enough to be considered in the community detection. So, for a certain refinement transformation, the reservation of the relation depends on the strength of the relation itself rather than the topology in which the relation is.

Lemma 1. *Suppose* G *is some directed graph. If* $E \subseteq F \subseteq E_G$*, then* $R_\epsilon(E) \subseteq R_\epsilon(F)$*.*

Proof. Pick $(u, v) \in R_\epsilon(E)$ arbitrary. So $u \leftrightarrow v$ is in E and reserved after applying the refinement function. Since $E \subseteq F$, then $u \leftrightarrow v \in F$. So $(u, v) \in R_\epsilon(F)$. □

Then the adjoint complete digraph can derive a simplified undirected unweighted graph whose edges represent the two-direction relations strong enough to construct communities.

Definition 11 (Effective edge graph)**.** *Suppose* G *is some directed graph.* g *is some RSM.* ϵ *is some CP. Then the effective edge graph is* $(V_G, R_\epsilon(E_{adj(G,g)}))$ *and denoted by* $eeg_{(g,\epsilon)}(G)$*. Moreover, suppose the vertices set* A^G *is a subset of* V_G*. The full subgraph of* $eeg_{(g,\epsilon)}(G)$ *over* A^G *is denoted by* $eeg_{(g,\epsilon)}(G)[A^G]$*.*

Before the detection of communities, the characteristic of them in *eeg* needed to be figured out. Theorem 1 discusses this problem.

Lemma 2. *Let g be some RSM and $\epsilon \in \mathbb{R}^*$ be some CP, and suppose G is some directed weighted graph. $W^G \subseteq V_G$. Then the vertices set W^G is a community if and only if $\forall u, v \in W^G . u \leftrightarrow v$ is reserved after applying the refinement transformation.*

Proof. The refinement transformation will remove all the relations that cannot be used in a community structure. In other words, if all relations are reserved after applying refinement transformation, the relation in any pair of nodes is strong enough. This is exactly what the definition of community requires. So, W^G is a community. On the other hand, if W^G is a community, the relation(in both directions) in any pair of nodes should be strong enough. Thus, all of them will be reserved after applying the refinement transformation. □

Lemma 3. *Any full subgraph of a complete graph is again complete.*

Proof. Suppose S is a full subgraph of C_n for some n. Then for arbitrary vertices u and v in S, edge $(u, v) \in E_{C_n}$. So by the definition of full subgraph, $(u, v) \in E_S$. That is, there is an edge in arbitrary pair of nodes in S. So S is a complete graph. □

Theorem 1. *Suppose G is some directed graph. g is some RSM. ϵ is some CP. $A^G \subseteq V_G$ is a community if and only if $eeg_{(g,\epsilon)}(G)[A^G]$ is complete.*

Proof.
($\Rightarrow$) Suppose A^G is a community. Pick vertices u and v in A^G arbitrary. Since A^G is a community, then all the relations will be reserved after applying the refinement transformation. Moreover, since $adj(G, g)[A^G]$ is complete, then $u \leftrightarrow v \in E_{adj(G,g)[A^G]}$. Thus, $(u, v) \in R_\epsilon(E_{adj(G,g)[A^G]})$. Since we pick u and v arbitrary in A^G, there isa n edge in arbitrary pair of nodes in $eeg_{(g,\epsilon)}(G)[A^G]$. Hence, $eeg_{(g,\epsilon)}(G)[A^G]$ is complete.
($\Leftarrow$) Suppose $eeg_{(g,\epsilon)}(G)[A^G]$ is complete. Then $\forall (u, v) \in A^G \times A^G$, $(u, v) \in R_\epsilon(E_{adj(G,g)[A^G]})$. Since $eeg(G, g)[A^G]$ is complete, all the relations in $E_{adj(G,g)[A^G]}$ are reserved after applying the refinement function. Therefore, A^G is a community. □

It is easy to find that all single nodes can be considered as a community because they relate to themselves only, and RSM is always 0. However, this kind of result does not quite follow our intuition since the community should be some set of nodes. The definition of the maximal community will tackle this inconsistency. For a better understanding of the definition, a very important theorem needs to be introduced at first.

Theorem 2. *Suppose G is a directed graph. ϵ is a CP. g is a RSM and $A^G \subseteq B^G \subseteq V_G$. If B^G is a community, then so is A^G*

Proof. Since B^G is a community, then by Theorem 1, $eeg_{(g,\epsilon)}(G)[B^G]$ is complete. Besides, since $A^G \subseteq B^G$, then by lemma 1, $eeg_{(g,\epsilon)}(G)[A^G]$ is a subgraph of $eeg_{(g,\epsilon)}(G)[B^G]$.

Moreover, pick $u, v \in A^G$ arbitrary. Then $u, v \in B^G$ as well. Notice that $eeg_{(g,\epsilon)}(G)[B^G]$ is complete. So the relation $u \leftrightarrow v \in E_{adj(G,g)[B^G]}$ is reserved, which implies $(u, v) \in E_{eeg_{(g,\epsilon)}(G)[A^G]}$. Thus, $eeg_{(g,\epsilon)}(G)[A^G]$ is complete. So A^G is a community as well. □

Theorem 2 shows that, if B_G can be considered as a community with respect to some RSM and CP, then all the subsets of B_G can be considered as a community. This observation leads to the definition of maximal community.

Definition 12 (Maximal community)**.** *Suppose G is a directed graph. RSM and CP are given. Moreover, A^G is a subset of V_G. Then A^G is a maximal community if and only if*

1. *A^G is a community, and*
2. *There is no $B^G \subseteq V_G$ such that B^G is a community and $A^G \subsetneq B^G$.*

The following theorem shows the equivalence between detecting all the maximal communities in some graph G and finding all the maximal cliques in the corresponding $eef_{(g,\epsilon)}(G)$.

Theorem 3. *Suppose G is a directed graph, ϵ is some CP, g is a RSM, and A^G is a subset of V_G. Then A^G is a maximal community if and only if $eef_{(g,\epsilon)}(G)[A^G]$ is a maximal clique in graph $eef_{(g,\epsilon)}(G)$.*

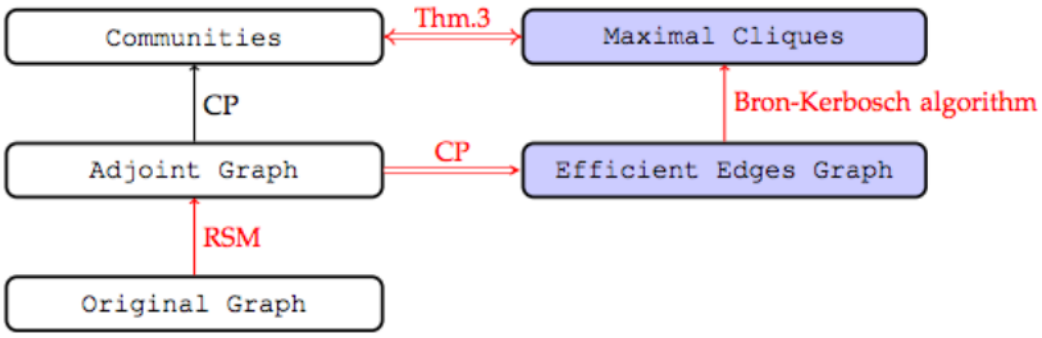


Figure 3.3: Whole Algorithm Structure

Proof.

($\Rightarrow$) Suppose A^G is a maximal community. Then since A_G is a community, then $eef_{(g,\epsilon)}(G)[A^G]$ is complete. So it is a clique. Assume $eef_{(g,\epsilon)}(G)[A^G]$ is not maximal. Then there exists some vertices set B^G such that $eef_{(g,\epsilon)}(G)[A^G] \subsetneq eef_{(g,\epsilon)}(G)[B^G]$ and $eef_{(g,\epsilon)}(G)[B^G]$ is complete. So $A^G \subseteq B^G$. Moreover, since both two graphs are complete, the equality cannot hold. Otherwise, $eef_{(g,\epsilon)}(G)[A^G] = eef_{(g,\epsilon)}(G)[B^G]$. So we have $A^G \subsetneq B^G$. Besides, since $eef_{(g,\epsilon)}(G)[B^G]$ is complete, B^G is a community. Hence, A^G cannot be a maximal community, which is a contradiction.

($\Leftarrow$) Suppose $eef_{(g,\epsilon)}(G)[A^G]$ is a maximal clique. Since $eef_{(g,\epsilon)}(G)[A^G]$ is complete, then A^G is a community. Assume A^G is not maximal, then there exists some community B^G such that $A^G \subsetneq B^G$. Then $eef_{(g,\epsilon)}(G)[B^G]$ is complete. Moreover, we have $eef_{(g,\epsilon)}(G)[A^G] \subsetneq eef_{(g,\epsilon)}(G)[A^G]$ by lemma 1. Therefore, $eef_{(g,\epsilon)}(G)[A^G]$ cannot be maximal, which is a contradiction. □

Remark 8. *Bron-Kerbosch Algorithm [53] is a well-known algorithm to find maximal cliques. Then hence, by Theorem 3, we can also use this algorithm to find the maximal communities as well.*

Fig. 3.3 shows the relationships among the important propositions and transformations introduced. The path in red reveals an algorithm to find all the maximal commu-

nities in a graph. In more detail, suppose G is some graph. g is some RSM. ϵ is some CP. Moreover, all the vertices in G have been indexed from 1 to $|V_G|$. Then we have the following algorithm;

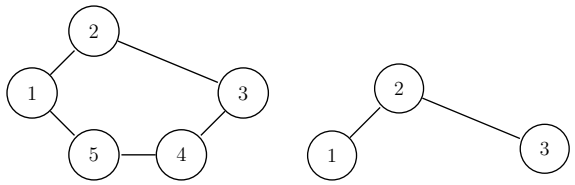
Algorithm 3.1 Find all maximal communities in a graph

```
1: procedure findMaximal Communities
2:     for (i, j) ∈ V_G × V_G where i ≤ j do
3:         if g(i, j) ≤ ε & g(j, i) ≤ ε then
4:             EEG[i, j] = 1
5:         else
6:             EEG[i, j] = 0
7:         end if
8:     end for
9:     SetOfCommunities ← Apply Bron-Kerbosch algorithm on matrix EEG
10: end procedure
```

3.4 Demonstration

3.4.1 Klein and Randic's Model

The definition of communities indicates that some certain RSM is required. We have shown that SDF is RSMFTN in Remark 3, so that SDF is RSM. Although many community detections work on SDF, it may not always give reasonable result. Intuitively, the relation of a pair nodes will get stronger if there are more paths between them. However, SDF does not consider this case as shown in Fig. 3.4. More specifically, in SDF view, the relation will not get stronger unless a path shorter than the previous shortest path is added.



