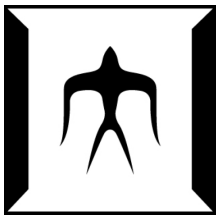
論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Community Detection Algorithms for Scale-free Networks
著者(和文)	Jarukasemratana Sorn
Author(English)	Sorn Jarukasemratana
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第9663号, 授与年月日:2014年9月25日, 学位の種別:課程博士, 審査員:村田　剛志,佐藤　泰介,秋山　泰,篠田　浩一,藤井　敦
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9663号, Conferred date:2014/9/25, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Community Detection Algorithms for Scale-free Networks

Sorn Jarukasemratana

August 2014



Department of Computer Science
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Thesis Committee:
Tsuyoshi Murata
Taisuke Sato
Yutaka Akiyama
Koichi Shinoda
Atsushi Fujii

Submitted in partial fulfillment of
the requirements for the degree of

Doctor of Engineering

Keywords: Community detection algorithms, Scale-free networks, Centrality

To my family and my friends.

Abstract

In the study of complex networks, community structure is one of the most important topics. Communities are groups of nodes that have many edges connecting among them while there are few edges connecting to other nodes in different communities. Finding communities can yield many useful insights about the datasets because nodes that belong to the same community usually share the same attributes.

Scale-free network is introduced by Barabási (Barabási and Bonabeau (2003)). Researchers claimed that many real-world networks contain a scale-free property. However, many community detection approaches such as modularity optimization approach are not corresponded to scale-free property.

In this thesis, we focus on developing community detection algorithms based on the unique characteristics of scale-free networks, and propose two novel approaches. The first approach is based on node centrality and the second approach is based on edge weighting scheme.

Our first contribution is to propose a method for detecting community structures based on node centrality and edge distance. In this algorithm, communities are formed from hub nodes. Thus communities with scale-free property can be identified correctly. The method does not contain any random element, nor requires any pre-determined value such as the number of communities. Our experiments have shown that our algorithm is better than those based on modularity optimization in both real-world and computer generated datasets.

Our second contribution is to propose an edge-weight based method for community detection in scale-free networks. The difference from our first contribution is that the first approach focuses on nodes, while our main focus in the second contribution is on edges. We believe that in scale-free networks, each edge is

not equal. Edges that connect to hub nodes are more important and should be weighted more than edges that connect to other nodes. The benefit of this approach is that our method can work on both scale-free, non scale-free, and mixed scale-free networks - networks that contain both scale-free and non scale-free communities.

To evaluate our methods, we use NMI - Normalized Mutual Information - to measure our results on both synthetic and real-world datasets. We compare our methods with both scale-free and non scale-free community detection methods. As it shown in this thesis, utilizing the characteristics of scale-free networks for designing community detection algorithms can increase the accuracy of the community detection results.

Acknowledgments

Firstly, I want to express my sincere thanks and gratitude to my advisor Associate Professor Murata for his support, patience, and insightful advice during my study at Tokyo Institute of Technology. I am also very grateful to Professor Sato, Professor Akiyama, Professor Shinoda, and Associate Professor Fujii for reviewing and evaluating my thesis.

Secondly, I would like to thank Japanese Government Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for the generous scholarship during my 3 years in Japan. Without this financial support, I would not be able to support my own living in Japan.

Finally, I want to thank my friends and my family for the support both physically and mentally through my study in Tokyo Institute of Technology. Without their supports, it would be difficult for me to get through life in foreign country.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is dedicated to creating community detection algorithms for scale-free networks. In this chapter, we introduce two fundamental ideas of this thesis, community detection and scale-free networks. The outline of this thesis is explained at the end of this chapter.

1.1 Community Detection in Networks

Networks or graphs are the representation of objects and their relations. Nodes or vertices are the representation of the objects while edges or lines are the representation of the relations. Many real-world complex systems can be studied as graphs, such as in biology, physics, or information science (Fortunato (2010)). Communities in networks refer to groups of nodes that are closely connected together and probably share common characteristics or have similar roles within the networks. Community detection is used to find such communities. The example of community detection result is shown in Figure 1.1. The network used in Figure 1.1 represents members from a karate club. This network collected by Zachary in 1953 (Zachary (1977)) is the story about members of this karate club that got into a dispute and divided into two groups. Nodes in the graphs represent members of the club, while edges represent the relation among members. The first group leader is Mr. Hi and another group leader is John. Red nodes are group of members that support Mr. Hi and blue nodes are members that support John. It can be seen that these two persons act as hubs for each group.

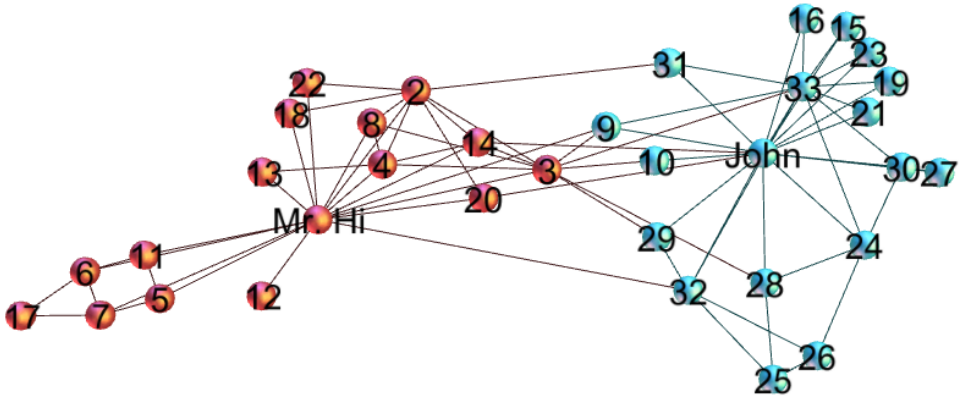


Figure 1.1: Zachary's karate club network.

1.1.1 Fundamental Properties of Community Detection

There are many subjects to consider when performing community detection. We like to explain about the fundamental properties of community detection first. This is because community detection in each field of studies normally has different interpretation of each term.

Definition of Community

One thing that needs to be discussed is the definition of community. There is no clear cut definition of community. Usually it is depended on the purpose of the community detection. In consensus, communities mean groups of nodes that have more edges among themselves than edges between different communities. However, in real practices, communities are usually defined by the algorithms without any a priori definition (Fortunato (2010)). For this thesis, communities mean groups of nodes that are closely connected together by edges and shared the same attributes based on the ground truth value of the network. For example, in karate network, members are in the same community if they support the same person, or in dolphins network, dolphins that belong to the same pod are considered in the same community. Communities can also be divided by its overlapping characteristic. Overlapping communities refer to communities that can contain nodes from another communities and each node can belong to more than one communities. This is different from disjoint communities where there is no overlap and

each node can only belong to one community. The scope of this thesis is on disjoint communities only. There are more definition of communities other then the one we just discussed. For example, when community is focused on structural properties, communities are defined as a group of nodes that group up in a certain specific structure. This type of community detection are usually found in biology or chemistry especially when some type of proteins or atoms group up into a certain structure in order to transform or become active.

Network Models

Another important aspect of community detection is the model of the networks. The most common model is the unipartite networks where there are only one type of nodes, and one type of edges. For example, a friendship network where nodes represent persons and edges represent friendship is a unipartite network. Bipartite network is the network with two type of nodes and one type of edge where each edge connects nodes from different types (nodes of the same type cannot be connected). This network model can be found naturally when relations between two types of objects are represented as a network. For example, in authorship network, one research paper can be written by many authors while an author can write many papers. Edges also can be directed and undirected. Directed edge refers to one-sided relation, from a source node to a target node. Hyperlink networks are one of the most studied directed networks where each edge means there is a link from a source page to a target page.