(a) (b)

Figure 3.4: Both in (a) and (b), the shortest distance between Node1 and Node3 is 2. So in SDF view, the relation strengths between Node1 and Node3 in these two cases are same. However, there is one more path between Node1 and Node3 in (a). So, intuitively, the relation between Node1 and Node3 in (a) should be stronger than the one in (b)

In order to avoid this problem, we try to use the Klein and Randic's effective resistance function (ERF) [54] to measure the relation strength instead of SDF.

3.4.2 Klein and Randic's effective resistance

Suppose G is an indirect connected graph. Then G can be considered as an electrical network that all the edges are resistances with the corresponding weight values (if G is an unweighted graph, then the resistances of all edges are 1).

Let u and v be two vertices in the graph. Then the effective resistance of these two vertices can be defined like this:

Definition 13. *Let the voltage of u be U and the one of v be 0. We can measure the current I from u to v. Then the efficient resistance $R(u, v)$ between u and v is $\frac{U}{I}$. Briefly, $R(u, v) = \frac{U}{I}$.*

3.4.2.1 Algorithm to get Efficient Resistance Distance

Klein and Randic [54] also provided an algorithm to compute the resistance distance for a connected indirect weighted graph.

Suppose graph G is connected. Let A be the adjacent matrix and D be the diagonal

degree matrix of graph G. It is worthy to note that, in a weighted indirected graph, the degree of a vertex is the sum of the weights of all its adjacent edges. Then the Laplacian matrix L can be computed using formula $L = D - A$. Let $L^{\dagger}$ be the generalized inverse [55] of L. Then the efficient resistance distance $R_{i,j}$ of any pair of vertices (i, j) in graph G can be computed by

$$R_{i,j} = L^{\dagger}_{i,i} + L^{\dagger}_{j,j} - 2L^{\dagger}_{i,j}$$

And we usually call the corresponding matrix R **resistance matrix**.

3.4.2.2 ERF is RSMFTN

Since the definition of community is based on RSM, we have to prove ERF is an RSM first. Essentially, in this case, the relations among the nodes are derived from the electron flow in the wires among the vertices. So we need to consider the criteria of RSMFSN.

Lemma 4. *Resistance is distance. That is, the resistance satisfies the following properties:*

1. $R_{a,b} \geq 0$
2. $R_{a,b} = 0 \Leftrightarrow a = b$
3. $R_{a,b} = R_{b,a}$
4. $R_{a,c} + R_{c,b} \geq R_{a,b}$

Lemma 5. *Let x be a cut-point of a connected graph, and let a and b be points occurring in different components which arise upon deletion of x. Then,*

$$R_{a,b} = R_{a,x} + R_{x,b}$$

Remark 9. *The proofs of Lemma 4 and Lemma 5 have been given by Klein and Randic [54].*

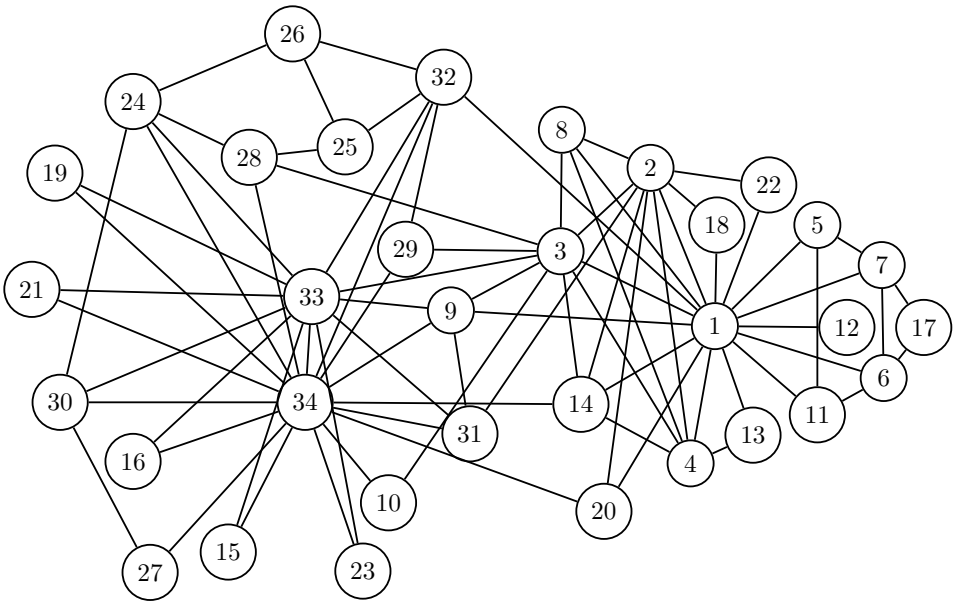


Figure 3.5: Zachary's karate club

Lemma 6. *RDF satisfies the Property 5 of RSM.*

Proof. Suppose G is some graph and G' is same as G but the edges weights in G' are all α times greater than the ones in G. Let A and A' be the adjacent matrixes of G and G' respectively. Then we have $A' = \alpha A$. So for the corresponding degree matrixes D and D', we also have $D' = \alpha D$. Therefore, we have

$$L' = D' - A' = \alpha D - \alpha A = \alpha(D - A) = \alpha L$$

Let $L^\dagger$ and $L'^\dagger$ be the generalized inverse of L and L' respectively. Then by the definition of the generalized inverse, we have

$$LL^\dagger L = L \tag{3.1}$$

$$L'L'^\dagger L' = L' \tag{3.2}$$

Since $L' = \alpha L$, we can simplify Eqn. 3.2

$$\begin{aligned} L'L'^{\dagger}L' &= L' && (\Leftrightarrow) \\ (\alpha L)L'^{\dagger}(\alpha L') &= (\alpha L') && (\Leftrightarrow) \\ L(\alpha L'^{\dagger})L &= L \end{aligned}$$

Comparing with Eqn. 3.1, $\alpha L'^{\dagger}$ has the same function as $L^{\dagger}$. Since the final result does not rely on the choice of the generalized inverse matrix, we can let $L^{\dagger}$ be the one satisfying the equation

$$\alpha L^{\dagger} = L'^{\dagger} \tag{3.3}$$

Hence, we have

$$\begin{aligned} R'_{i,j} &= L'^{\dagger}_{i,i} + L'^{\dagger}_{j,j} - 2L'^{\dagger}_{i,j} \\ &= \alpha L^{\dagger} + \alpha L^{\dagger} - 2\alpha L^{\dagger} \\ &= \alpha(L^{\dagger}_{i,i} + L^{\dagger}_{j,j} - 2L^{\dagger}_{i,j}) \\ &= \alpha R_{i,j} \end{aligned}$$

□

Proposition 1. *ERF is an RSMFSN.*

Proof. We can define that the resistance distance of a pair of vertices is infinite if there is no path between them. Then the proposition is immediate from Lemma 4, Lemma 5 and Lemma 6. □

Therefore, ERF is RSM.

3.4.3 Community detection in Zachary's karate club

The graph we use for demonstration is Zachary's karate club [49], shown in (Fig. 3.5), which is a popular test case in community detection research. We have discussed about

the features of this certain kind of network and also some of the popular networks in the later sections of our discussion.

At the beginning, we need to choose some proper CP, which is the lower bound of the relation strength within the communities. Here, we choose CP = 1.5.

Then, we use Klein-Randic method to compute the resistance distance between each pair of nodes in the network and get the corresponding resistance matrix R.

After that, we get the corresponding *adjoint graph(adj)* from R and remove all the edges whose weights are greater than CP. So we get the *efficient edges graph(EEF)*.

Then we apply Bron-Kerbosch algorithm on ERF and get a list of maximal communities.

If we plot those maximal communities in the original graph, we get Fig. 3.6. Here, we have three maximal communities represented by red, blue and yellow respectively. Some nodes are multicolour, which means they belong to various maximal communities simultaneously.

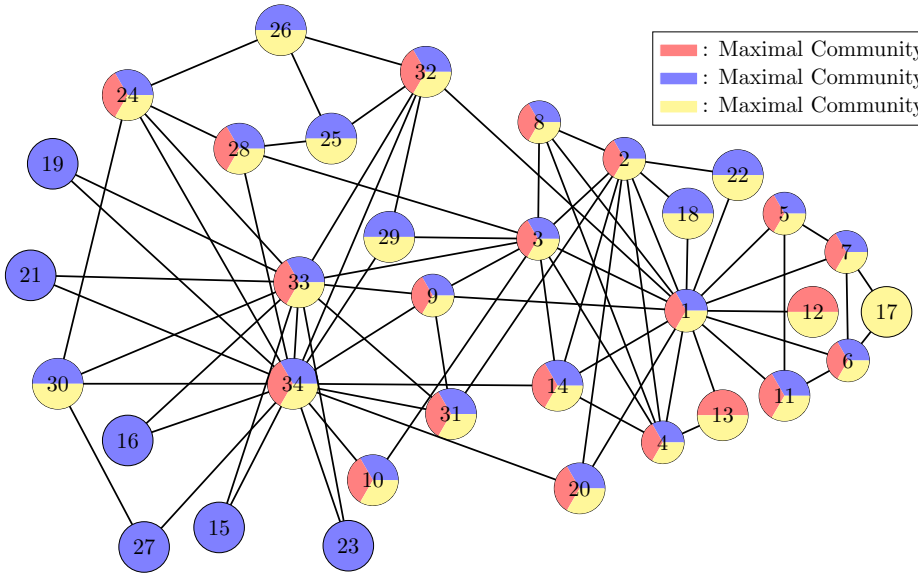


Figure 3.6: Maximal Communities ($CP = 1.5$)

3.5 Discussions

Based on the definition of communities on *RSM*, the role of the community threshold parameter (CP) is very important in the determination of communities. We will see some of the examples to see how these two approaches makes a subtle difference in finding communities.