Apart from types of nodes and edges, the arrangement of nodes and edges can also be used to differentiate the model of the networks. For example, random graphs are networks which edges are randomly link two nodes at some probability p. Random graphs used to be the main focus of researchers during the early stage of the study of network science. One of the most famous random graph model is proposed by P. Erdös and A. Rényi. (Erdös and Rényi (1959)). One important characteristic of random graph is that node degree follows normal distribution or most nodes have almost the same degree while there are some lesser degree and some higher degree nodes. However, it is shown that real-world networks are not random but have some form of structure (Girvan and Newman (2002)).

Scale-free networks, proposed by Barabási et al. (Barabási and Bonabeau

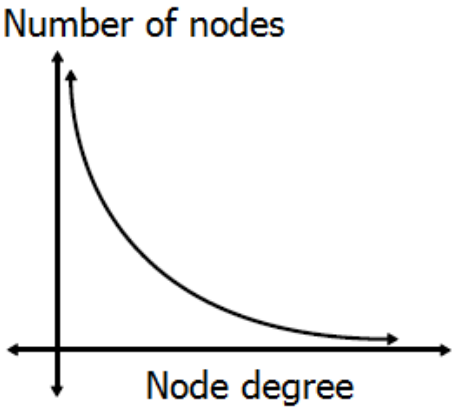


Figure 1.2: A power law graph between number of nodes and node degree.

(2003)), are network model that captures the structure of real-world networks. The main characteristic of scale-free networks is that the degree distributions of the network follows the power law, and the degree distribution on a log-log scale can be seen as a straight line. A power law is a statistic term used to explain the relationship between two quantities, where one quantity varies as a power of another. A power law can be explained by this formula:

$$y = \beta x^{-k} + \varepsilon,$$

where β is a constant value, k is a coefficient of scale-free, and ε represents a deviation of the formula. Normally, k is within a range between 2 and 3. An example of a power law graph between number of nodes and node degree is shown in Figure 1.2. In this case, there are many low degree nodes and only some high degree nodes. The high degree nodes are often called hubs. Hubs are normally connected with lower degree nodes which are then connected to even lower degree nodes and so on. It is also possible for hubs to connect with other hubs. Many real-world networks are considered scale-free networks. For example, social networks, citations networks, the World Wide Web, the internet infrastructures, or biological networks are considered as scale-free networks.

For each model of networks, different algorithms are needed. In this thesis, our aim is to create algorithms for community detection in scale-free networks. To reduce the complexity of the algorithms, we use undirected unipartite networks as

our network model. For further research, it is possible to extend our research for another network models such as directed networks or bipartite networks.

Metric of Community Detection

Another difficulty in community detection is how to measure the results of the algorithms. The simplest method is to test algorithms on networks that have ground-truth available. However, not all networks have ground-truth. To solve this problem, quality function is introduced. To measure the goodness of the result of community detection, a quality function calculates the score for each community. The summation of scores from all communities represents the quality of the community detection result. The most popular quality function is modularity by Newman and Girvan (Girvan and Newman (2002)). Modularity can be calculated by this formula:

$$Q = \sum_{i=1}^{c} e_{ii} - a_i^2,$$

where c is the number of communities. e_{ii} represents the sum of values of an adjacency matrix where starting and ending nodes are both in community i. e_{ii} can be calculated by this formula:

$$e_{ii} = \sum_{vw} \frac{A_{vw}}{2m} \delta(v, w),$$

where $\delta(v, w)$ returns 1 if node v and node w share the same community or 0 if not, m is the number of edges, and A_{vw} is the value from an adjacency matrix. a_i is the sum of cell values of an adjacency matrix where starting or ending nodes are in community i. a_i can be calculated by this formula:

$$a_i = \frac{k_i}{2m},$$

where k_i represents the number of edges that connect to community i. Since the introduction of modularity, it has quickly become one of the most popular benchmarks for community detection algorithms.

In case that ground-truth values are available, there are many possible ways to represent the correctness of the community detection results. For example, precision and recall is one of the popular methods. In this thesis, we use NMI or

normalized mutual information. NMI is used to determine the similarity between two partitioning results. In order to obtain the accuracy of the community detection results, we have to calculate the NMI between the ground-truth communities and the algorithm generated communities. The higher the scores are the closer the results are to the ground-truth. In many real-world networks, there is no absolute truth value. Thus using NMI is limited to the well-known networks which truth values are available or synthetic networks where truth values are already predetermined. There are many versions of NMI algorithms available. In this research, we use the version by Lancichinetti, Fortunato and Kertész (Lancichinetti et al. (2009)) because it is shown in their research that their NMI can perform correctly on disjointed networks, which are the target networks of this thesis.

1.1.2 Community Detection Algorithms

Detecting communities in complex networks is one of the most important topics in network science. In an early stage, graph partitioning methods are studied most. The most notable method is spectral clustering (von Luxburg (2007)). This method uses the eigenvector of the second smallest eigenvalue of the Laplacian matrix. Laplacian matrix is defined by this formula:

$$L = D - A,$$

where D is the diagonal matrix that each value represents node degree and A is an adjacency matrix. Networks can be divided into two partitions with very small cut size. The spectral clustering works by transforming current adjacency matrix of the graphs into easier to divided form using eigenvector transformation. After that, normal clustering method can be worked on the transformed matrix.

Another approach for community detection method is to focus on the edges that connect between communities. Communities can be isolated and obtained by removing edges that connect between communities. One of the most famous community detection algorithm proposed by Girvan and Newman (Newman and Girvan (2003)) is using this strategy. To identify the edges between communities, Girvan and Newman use betweenness centrality. Betweenness centrality is calculated based on the number of the shortest paths that go through that edge. Normally, betweenness value of edges that connects communities together is high. In

Girvan-Newman algorithm, betweenness of all edges are calculated. After that, the edge with the highest betweenness value is removed. Finally, the betweenness is calculated again followed by another edge removal. This process is repeated until no edge left. The partitioning result of this method is usually represented in dendogram. The problem of this method is the complexity and computation time. Because betweenness centrality of each edge has to be recalculated every time an edge is removed, this algorithm is too time consuming when the size of the network is large.

To cope with large networks, many new algorithms that have low computation complexity are proposed. One of these algorithms for large networks is proposed by Raghavan et al. in 2007 (Raghavan et al. (2007)). This algorithm is based on label propagation technique. In this method, each node starts with its own unique label. For each iteration, these labels are passed around to their neighbors in random sequences. At the end of each iteration, each node's label is updated to the majority label received from its neighbors and the next iteration begins. With this method, the number of unique labels is reduced in each iteration and finally the result is converged. Nodes that have the same label are in the same community. This method is very fast with the computation complexity of $O(m)$, where m is the number of edges.

Since the introduction of modularity by Newman and Girvan (Girvan and Newman (2002)), many researchers aim to create algorithms that yield the highest modularity as possible. However, it has been mathematical proven by Brandes et al. (Brandes et al. (2006)) that maximizing modularity is NP-complete problem, thus it is not possible to find the maximum modularity in polynomial time. Nevertheless, many algorithms can achieve fairly high modularity value within reasonable time. One of the most famous algorithm in this category is Louvain algorithm by Blondel et al. or known as fast unfolding algorithm (Blondel et al. (2008)). The modularity score of this method has been tested and it is shown to be excellent in comparison with other methods. Louvain algorithm starts by randomly choosing a node in a graph and try to group it with another node. After finding the best modularity gain pair, these two nodes are grouped together. This process is repeated until total modularity is at the maximum. Then, the algorithm starts randomly choosing groups instead of nodes to find the maximum modular-

ity gain. The algorithm is finished when the modularity is at the maximum and no groups can be merged to gain more modularity score. It is hard to calculate the exact time complexity of Louvain algorithm, but authors claimed that this algorithm has $O(n \log n)$ time complexity, where n is the number of nodes. It is shown that this algorithm can handle up to network that has 118 million nodes and 1 billion edges.