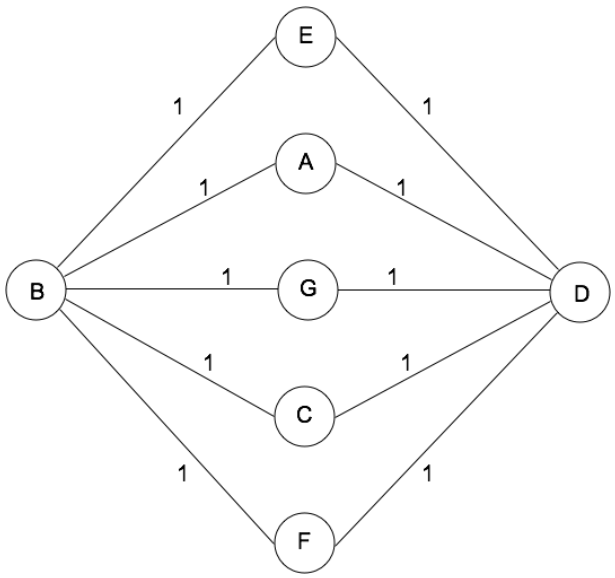


Figure 3.7: An arbitrary weighted network

Example 1. In Fig. 3.7, let B be a super-node which is a community by itself. Although the weights of R_1,R_2,..R_{10} are relatively high (assuming it to be = 1), it is still possible that the node D on the right belongs to the community B on the left since there are enough number of paths between them. Let the community threshold parameter be $\beta = 0.5$. We calculate the ERF between two pair of nodes and if it lies below 0.5, then it can be considered as a community.

By Definition 13, we find

$$f(B,E) = \frac{1}{\frac{1}{1} + \frac{1}{1+\beta}} = \frac{1+\beta}{2+\beta} = 1 - \frac{1}{2+\beta} \geq 0.5(\beta), \tag{3.4}$$

This states that the effective resistance distance between the nodes (B,E) is strictly greater than 0.5 which clearly tells that even though they have a relation/path between them but still it can not be considered as a part of community.

On the other hand, finding the effective resistance between nodes B and D we get,

$$f(B, D) = \frac{2}{5} = 0.4 < \beta, \tag{3.5}$$

This shows that the node D could be the same community as of node B. Here the selection of community parameter is very important and it depends on the interest of the user, how he wants to view the community structure as.

Wu-Huberman's algorithm [42] used a similar model to find communities in a linear time but the detection results completely depend on the choice of the centre node, which may produce some unreasonable results. Although there are some algorithms [25] to find good centre nodes, the reasonability of these algorithm is still open to dispute.

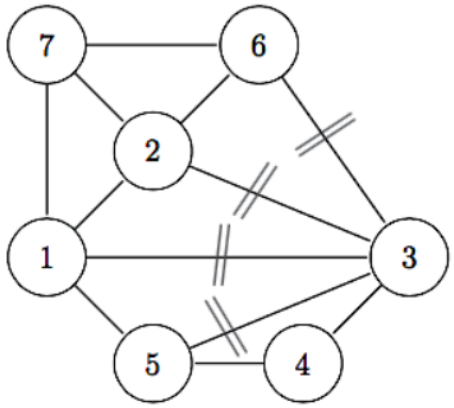


Figure 3.8: Source-Sink node pair selection for an arbitrary network

.

A heuristic approach is followed to pick the source and the sink first which is inconsistent. They are chosen based on the average distance between the two nodes. The one with the farthest makes the highest probability as the source-sink.

On the contrary, in the Fig. 3.8, (2,3), (6,3), (1,3) and (5,3) all have the same average distance between them. This makes it difficult for the algorithm to choose the correct pair of source and sink.

The threshold parameter or the cut between the two communities is suggested at the voltage gap near the middle. We do have a certain voltage drop at each pair of nodes, the largest being at the the nodes having the farthest resistance distance. Reasonably it makes sense to divide the communities at the largest voltage drop but it may give ambiguous results.

If after the selection of the source and sink, we have the equal voltage drop for a certain topology, it becomes difficult for the user to select. Also a good approximation of the graph will lead to correct results. Rather it should depend on the user how he wants to view community structure as.

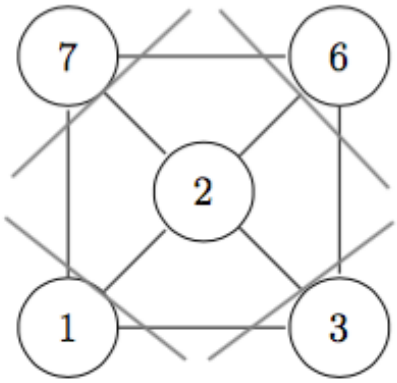


Figure 3.9: An arbitrary unit resistance distance network

.

Most of the algorithms needs an approximation of the graph prior implementation. In Fig. 3.9, each of the nodes are separated by an unit resistance distance.The community parameter in our algorithm if CP ≤ 1 and CP > 1 results in 4 and 1 communities respectively.

The point here we want to state is any algorithm should work well and find the appropriate communities when compared with the ground truth of any arbitrary graph. In that case,our methodology without any prior approximation, takes into account every relation (distance) between every pair of nodes or sets of vertices, rather than the largest or the smallest and finds the communities. Here both the result the number of community as 1 or 4 communities are correct. The community threshold parameter adjusts according to the network and results an appropriate community division structure.

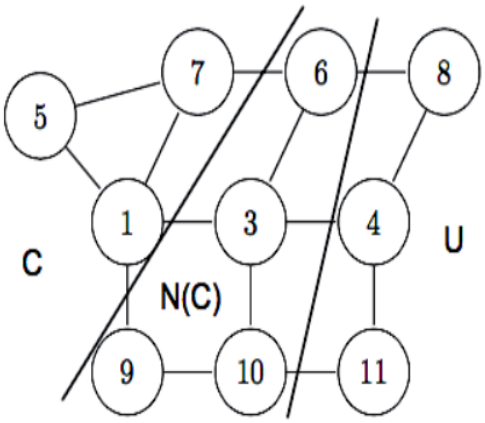


Figure 3.10: An illustration of an arbitrary equal weighted network containing the local community C, its neighborhood $N(C)$, and the external set U.

The boundary detection method takes into account the link-similarity of a supernode, which is a community in the above figure. Nodes (1, 7 and 5) constitutes *C*. *N(C)* represents the corresponding neighbourhood nodes of the supernode. The link similarity is calculated with the nodes that are in the external set *U* (Nodes: 8, 4 and 11). The link similarity is calculated and a boundary is created based upon the maximum-distance between the nodes at *N(C)* and *U*. But here we assume that if all the distance between them are equal, then at such conditions, it becomes trivial using this kind of approach.

3.6 Work Flow of the Algorithm

We take into account an example to visualize the process flow of our algorithm.

Example 2. Suppose we have the following undirected unweighted graph G as shown in Fig. 3.11. Let RSM be SDF and CP be 3.

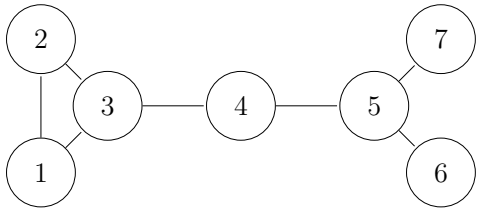
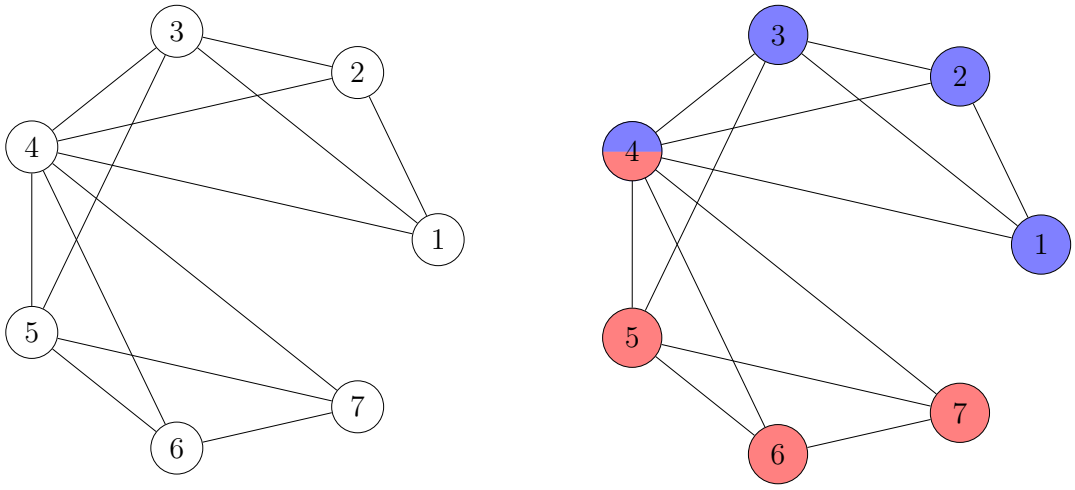


Figure 3.11: Original graph

Then we get the corresponding efficient edges graph as shown in Fig. 3.12(a), and then find all the maximal communities Fig. 3.12(b) in it. In this example, we get two communities and Node 4 is in both of them.

At last, we plot the communities in the original graph as shown in Fig. 3.13.



(a) Efficient Edges Graph

(b) All Maximal Communities

Figure 3.12: Maximal Communities from Effective Edge Graph

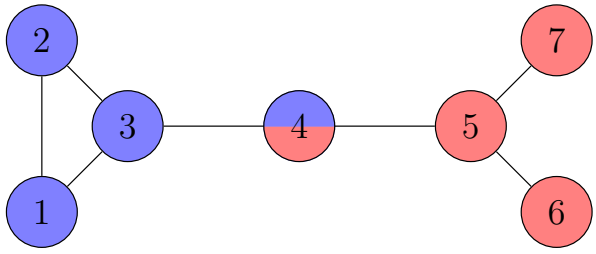


Figure 3.13: Efficient Edges Graph

3.7 Summary

Based on their respective characteristics, we categorized networks into two major types: transmission and similarity networks. We defined each of their attributes and correlated with that of the elemental properties (node, medium) of a community. We defined RSM specifically between the nodes, for each of the networks (RSMFTN and RSMFSN).

Although the definitions of their RSMs are distinct based on the network properties, we presented an unified math model and proposed a general definition of communities stating that RSM is some global property and not limited to specific networks.

We introduced the Community Threshold Parameter (CP) to give a threshold of the relation strength among the nodes, a parameter based on which we can determine the corresponding communities. Based on the proposed definitions for absolute communities, a detection algorithm was implemented.

We studied the Klein and Randic's model and proved Effective Resistance Function (ERF) to be a RSM. As a part of demonstration, we used ERF, being on the RSMs to detect communities in a real-world network.

Chapter 4

Community Evaluation Using Performance Parameters

When any algorithm detects communities, it must be evaluated based upon how accurately it detects it. A very popular evaluation methodology for a community detection algorithm is to test it against an artificial network with "observed" community structure and measure how well the algorithm extracts the underlying mesoscopic organisation. The proposed algorithm's performance will be tested on a variety of topologies (Synthetic and Real-World Networks) based upon the widely used performance parameters to measure the efficiency of any community detection method. We will emphasize the performance of RSM, based upon the identification of overlapping communities (community level) as well as identification of the overlapping nodes (node level) in the networks. We will analyze the behaviour of the community threshold parameter (CP) introduced in the algorithm and how it plays a vital role in the detection of communities in the real-world networks.

4.1 Performance Metrics

4.1.1 Precision, Recall and F-Score

We note that our community detection approach allows for overlaps, and can assign each node to more than one cluster. Furthermore, the ground-truth may also allow for overlaps, when multiple groups were associated with a node.

Therefore, we need to revise the commonly used pairwise precision and recall measures for clustering algorithms [56], in order to create a meaningful measure.

Let O_n be the generated nodes by the network's algorithm and denotes the set of node pairs that share at least one cluster class. Similarly, let O_n^d denote the set of node pairs that are assigned to at least once to the same cluster by the proposed algorithm. Then, we can compute the pairwise precision and recall [57] as follows:

$$Precision = \frac{|O_n^d \cap O_n|}{|O_n^d|}, Recall = \frac{|O_n^d \cap O_n|}{|O_n|}, \tag{4.1}$$

The aforementioned measures of Precision and Recall can be used to to define the pairwise F-measure, which is the harmonic mean of both of them.

$$F\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.2}$$

Remark 10. *By the Definition 4.1, Precision = 1 only when $O_n^d = O_n$. It implies that the set of overlapping node pairs forming communities, detected by our algorithm is same as that of the generated network's algorithm. A well-performed algorithm should achieve higher precision, recall and F-score. It suggests the underlying clustering is of good quality.*

4.1.2 Normalized Mutual Information

Evaluating the quality of a detected partitioning or cover is nontrivial, and extending evaluation measures from disjoint to overlapping communities is rarely straightforward. Unlike disjoint community detection, where a number of measures have been proposed for comparing identified partitions with the known partitions, only a few measures are suitable for a set of overlapping communities. Two most widely used measures are the Normalized Mutual Information (NMI) and Omega Index (OI) [38].

The mutual information between two discrete random variables X, Y measures how much we learn about X if we know Y. It evaluates the goodness of the result. The mutual information ($I(X,Y)$) [58] between two discrete random variables X, Y is given as:

$$I(X,Y) = \Sigma_x \Sigma_y P(x,y) log \frac{P(x,y)}{P(x)P(y)}, \tag{4.3}$$

where $P(X,Y)$ is the joint probability distribution function of X and Y, and $P(X)$ and $P(Y)$ are the marginal probability distribution functions of X and Y respectively.

Unfortunately, the mutual information is unbounded; however, it is common for measures to have values in range (0,1). To address this issue, we can use normalized mutual information [38] to find the mutual information between two partitions. Given a partition (treated as random variable X), all the partitions derived from X by further partitioning (some of) its clusters would have the same mutual information with X, even they could be very different from X.

Hence, Normalized Mutual Information $I_{norm}(X,Y)$ [38] is used as:

$$I_{(norm)}(X,Y) = \frac{2I(X,Y)}{H(X) + H(Y)}, \tag{4.4}$$

where $H(X)$ and $H(X)$ are the entropy for the random variables X and Y respectively.

$I_{norm}(X, Y)$ is 1 if the community structures are identical and is 0, if the community structures are independent.

Lancichinetti et al. [38] has extended the above notion as normalized mutual information (NMI) to account for overlapping between communities

$$I_{(norm)}(X, Y) = \frac{H(X) + H(Y) - H(X, Y)}{(H(X) + H(Y))/2}, \tag{4.5}$$

where $H(X)$ = Labels of the node sets belonging to the same cluster generated by the network's algorithm and $H(Y)$ are that of our proposed algorithm.

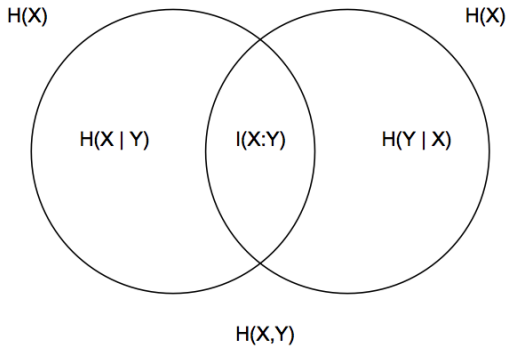


Figure 4.1: Venn diagram for various information measures associated with correlated variables X and Y. The region contained by both circles is the joint entropy H(X,Y). The circle on the left is the individual entropy of H(X), with the left most being the conditional entropy H(X—Y). The circle on the right is H(Y), with the right most being H(Y—X). The violet is the mutual information I(X:Y).

4.1.3 Omega Index

The Rand Index [59] is a way of comparing disjoint clustering solutions that is based on pairs of the objects being clustered. Two solutions are said to agree on a pair of objects

if they each put both objects into the same cluster or each into different clusters. Thus Rand Index can be defined as

$$RI = \frac{m + k}{N} \tag{4.6}$$

where m is the number of times the solutions agree in assigning a pair in the same cluster and k is the number of times the solutions agree in assigning a pair in the different clusters. The Rand Index is ineffective in identifying the overlapping solution as a pair of objects can exist together in more than one community. In those cases, the two solutions might agree on the occurrence of a pair of objects in one community but disagree on the occurrence of that pair in another community. The Rand Index cannot distinguish this distinction.

An improvement to Rand Index is Adjusted Rand Index, which adjusts the level of agreement according to the expected amount of agreement based on chance. However, the Adjusted Rand Index also cannot account for disjoint solutions.

Thus, a new metric Omega Index was introduced which is built upon Rand Index and Adjusted Rand Index. Omega index [59] is the overlapping version of the Adjusted Rand Index. In a similar way, it is based on pairs of nodes in agreement in two covers. A pair of nodes is considered to be in agreement if they are clustered in exactly the same number of communities (possibly none). That is, the Omega index considers how many pairs of nodes belong together in no clusters, how many are placed together in exactly one cluster, how many are placed in exactly two clusters, and so on.

Let $S1$ and $S2$ be the number of communities in covers $C1$ and $C2$ respectively, then Omega Index is defined as:

$$\omega(C1, C2) = \frac{\omega_a(C1, C2) - \omega_b(C1, C2)}{1 - \omega_b(C1, C2)} \tag{4.7}$$

The observed Omega index ω_a is defined as

$$\omega_a(C1, C2) = \frac{1}{T} \sum_{j=0}^{max(S1,S2)} |m_j(C1) \cap m_j(C2)|, \tag{4.8}$$

where T equals $\frac{n(n-1)}{2}$, represents the number of pair of nodes, and $m_j(C1)$, $m_j(C2)$ is the set of pairs that appear exactly j times in a cover $C1$ and $C2$ respectively.

The expected Omega index ω_b is defined as

$$\omega_b(C1, C2) = \frac{1}{T^2} \sum_{j=0}^{max(S1,S2)} |m_j(C1).m_j(C2)| \tag{4.9}$$

Eqn. 4.7 is used for the numerical calculations and for measuring the Omega Index of the proposed algorithm for different networks generated by the LFR Benchmark. In Eqn. 4.7, the numerator is the observed agreement adjusted by expected agreement, while the denominator is maximum possible agreement(= 1) adjusted by expected agreement. The subtraction of the expected value in (4.7) takes into account agreements resulting from chance alone. The larger the Omega index, the better the matching between two covers. A value of 1 indicates perfect matching. When there is no overlap, the Omega index reduces to the Adjusted Rand Index.

4.2 Benchmark Algorithm

The performance of community detection methods are usually measured with constructed synthetic datasets, or real-world networks where community size and degree size of nodes are close to homogenous. But many real life networks have heterogenous structure in terms of community size and degree size, which have a skewed degree distribution with a long tail. Therefore the LFR benchmark has been introduced to generate networks with heterogenous community and degree sizes according to a power law. It has parameters

that allow you to set the exponential value for the power law distribution of community size and degree size, and also a mixing parameter that specifies the fraction of links inside the communities. Comparing methods with each other is not easy and may not be accurate. Even the LFR benchmark may not be very accurate and cannot reflect all properties of real-world networks, because it generates networks according to a set of specified parameters. But it may help us to find the limits of a method in terms of network size, community size, mixing parameter, degree sequence etc. Therefore we choose the LFR benchmark graphs for benchmarking.

The implementation of the benchmark algorithm is described in [38]. In particular, this algorithm is to produce directed unweighted/weighted networks with overlapping nodes. We will briefly discuss on the procedures to run the benchmark algorithm and generate the networks accordingly. The notations used in this chapter are presented in Table 4.1.

Table 4.1: Notations and their Meanings

Notation	Meaning
N	number of nodes
k	average in-degree
$maxk$	maximum in-degree
mu	mixing parameter
$t1$	minus exponent for the degree sequence
$t2$	minus exponent for the community size distribution
$minc$	minimum for the community sizes
$maxc$	maximum for the community sizes
O_n	number of overlapping nodes
Om	number of memberships of the overlapping nodes

- To run the program, type: ./benchmark [FLAG] [P] in the MAC Terminal OS.

- The degree we can set is the in-degree. The out-degree distribution will be chosen but the program close to a delta function. The mixing parameter is the same for both the in-degree and the out-degree, but the latter one might be modified to satisfy the constraints necessary to close the network.

- In this algorithm, we can assign the number of overlapping nodes (option -On) and assign the number of memberships for them (option -Om). The other nodes will have only one membership.

- For instance, executing ./benchmark [flags...] -N 1000 -On 20 -Om 2 will produce a network with 1000 nodes, 980 with only one membership and 20 nodes with two memberships.

Other options:

- To have a random network use: -rand

Using this option will set mu=0, and minc=maxc=N, i.e. there will be one only community.