1.2 Scale-free Networks

During the early studies of complex networks, researchers found many characteristics that differentiate real-world networks from random networks. One of the earliest report on these characteristics is the research on networks of citations between research papers by de Solla Price in 1965 (de Solla Price (1965)). It is shown that the networks had heavy-tailed degree distribution following a power law. However, the term scale-free is not invented until 1999 by Barabási et al. (Barabási and Bonabeau (2003)). In their research, Barabási mapped the portion of the internet and found out that there are some nodes that have extremely high degree and act as hubs while degree distribution of the graph follows a power law. The term scale-free refers that at any scale of the graph, there are always hub nodes and node degree follows a power law. Thus these properties are valid at any scale.

1.2.1 Characteristics of Scale-free Networks

The most distinguished characteristic between random graphs and scale-free networks is degree distribution. In random graphs, each node has a percentage p to have an edge connect any other node in the graphs. Random graphs can be generated by using $G(n, p)$ model, where n refers to a number of nodes and p is the probability that an edge will be formed between two nodes. With this model, graph that has n nodes and m edges has the occurrence of

$$p^m(1-p)^{(\binom{n}{2})-m}.$$

This makes degree distribution of random graph follows random distribution according from this formula:

$$G(deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k},$$

where v refers to vertice v and k refers to degree of vertice v. This means most nodes have almost the same degree while some nodes have less than average degree and some nodes have more than average degree. On the other hand, degree distribution of scale-free networks is fat-tail distribution - a skewed distribution that has large standard deviations value - and follows a power law. When degree distribution of scale-free networks is displayed in log-log scale, it becomes straight line. Figure 1.3 shows the degree distribution of a co-authorship network of scientists working on network theory and experiment, compiled by Mark Newman in May 2006 (Newman (2006)). There are some nodes that have very high degree. These nodes are called hubs. The node degree distribution graph is plotted in log-log scale and the red line represents a power law.