- Use option: -sup (-inf)

To produce a benchmark whose distribution of the ratio of external in-degree/total in-degree is superiorly (inferiorly) bounded by the mixing parameter.

- In other words, if you use one of these options, the mixing parameter is not the average ratio of external degree/total degree (as it used to be) but the maximum (or the minimum) of that distribution. When using one of these options, what the program essentially does is to approximate the external degree always by excess

(or by defect) and if necessary to modify the degree distribution. Nevertheless, this last possibility occurs for a few nodes and numerical simulations show that it does not affect the degree distribution appreciably.

- Example: To produce a kind of Girvan-Newman benchmark, type: ./benchmark -N 128 -k 16 -maxk 16 -mu 0.1 -minc 32 -maxc 32

Output:

The community size distribution can be modified by the program to satisfy several constraints (a warning will be displayed).

Executing the program produces three files:

1) network.dat contains the list of edges (nodes are labelled from 1 to the number of nodes; the edges are ordered and repeated once, i.e. source-target).

2) community.dat contains a list of the nodes and their membership (memberships are labelled by integer numbers >= 1).

3) statistics.dat contains the in and out-degree distribution (in logarithmic bins), the community size distribution, and the distribution of the mixing parameter (in and out).

- Seed for the random number generator: In the file seed.dat you can edit the seed which generates the random numbers. After reading, the program will increase this number by 1 (this is done to generate different networks running the program again and again). If the file is erased, it will be produced by the program again.

4.3 Working Principle of the Comparison Algorithms

In this section, we will discuss in brief on the working principle of some of the well known community detection algorithms which are used to detect the overlapping communities in complex networks.

- CPM (Clique Percolation Method) is based on the assumption that a network is composed of cliques which overlap with each other. CPM finds overlapping communities by searching for adjacent cliques. As a vertex can be a member of more than one clique, so overlap is possible between communities. The parameter k ($k \sim n$ for n cliques in graph theory) is of utmost importance in finding communities via CPM, empirically small values of k have shown effective results. An efficient implementation of the CPM method is **CFinder**. CPM is suitable for dense graphs where cliques are present. In case there are few cliques only CPM fails to produce meaningful covers. However, it also fails to terminate in many large social networks.

- On the contrary, **GCE** (Greedy Clique Expansion), starts with all maximal cliques as initial communities, and removes communities that are too similar, using a local community quantity measure. Thus it produces communities in which there exists, for each maximal clique, atleast one community that fully contains it.

- Ahn, Bagrow and Lehmann [24] proposed the **LINK** based algorithm to detect overlapping communities.The algorithm first calculates the similarity of all edges of the network and assign each edge to its own community. At each step, the method chooses pair of edges with the largest similarity and merges their respective communities until all edges becomes a single cluster. The history of clustering process is stored in a dendrogram, and the partition with the largest partition density is chosen as the final result. This algorithm is effective in finding the smaller communities, however it fails in giving a larger scope of the topology.

- **CIS** (Connected Iterative Scan) examines each node of the network, adding or removing it, if the density of the set is increased as a result. The scans are repeated until the set is locally optimal with respect to a defined density metric. The choice of the seed sets is not predetermined: they can be the nodes, or the edges of the network. To ensure the connectivity of the identified communities, a number of scans are repeated for a set until no change of the set occurs. Then the set is declared to be a community [60]. Once a scan is finished, the set's connectivity is examined. If the set consists of multiple connected components, it is replaced by the connected component with the highest density, after which the next scan starts. The major disadvantage of this algorithm is that it finds a large number of highly overlapping communities.

- Diffusion-based approaches tackle the problem of community detection using a communication paradigm. They rely on the assumption that information is more efficiently exchanged between nodes of the same community. Therefore, communities can be detected by considering how information is propagated in the network. Community Overlap Propagation Algorithm (**COPRA**) is one of the popular methods followed.

- **COPRA** is an extended version of Label Propagation algorithm, proposed by Gregory [61]. The information takes the form of a label, and the propagation mechanism relies on a vote between neighbours. Initially, each node is labelled with a unique value. Then an iterative process takes place, where each node takes the label which is the most spread in its neighbourhood (ties are broken randomly). This process goes on until convergence, i.e. each node has the majority label of its neighbours. Communities are then obtained by considering groups of nodes with the same label. By construction, one node has more neighbours in its community than in the others. This algorithm is faster than most other algorithms. Gregory's

version to detect mutually exclusive communities in undirected unweighted unipartite networks. However, it is able to handle overlapping communities, for both weighted and bipartite networks.

Based upon the functionality of those algorithms, as all find the overlapping communities, we select them and compare their respective performance with our algorithm on the LFR Benchmark networks and Real-World networks. The simulation setup is listed in the Table 4.2.

Table 4.2: Simulation Environment for calculating Performance Metrics

OS	MacOS
Tool	MATLAB R2013a
Programming Language	JavaScript
Datasets	http://www-personal.umich.edu/ mejn/netdata/
Graph vizualisation software	Graphviz
Graph analysis software	Gephi
Performance Graphs	High-Charts (JS Library)

4.4 Results

We consider performing our tests in both synthetic networks (when ground truth is available) and in the real-world networks (when ground truth is not available). Table 4.1 shows the Benchmark parameters that are essentially used by any algorithm.

4.4.1 Tests in Synthetic Networks

4.4.1.1 Simulation Setup

For synthetic random networks, we adopted the widely used LFR benchmark [62], which allows heterogeneous distributions of node degrees and community sizes.

We generated random networks with size $n = 100$ for each instance. The average degree is kept at $k = 6$. The degree of overlapping is determined by two parameters. O_n defines the number of overlapping nodes and is set to 10% of all nodes. Om defines the number of communities to which each overlapping node belongs and varies from 2 to 8 indicating the diversity of overlap. By increasing the value of Om, we create harder detection tasks. Other parameters are as follows: node degrees and community sizes are governed by the power laws with exponents 2 and 1; the maximum degree is 50; the community size varies from 20 to 100; the expected fraction of links of a node connecting it to other communities, called the mixing parameter μ, is set to 0.3. We generated six instances (25 random generated networks) for each setting.

4.4.1.2 Identifying Overlapping Communities in LFR

As discussed above, we evaluated the following metrics. The generalized *Normal Mutual Information (NMI)* [38] is used specially for overlapping communities.

NMI is a standard measure since NMI (U,V) = 1, if structures U and V are identical and 0 if they are separated. The NMI value ranges between 0 to 1 for any community detection method. Fig. 4.2 shows the plot of the **NMI** of our algorithm vs to the **Om** (Number of the memberships of the overlapping nodes). Ignoring the singleton communities and the unassigned nodes, the NMI ranges around 0.4 to 0.57.

Omega Index is the accuracy on estimating the number of communities that each pair of nodes share between ground truth communities and detected communities. Both NMI and Omega yield the values between 0 and 1. The closer this value is to 1, the

better the performance is.

We have used NMI as a metric for identifying the overlapping communities in LFR. NMI ranges between 0 to 1, with 1 corresponding to a perfect matching.

For the default settings: k=6, μ=0.3 and absolute value of CP = 1.2 (Range:1 $\leq$ CP $\leq$ 1.5), we found that the NMI ranges from 0.4 to 0.57. For Om=2, we got NMI as 0.57 which is better than other algorithms (Link and C-Finder). The stable NMI of our algorithm clearly shows that it outperforms other algorithms over different networks structures(i.e., with different μ's). Comparing the number of detected communities and the average number of detected memberships with the ground truth in the benchmark helps in understanding the results. 0.57 NMI suggests that the algorithm achieves 57% of identifying correctly the overlapping communities in LFR when compared to the ground truth data available. On the contrary, when Om=8 (At most 8 communities to which the overlapping nodes belong to), we get NMI=0.4 which is better than C-Finder, COPRA and Link. It shows with the harder detection tasks assigned, RSM is able to achieve 40 % of correctly identifying the overlapping communities in LFR. The decrease in NMI is also relatively slow, indicating that RSM is less sensitive to the diversity of Om.

4.4.1.3 Identifying Overlapping Nodes in LFR

Identifying nodes overlapping multiple communities is an essential component of measuring the quality of a detection algorithm. To provide precise analysis, we consider the identification of overlapping nodes as a binary classification problem. A node is labelled as overlapping as long as Om $>$ 1. Within this framework, we can use F-score, which combines precision and recall and defined as in Eqn. 4.2. Recall is the number of correctly detected overlapping nodes divided by O_n, and Precision is the number of correctly detected overlapping nodes divided by O_n^d. F-score reaches its best value at 1 and worst score at 0.1.

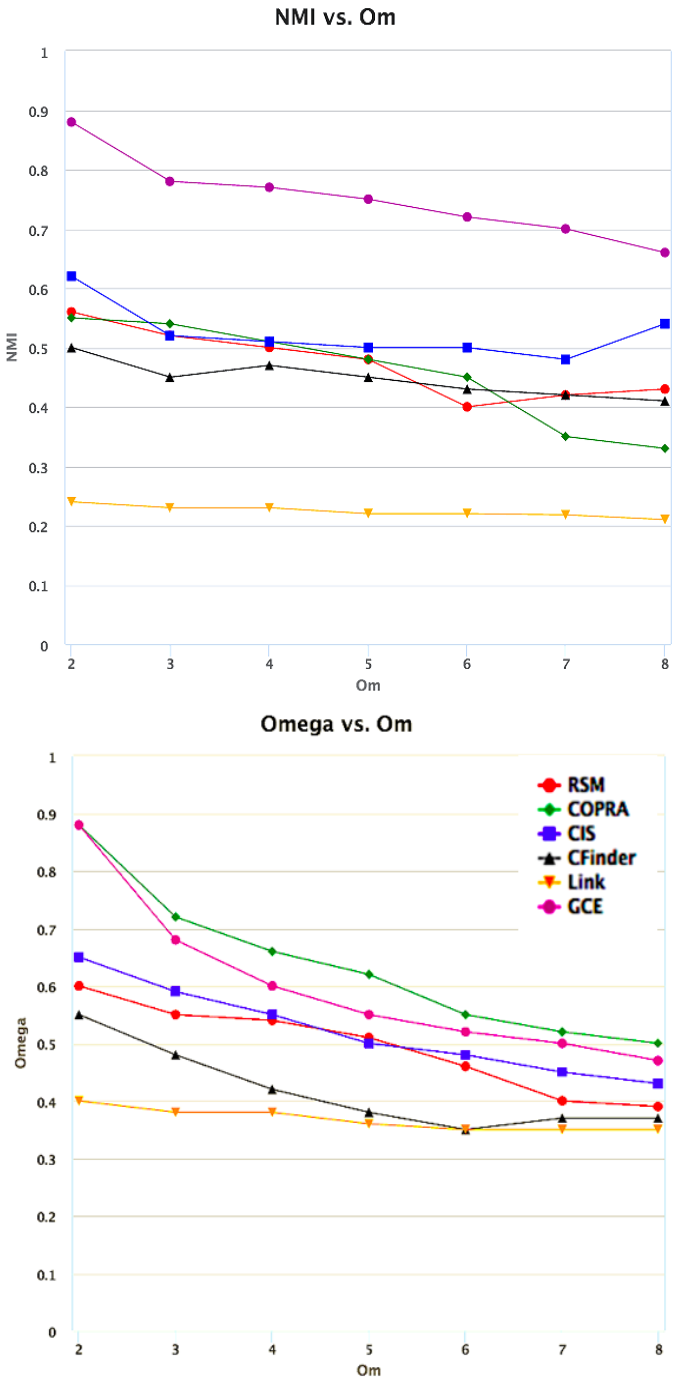


Figure 4.2: NMI and Omega Index as a function of the number of memberships Om in LFR

A well-performed algorithm should achieve high precision, recall, and F-score at the same time. We calculated the related parameter values and plotted them w.r.t to Om (number of memberships of the overlapping nodes) for our algorithm and compared them with the other algorithms as shown in the Fig. 4.3. By the Definition (4.1), Precision=1, only when $O_n^d = O_n$. It implies that the set of overlapping node pairs forming communities, detected by our algorithm (RSM) is same as that of the generated network's algorithm (LFR Benchmark) for directed weighted graphs with possible overlapping communities [29].

From Fig. 4.3, we see that the Precision=1 for Om=2 and decreases gradually with the increase in Om. Om, the most important parameter for our test, varies from 2 to 8 indicating the diversity of overlapping nodes. By increasing the value of Om, we create harder detection tasks. Even when Om=7, we see Precision=0.5 which implies that our algorithm is effective at least by 50% in correctly identifying the set of overlapping node pairs forming communities when compared with the ground truth available.

We use pairwise F-score as a measure of accuracy, which is the harmonic mean of Precision and Recall. Although the F-score of our algorithm has a negative correlation with the Om, like other algorithms with the increase in Om, but it achieves a reasonable better performance.

Particularly, when the Om is less than 3, the average F-score of our algorithm is almost 0.90. This means our methodology can discover exactly the full local community structure from the given node. It achieves 90% of identifying the correct set of overlapping nodes from the overlapping communities formed. This is due to the high precision achieved by the proposed algorithm. Also the score is reasonably better till when Om ≤ 7 which again shows that the algorithm can able to find results with the harder community detections. But beyond that, the proposed algorithm suffers performance degradation and becomes ineffective to detect communities.

The high precision of our algorithm (also CFinder and GCE for Om = 2) shows that clique-like assumption of communities helps us to identify overlapping nodes. Taking both community level performance NMI and node level performance (F-score) into account, we conclude that our algorithm performs well in the LFR benchmark networks.

Remark 11. *A well-performed algorithm should achieve higher precision, recall and F-score at the same time. Taking into account both community (NMI=0.5 and OI=0.6) and node level performance (PFR=1) when Om=2, suggests that our algorithm performs well in LFR benchmark networks. Although algorithm has a negative correlation with the increase in Om, it is still stable and exhibits a better performance than other algorithms.*

4.4.2 Tests with Real-World Social Networks

We took some of the real-world network examples to evaluate the performance of our proposed algorithm.

4.4.2.1 Zachary Karate Club

The first example is taken from one of the classic studies in social network analysis. In the late 1970s Wayne Zachary observed social interactions between the members of a karate club at an American university [49]. The karate club network represents friendships between 34 members of the karate club at a US university, as recorded over a two-year period by Zachary. During the course of the study, the club split into two factions centered around the administrator and the teacher as a result of a dispute within the organization, and the members of one fraction left to start their own club. Finally two communities were partitioned. Many algorithms for community detection found the correct partition of this network. One of the communities represented individuals who ended up aligning with the club's administrator after the fission of the club, and the other represented those who aligned with the instructor.

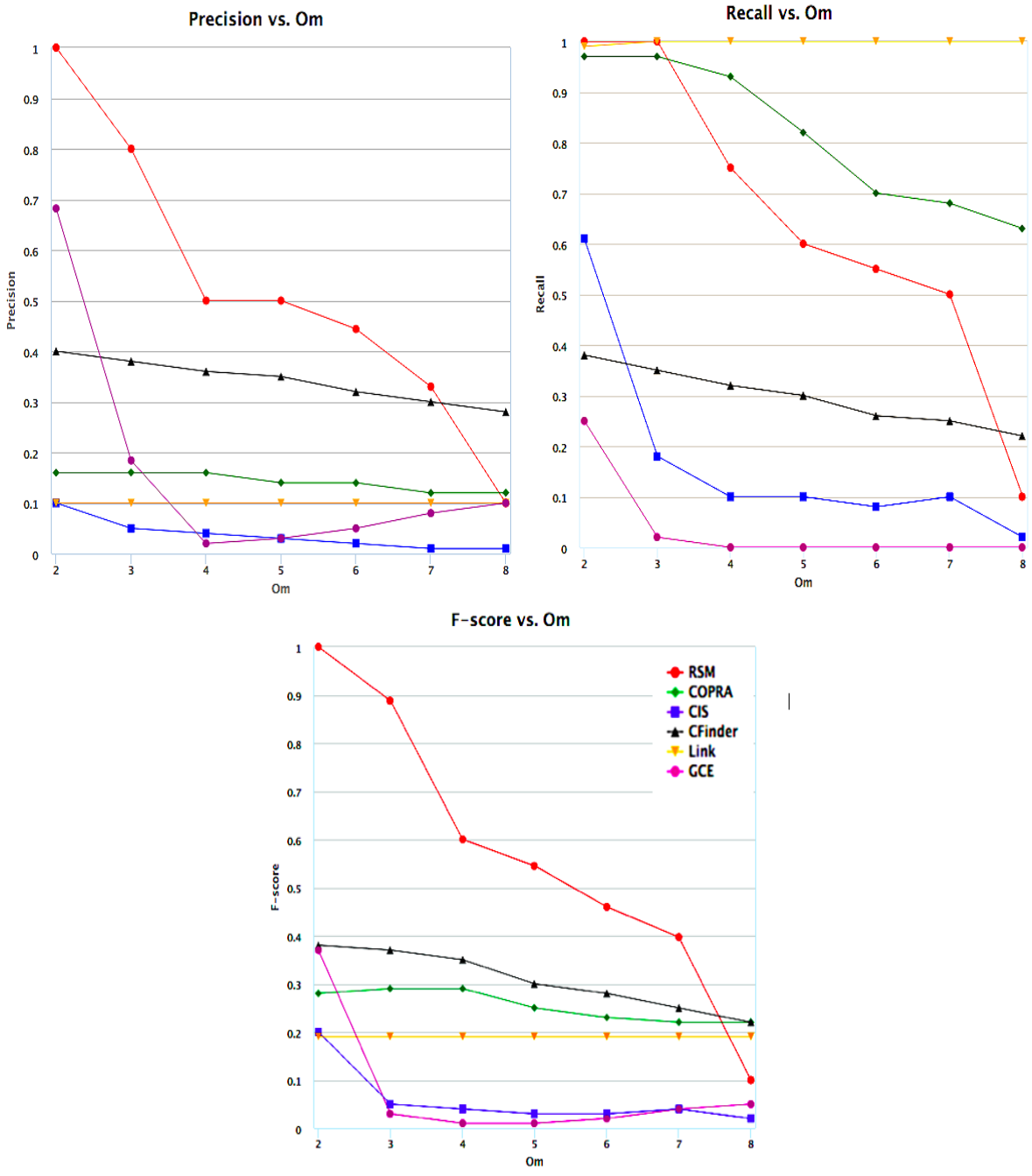


Figure 4.3: Comparison of Precision, Recall and F-Score among various community detection methods for random network generated for default settings k=6, μ=0.3 and CP=1.2 (Range of CP=1 to 1.5).

Nevertheless, further studies on this network showed that if the constraint of having just two communities is relaxed and we search for partitions with more communities, then there is a better decomposition [60]. In Fig. 4.4, we have three maximal communities represented by red, blue and yellow respectively. Some nodes are multicolour, which means they belong to various maximal communities simultaneously.

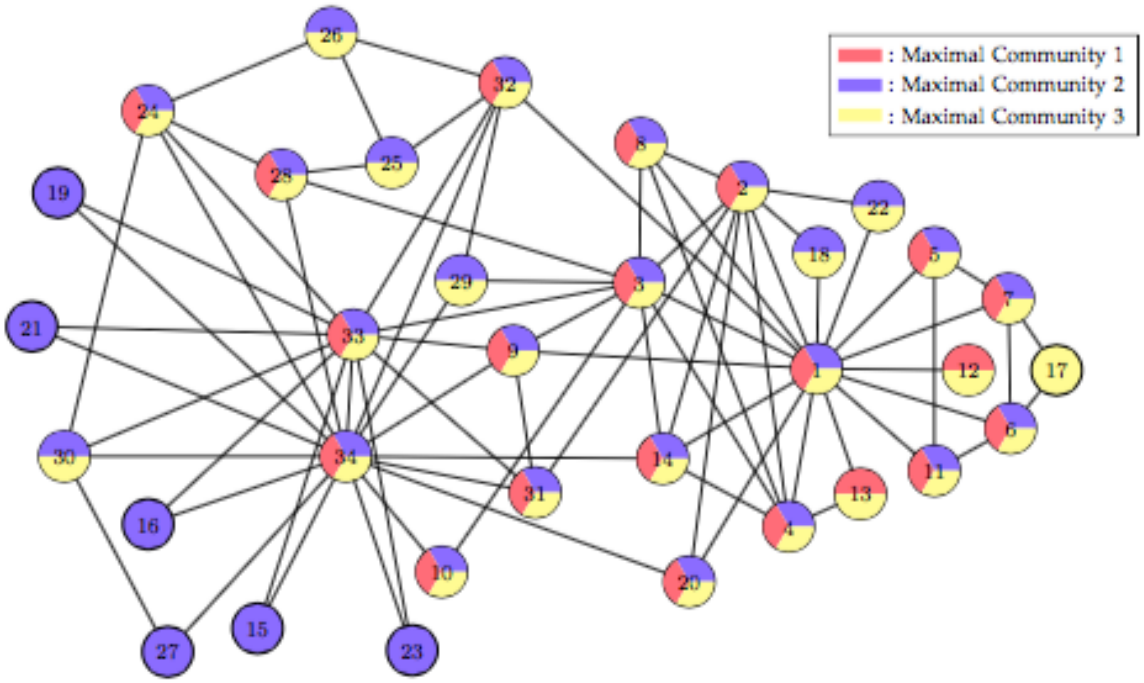


Fig. 1: Maximal Communities ($CP = 1.5$)