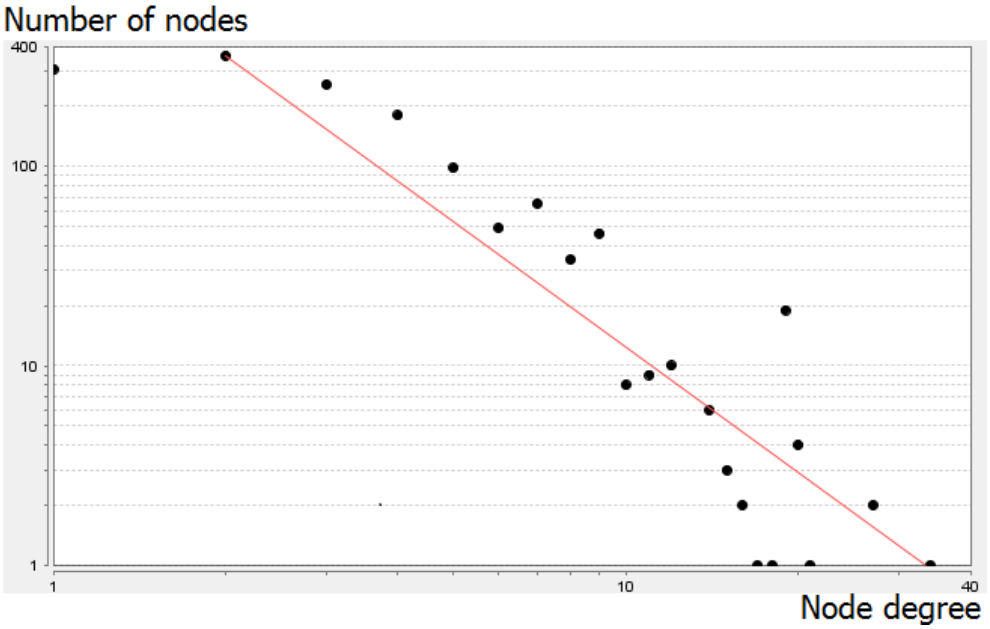


Figure 1.3: Degree distribution of a co-authorship network of scientists working on network theory and experiment (Newman (2006)).

These hubs are normally surrounded by smaller hubs and smaller hubs are surrounded by even smaller hubs. This causes networks to have scale-free effect. This structure also offers very high connectivity and leads to small-world effect. Scale-free networks have low average distance between two nodes. Facebook, one of the world largest social networks which is believed to be scale-free, reported that it has 3.74 degree of separation, very small average distance while retaining a very large clustering coefficient - a measure if nodes in a graph tend to cluster together. (Backstrom et al. (2012)).

Scale-free networks possess very high robustness. In case of a random failure of nodes, since most nodes have few degrees, even if the node fails the structure of the networks is not lost. Even in the case of hub nodes failure, due the high clustering coefficient, the smaller hubs can support and carrying the graph structure, preventing it from separation.

1.2.2 Previous Approaches for Community Detection on Scale-free Networks

There are also several algorithms that design specifically for scale-free networks. This is because scale-free networks have some properties that make them different from normal networks. Algorithms that are made for scale-free networks should outperform algorithms that are not made specifically for scale-free. Probably the first one of this kind is the work of Wu et al.(Wu et al. (2004)), titled “Mining Scale-free Networks using Geodesic Clustering”. Geodesic distance or shortest path is used in their method called geodesic clustering. Mean geodesic distance is calculated, and it is used for selecting a node to be in the same group. Abou-Rjeili and Karypis (Abou-Rjeili and Karypis (2006)) offered a multilevel algorithm for partitioning power law graphs. The key idea of their algorithm is instead of partitioning directly from the original graph, the approximations of original graph should be obtained first. The approximation can be done multiple times until the approximated graph is small enough to compute with any algorithms. Once partitioning is done, the last step is to derive partitioning results to the original graph. The work of Wu et al. and Abou-Rjeili and Karypis both suffer problems of graph data reduction, which affect the accuracy of the community detection result. Qian

et al.(Qian et al. (2007)) used hypergraph for clustering scale-free networks. In a normal graph, an edge can only connect two nodes together. However, in a hypergraph, an edge can connect to any number of nodes, which is called hyperedge. In their work, they created hyperedges from the density of a graph to find the communities. Hypergraph method is also suffered the problem of data reduction to some degree when transforming edges to hyperedges. Xu et al.(Xu et al. (2007)) introduced SCAN: A Structural Clustering Algorithm for Networks. Their algorithm is working by identifying each node into many classes, such as core, hub, and outlier. The partitioning is done according to the roles of each node class. The difficulties of SCAN is the definition of each node types which is dependent on two configurable parameters μ and ε. Altering these 2 values can have great effect on the community detection results, thus the appropriate values for each graphs are required beforehand. These values are tuned separately for each graph in their experiments.

1.3 Contribution of This Thesis

We contribute two works for community detection algorithms for scale-free networks: (1) node centrality and edge distance approach and (2) edge weight approach.

1.3.1 Community Detection in Scale-free Networks

Many community detection algorithms used today are invented without the consideration of scale-free networks and its properties. When using these algorithms on scale-free networks, it is normal that the results are not optimal. For example, suppose one of the most popular modularity optimization method, Louvain algorithm, is used for scale-free networks. There are some chances that communities that have scale-free properties (have hub nodes as the center the community) are broken into several communities (Jarukasemratana et al. (2013)). Many researches such as Lancichinetti and Fortunato (Lancichinetti and Fortunato (2009)) are confirming this fact.