Figure 4.4: The maximal communities found in the Zachary Karate Club Network by RSM for CP = 1.5

4.4.2.2 High school Friendship Networks

We also used a set of high school friendship networks [50] created by a project funded by the National Institute of Child Health and Human Development. With known attributes, we verified the output of our algorithm. As shown in Fig. 4.5, there is a good agreement between the found and known partitions in term of student's grades. We see that by our algorithm, the grade 9 community is further divided into two subgroups. The larger group contains only white students, while the smaller group demonstrates race diversity. These two groups are connected partially via an overlapping node. It also suggests that overlapping nodes only exist on the boundaries of communities. A few overlapping nodes are assigned to three communities, while the others are assigned to two communities (i.e., their Om is 2).

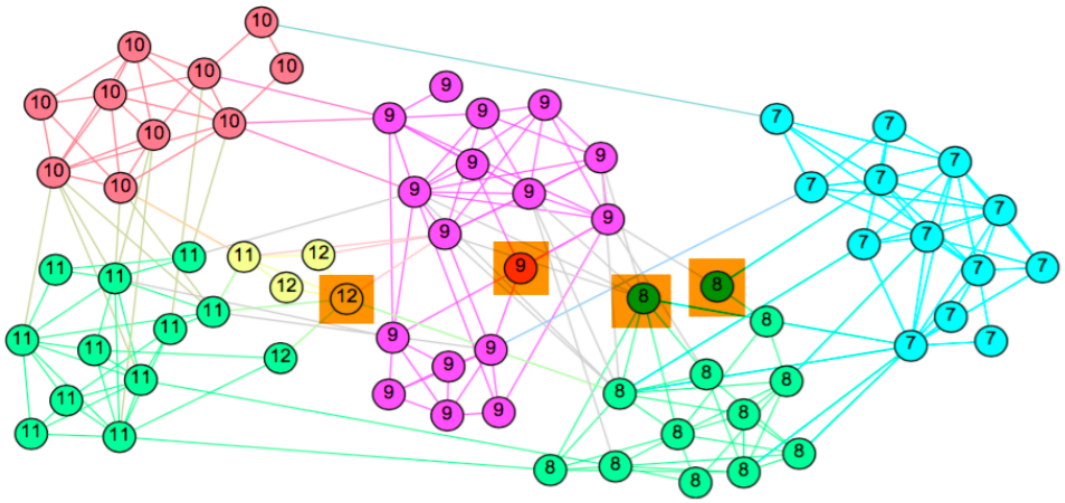


Figure 4.5: High school network (n = 69, k = 6.4). For an absolute value of CP=2.2 (Range: $2 \leq$ CP 2.5), labels are the known grades ranging from 7 to 12. Colors represent communities discovered by RSM. The overlapping nodes are highlighted by orange color .

Further more we also carried our experiments on the social network of Dolphins [1] and some large and highly sparse networks like Peer to Peer Networks to test the performance of our algorithm on different kind of networks.

Table 4.3: Social Networks in the tests

Network	Number of nodes (n)	Average Density (k)
Karate (KR)	34	4.5
Dolphins (DP)	62	5.1
Highschool (HS)	69	6.3
P2P	62561	2.4

4.4.2.3 Identifying Overlapping Communities in Real-World Social Networks

To evaluate the performance of our algorithm in detecting communities in real-world networks, we used the overlapping modularity function (Q_{ov}) proposed by Nicosia [37], which is an extension of Newman's modularity. Higher values indicates a significant overlapping community structure relative to the null model.

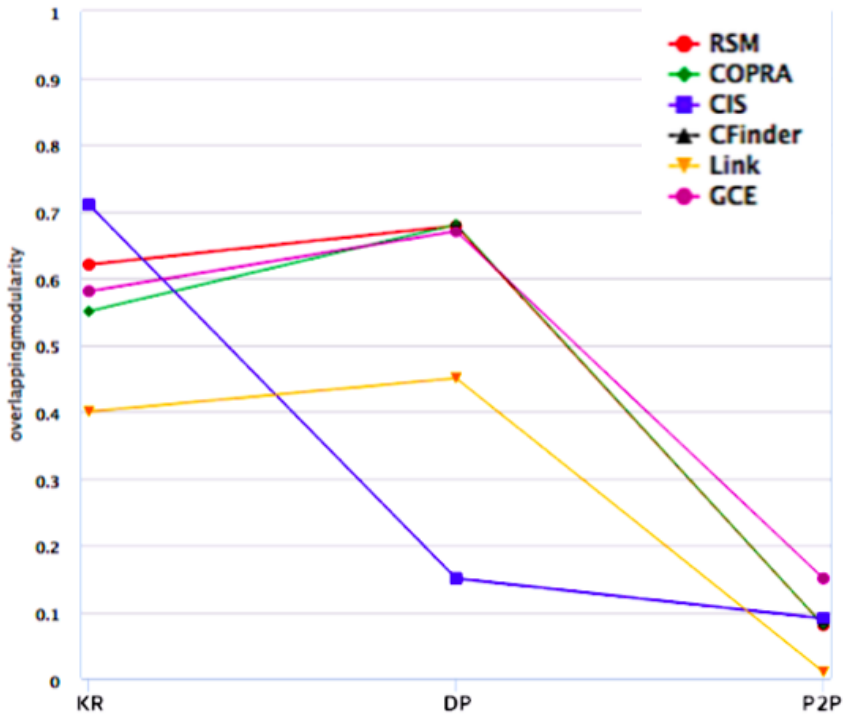


Figure 4.6: Overlapping Modularity Q_{ov} for different datasets for absolute value of CP=1.56 (Range of CP: $1.1 \leq \text{CP} \leq 1.8$).

As we see in Fig. 4.6, RSM achieves a stable modularity score Q_{ov} for Karate club and Dolphin networks. However, we tend to see fluctuations in the performance graph when we examine on the sparse networks (P2P). It detects very few communities or single large community. Increasing our community threshold parameter for these kind of networks so as to cover the whole results in this kind of single giant community. On the contrary, on the smaller data sets or for the dense networks (k $\sim$ 10), it proves to be very effective in detecting the communities, better than the other algorithms. Therefore, we expect better results from our algorithm on similar kind of networks (precisely number of nodes (n) $\sim$ 1000).

Remark 12. *We have used ERF (proved earlier as one of the RSM's) for our simulations and we found that it doesn't perform well on sparse networks. [63] can be referred for a better analysis for this kind of observation.*

We have used Q_{ov} function [37], the value of which varies between 0 and 1. The larger the value is, the better the performance of the algorithm is. We see here that Q_{ov} varies with the density and the number of nodes in the network. For DP network with n=62 and k=5.1 (higher density than KR network), we achieve Q_{ov}=0.68 but for P2P network (n=62561 and k=2.4) which is regarded as a sparse network, the proposed algorithm yields a value of 0.08. For a CP=1.56, we get Q_{ov}=0.68 closer to 1 for dense network (k $<$ 5) where as for sparse networks (k $\leq$ 2) and number of nodes (n $\geq$ 2000), we get poor Q_{ov} values.

Remark 13. *To tackle with these kind of complex networks independently, we could tune our CP on a range of 0.2 to 0.8. As a matter of fact, a decrease in CP will yield more communities for the sparse networks but the accuracy of those communities is indecisive. To be more precise, if there is a network of 100 nodes, a further decrease in CP from its absolute value (varies with different networks) will yield more communities but that might result in considering each node as a single community which is quite an unreasonable approach, affecting the quality of overlapping communities as well.*

Section 4.5 presents a brief analysis on the observations that could be realized using our network model.

4.5 Observation

4.5.1 Choosing the Overlapping Threshold

The community threshold parameter CP is the unique input parameter required by our framework and selection of an appropriate value plays an important role in accessing

RSM's performance in both synthesized and real-world datasets. To best determine this threshold, we run our algorithm on all generated networks with different values of CP and recorded the similarities between embedded systems and detected communities via NMI scores in Fig. 4.2. As depicted in the figure, the best values for CP ranges from 1.1 to 1.5, among which the best value for CP is 1.325 for which the NMI corresponds to 0.57, for Om equals to 2.

Remark 14. *Given a network with certain average density (k), we could decrease CP to get better NMI and Omega values (closer to 1), as a result of which the number of communities would be high but inaccurate. We could get desired results but that would be a greedy approach and this tradeoff can quite affect the quality of the overlapping communities.*

4.5.2 CP vs Density of the Network

The one important characteristic of a complex network is its density. The increase in mixing parameter (μ), increases the the number of connection between the nodes inside the network/Node-degree (k) and hence the density of the network generated by the LFR Benchmark algorithm for a particular setting (keeping the number of nodes (n) constant) will increase. Also for a real-world network with the given density, CP of our proposed algorithm can handle accordingly to give the desired results.

Table 4.4 shows the how the community threshold parameter can be adjusted to detect the exact or appropriate number of communities for a given network (dense/sparse).

Given a very dense network ($k > 6$), it becomes difficult to differentiate between all the communities, so our algorithm can tackle this problem smartly by taking the advantage of the CP. So, in case of a dense network, we can just lower the community threshold parameter so as to detect more communities having less (appropriate) number of nodes in each community.