Therefore, in this thesis, we aim to create community detection algorithms

that are specifically for scale-free networks. In the first contribution of our thesis, we focus on hub nodes and the structure of scale-free networks where hub nodes are surrounded by smaller degree nodes to perform community detection. In our second contribution, we focus on the importance of each edge in scale-free networks.

We also pay attention to the visualization of our approaches. To make it easy to understand, we use 3D visualization in our first contribution. This makes understanding concept of our algorithms easier. In our second contribution, we use edge thickness to display the importance of each edge thus the community structure can be seen easily even before applying actual community detection phase.

1.3.2 Node Centrality and Edge Distance Approach

For the first contribution of this thesis, we propose community detection algorithm for scale-free networks based on node centrality and edge distance. There are two main ideas for this algorithm. The first idea is that a community should be formed from the nodes that play important roles in a network. In the case of scale-free networks, hub nodes are the most important nodes and communities should be formed from hub nodes. By using centrality value, the hub nodes can be identified correctly. The second idea is that two nodes should be in the same community if they are closely connected together. Community detection is started from the highest centrality node to the lowest centrality node. Edge distance is used to determine if two nodes should be in the same community or not.

We compare our method with the Louvain algorithm, one of the most popular community detection algorithms (this algorithm is also known as fast unfolding). Our algorithm can achieve higher NMI values than Louvain algorithm in scale-free networks. Moreover, when our algorithm is displayed with force-directed layout as x-y plane and our algorithm's centrality values are used as z axis, the scale-free community shows a mountain shape structure.

1.3.3 Edge Weight Approach

However, during our experiments, we encountered a sub-type of scale-free networks. In these networks, even overall statistics indicated that these networks are

scale-free, some of their communities do not follows scale-free properties. Since the first contribution of our thesis is designed to find a community based on hub nodes, it fails to identify the communities that do not have hub nodes.

For the second contribution of this thesis, we propose community detection algorithm for scale-free networks based on edge weight. This is the direct continuation from our previous contribution. We use the phrase "mixed scale-free networks" for a sub-type of scale-free networks that some parts of the networks do not have scale-free properties. These networks can be found among scale-free networks, especially in social networks. The first main idea of this contribution is that each edge is not equal. Edges that connect to important nodes should be more important than edges that connect to nodes in the peripheral areas. The second main idea is that since some parts of networks do not have scale-free properties, we should not use scale-free approaches on them, but we should use other methods instead. With this concept mind, our second contribution is a two-step algorithm where the first step is aimed to extract communities with scale-free properties, while the second step is aimed to extract communities that do not have scale-free properties. Modularity optimization method is shown to have excellent results at community detection (Fortunato (2010)). Therefore, in the second step of this approach, we employ modularity optimization method. Moreover, we also improved modularity optimization method by using our edge weighting scheme.

We compare our results with several famous community detection algorithms which are Louvain algorithm (Blondel et al. (2008)), Girvan and Newman edge betweenness algorithm (Newman and Girvan (2003)), label propagation, and our first contribution approach. The experimental results shown that our method are superior than previous methods in scale-free networks, especially when the networks are mixed scale-free networks and perform equally in case of non scale-free networks.

1.4 Organization of This Thesis

This thesis consists of 5 chapters. In this section, we describe the organization of our thesis.

In chapter 2, we introduce our first contribution, community detection on

scale-free networks based on node centrality and edge distance. Section 2.1 is the introduction about scale-free networks and community detection algorithms on scale-free networks. Section 2.2 discusses about problems of community detection on scale-free networks. Section 2.3 explains our algorithm in detail. Section 2.4 shows the experimental results from our algorithm compared with previous algorithms. Section 2.5 discusses about the results obtained from the experiments. Section 2.6 is the conclusion of this approach.