Table 4.4: Variation of Community Threshold Parameter vs Density of the Network

Density of the network(k)	CP	# of Communities
$Dense(4.5 \leq k \leq 9)$	$\downarrow$	$\uparrow$
$Sparse(2 \leq k < 4.5)$	$\uparrow$	$\downarrow$
$Dense(4.5 \leq k \leq 9)$	$\uparrow$	$\downarrow$
$Sparse(2 \leq k < 4.5)$	$\downarrow$	$\uparrow$

Whereas, having a large value of community threshold parameter on the dense network will only make things hard for the algorithm to detect the communities, as a result it will detect fewer communities with large number of nodes which is not sensible.

4.5.3 CP vs Real-World Networks

For the detection of communities for the Zachary Karate Club and the High school network, we got the appropriate results with the absolute value of CP as 1.5 and 2.2 respectively. The simulation results on the different real-world networks also suggested that the range of the CP varies between 1 to 3 for the network size with the number of nodes (n) ≤ 100.

4.6 Summary

In this chapter, we discussed on the performance metrics and its necessity in determining the performance of the detection algorithm proposed. Based upon the benchmark parameters defined, the benchmark algorithm and the proposed algorithm was implemented on synthetic networks. We analyzed both the node level (PFR) and community level (NMI and OI) performance and found promising results on synthetic networks. Based on an

absolute value of CP for different complex networks, we performed our simulation on the real-world networks and detected the respective communities. We analyzed the variation of the community threshold parameter with the different kinds of complex networks specifically with their properties (dense, sparse, networks with large number of nodes). The simulation results were inferred accordingly and a scope for probable future research objective was put on.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we analyzed various complex networks and formulated an approach for defining communities. An uniform network generator to determine the relation strength among the entities in the network was designed and implemented, based on which we detected the communities. An algorithm was proposed for detecting maximal communities and implement it on a network where no explicit community structure is available. Based on the common characteristics, we categorized the complex networks into two types: transmission networks and similarity networks. We also proposed the use of Performance Metrics and the benchmarking of our detection algorithm was implemented to determine its performance in various communication networks. The following can be concluded:

1. Two corresponding relation strength measurements (RSMFTN and RSMFSN) are defined. We reduced both the network types into the same graph model.

2. We provided a general community definition based on the graph model. Our definition considered each of the vertices of a particular topology in a network graph

rather than a particular subgraph, cluster or a snapshot of it.

3. A detection algorithm is derived based on the general community definition.

4. The Klein and Randic's Model is demonstrated and ERF was proven to be one of the RSM.

5. We considered some arbitrary networks to show the process flow of our algorithm and identified the instances where the other approaches fail.

6. We gave a demonstration to show how the algorithm works in a real world network.

   Our study gives a general procedure to detect community structures in the concrete real networks. Readers can specialise our general algorithm to derive their own community detection algorithms according to the problems confronting them.

Followed by the general definition of community, the latter part of our research involves the performance parameters determining the efficiency of our approach. The following can be concluded:

1. The results of performance parameters (Pairwise Precision, Recall and F-Score) and (NMI, OI) for identifying the overlapping nodes and communities in a network were evaluated and we found promising results.

2. The Community threshold parameter plays an important role in detecting the communities. We observed how the selection of values for the threshold parameter varies with the distinct characteristics of the network and favours our approach.

3. The working principles of the algorithms under study, were discussed and we compared each of their performance with the proposed algorithm. We observed that it showed better performance in certain networks with lesser number of nodes but as as we increase the membership between the nodes, the performance is affected.

4. Experiments on synthetic and real world data traces shows good results. We identified the range for community threshold parameter which yields correct results for the detection of communities in real world networks. The High *Precision* of our algorithm depends on the selection of a suitable CP to produce favourable results.

5.2 Limitations and Future Work

- Generally, RSMs consider the whole network topology and so does the algorithm to find the maximal community structures. While the algorithm gives the accurate results, the complexity of it is NP. So our algorithm is not expected to support the community detections in dynamic networks. Besides, the definition we give here is based on the absolute strengths among the nodes. So the users should always give a proper community threshold parameter (CP), which is hard to find sometimes.

- Similar to the methodology using which SDF is proved as a RSMFTN, many typical RSMFTNs can be derived from existing functions. Besides, the definition based on the absolute relation strength should derive a corresponding definition based on the relative relation strength. The key point is how to give a general definition of the neighbour nodes when applying different RSMs.

- We got our desired results using ERF as one of the RSMs for the detection of communities, there should be some more typical functions that could be a part of RSM and effective in finding communities in dynamic networks.

- It would be interesting to see how we can apply the properties of RSMs in mobile networks and find communities in the mobile datasets as a part of the future application.

Bibliography

[1] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, *Behavioral Ecology and Sociobiology*, vol. 54, pp. 396–405, 2003.

[2] I. Derényi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical review letters*, vol. 94, no. 16, p. 160202, 2005.

[3] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Physical Review E*, vol. 70, no. 2, p. 025101, 2004.

[4] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

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

[6] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[7] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their

use in designing approximation algorithms," *Journal of the ACM*, vol. 46, no. 6, pp. 787–832, 1999.

[8] S. Arora, S. Rao, and U. Vazirani, "Expander flows, geometric embeddings and graph partitioning," *Journal of the ACM*, vol. 56, no. 2, p. 5, 2009.

[9] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[10] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra and its Applications*, vol. 421, no. 2, pp. 284–305, 2007.

[11] M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7327–7331, 2007.

[12] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[13] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 2, pp. 75–174, 2010.

[14] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 6, pp. 291–307, 1970.

[15] A. Hlaoui and S. Wang, "A direct approach to graph clustering." *Neural Networks and Computational Intelligence*, vol. 4, no. 8, pp. 158–163, 2004.

[16] E. R. Barnes, "An algorithm for partitioning the nodes of a graph," *SIAM Journal on Algebraic Discrete Methods*, vol. 3, no. 4, pp. 541–550, 1982.

[17] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[18] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, p. 066133, 2004.

[19] B. Xiang, E.-H. Chen, and T. Zhou, "Finding community structure based on subgraph similarity," *Complex Networks*, vol. 207, pp. 73–81, 2009.

[20] A. Medus, G. Acuna, and C. Dorso, "Detection of community structures in networks via global optimization," *Physica A: Statistical Mechanics and its Applications*, vol. 358, 2005.

[21] J. Reichardt and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a potts model," *Phys. Rev. Lett.*, vol. 93, p. 218701, Nov 2004.

[22] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Jun. 2008.

[23] P. J. Bickel and A. Chen, "A nonparametric view of network models and newman-girvan and other modularities," *Proceedings of the National Academy of Sciences*, vol. 106, no. 50, pp. 21 068–21 073, 2009.

[24] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[25] R. Guimera, S. Mossa, A. Turtschi, and L. A. N. Amaral, "The worldwide air transportation network: Anomalous centrality, community structure, and cities'

global roles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 22, pp. 7794–7799, 2005.

[26] J. Chen and B. Yuan, "Detecting functional modules in the yeast protein–protein interaction network," *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.

[27] Y. Dourisboure, F. Geraci, and M. Pellegrini, "Extraction and classification of dense communities in the web," *Proceedings of the 16th International Conference on World Wide Web*, pp. 461–470, 2007.

[28] G. Flake, S. Lawrence, C. Giles, and F. Coetzee, "Self-organization and identification of web communities," *Computer*, vol. 35, no. 3, pp. 66–70, 2002.

[29] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[30] M. Newman, "Detecting community structure in networks," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, 2004.

[31] L. Freeman, "The development of social network analysis," *A Study in the Sociology of Science*, 2004.

[32] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, 1967.

[33] B. Scholkopf, J. Platt, and T. Hofmann, "Fundamental limitations of spectral clustering," *Proceedings of the 2006 Conference in Advances in Neural Information Processing Systems*, pp. 1017–1024, 2007.

[34] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[35] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[36] T. Evans and R. Lambiotte, "Line graphs of weighted networks for overlapping communities," *The European Physical Journal B*, vol. 77, no. 2, pp. 265–272, 2010.

[37] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 03, p. P03024, 2009.

[38] 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.

[39] J. Baumes, M. Goldberg, and M. Magdon-Ismail, "Efficient identification of overlapping communities," in *Intelligence and Security Informatics*. Springer, 2005, pp. 27–36.

[40] S. Kelley, *The existence and discovery of overlapping communities in large-scale networks*. Rensselaer Polytechnic Institute, 2009.

[41] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," *Nature*, vol. 444, no. 7305, pp. 23–65, 2010.

[42] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 331–338, 2004.

[43] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," *Computer*, vol. 35, no. 3, pp. 66–70, 2002.

[44] A. A. Bulatov, "The complexity of the counting constraint satisfaction problem," *Journal of the ACM*, vol. 60, no. 5, p. 34, 2013.

[45] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of web communities," *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 150–160, 2000.

[46] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Phys. Rev. Lett.*, vol. 96, p. 114102, 2006.

[47] D. Li, I. Leyva, J. A. Almendral, I. Sendiña Nadal, J. M. Buldú, S. Havlin, and S. Boccaletti, "Synchronization interfaces and overlapping communities in complex networks," *Phys. Rev. Lett.*, vol. 101, p. 168701, 2008.

[48] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proceedings of the 17th International Conference on World Wide Web*. ACM, 2008, pp. 695–704.

[49] W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[50] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2012, pp. 25–36.

[51] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th International Conference on World wide web*. ACM, 2010, pp. 591–600.

[52] L. Danon, A. DÃaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[53] C. Bron and J. Kerboscht, "Finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, pp. 575–577, 1973.

[54] D.J.Klein and M.Randic, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, pp. 81–95, 1993.

[55] E. T. Wong, "Generalised inverses as linear transformations," *The Mathematical Gazette*, vol. 63, no. 425, pp. 176–181, 1979.

[56] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: a discriminative approach," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and data mining*. ACM, 2009, pp. 927–936.

[57] G.-J. Qi, C. C. Aggarwal, and T. Huang, "Community detection with edge content in social media networks," in *IEEE 28th International Conference on Data Engineering, 2012*. IEEE, 2012, pp. 534–545.

[58] G. J. Chaitin, *Information, randomness & incompleteness on algorithmic information theory*. World Scientific, 1990.

[59] L. M. Collins and C. W. Dent, "Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions," *Multivariate Behavioral Research*, vol. 23, no. 2, pp. 231–242, 1988.

[60] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, p. 43, 2013.

[61] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[62] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[63] A. Radl, U. Von Luxburg, and M. Hein, "The resistance distance is meaningless for large random geometric graphs," in *Proc. Workshop on Analyzing Networks and Learning with Graphs*, 2009.