Chapter 3 covers our work on community detection on scale-free networks based on edge weighting scheme. Section 3.1 introduces the main idea of the approach and how it can solve the problem encountered in chapter 2, also the mix scale-free networks are introduced. Section 3.2 introduces previous works related to our approach. Section 3.3 explains each step of the algorithm in detail. Section 3.4 shows the results of our algorithm compared with other previous algorithms. Section 3.5 is the discussion about our results and other alternative implementation method. Section 3.6 is the conclusion of this approach.

Chapter 4 discusses about the difference, advantages, and disadvantages between our first and second contributions.

Chapter 5 is the final chapter where we summarize this thesis and show the possible future works based on the contribution from this thesis.

Chapter 2

Node Centrality and Edge Distance Approach

2.1 Introduction

As discussed in section 1.1.1, modularity is a measurement on how network is partitioned into communities. High modularity score means there are dense connections between the nodes within the same community, but sparse connections among different communities. However, in some cases, high modularity score does not represent the correct communities. We found out that modularity optimization based algorithms are less accurate on scale-free networks which contain communities with scale-free properties.

In the first contribution, we present a community detection algorithm which uses node centrality and edge distance to determine communities. There are two main ideas for this algorithm. The first idea is that a community should be formed from the nodes that play important roles in a network. The second idea is that two nodes should be in the same community if they are closely connected together. By using centrality value, the hub nodes that are considered important in scale-free networks can be identified correctly. Edge distance is used to determine if two nodes should be in the same community or not.

We compare our method with the Louvain algorithm, one of the most popular community detection algorithms (this algorithm is also known as fast unfolding). Our algorithm can achieve higher NMI value than Louvain algorithm in scale-free

Figure 2.1: 3D visualization of Zachary's karate club.

networks. Moreover, when our algorithm is displayed with force-directed layout as x-y plane and our algorithm's centrality values are used as z axis, the scale-free community shows a mountain shape structure. Figure 2.1 is a 3D visualization of our algorithm on Zachary's karate club network (Zachary (1977)). x-y plane layout is generated by force atlas 2, z axis is centrality value of each node. Node 1 and node 34 which are hub nodes have very high centrality scores as they can be seen as the top two highest nodes in the graph. Two communities can be observed by two mountain shape structures where nodes at the peak of the mountain are the hub nodes.

This community detection algorithm is similar to pouring colored water onto the peak of each mountain, where each color represents a community. For example, we pour red water onto the top of the highest mountain peak (node 1 in Figure 2.1). The water will flow to its closest node (node 2). The second highest node is node 34. Since it is the peak of the mountain, we pour new color on it, in this case blue. As our algorithm continues, the water flows down from the top of the mountain until it reaches the base. Finally, the whole mountains are covered with color and our algorithm is finished. Similar to a characteristic of water that it will not flow up, our algorithm is following this characteristic as well. This means the node that has higher centrality value will not be assigned community based on lower centrality nodes.

2.2 Problems of Community Detection on Scale-free Networks

The problem of modularity optimization based algorithms on scale-free networks occurs when low degree nodes form a strongly connected group inside a community. Even if they are connected by hub nodes, if the number of edges in this strongly connected group is high enough (and provides high enough e_{ii} value), the algorithm will declare this sub-community as a new community. This behavior can break a community with a scale-free property into many sub-communities. For example, in Figure 2.8(b), a network got separated into 4 communities by Louvain method, but the ground-truth shows that there are only 2 communities. Normally, the communitiy of scale-free network is based on hub nodes. However, based on the formula of modularity all nodes and edges are treated equally, thus ignoring the fact that some nodes are more important than others. In our opinion, that is not the accurate way. In scale-free networks, nodes with high degree or hubs should be considered more important than these with lower degree. Moreover, edges that originated from hubs should be considered more important than those that connect the lower degree nodes. Figure 2.2 shows the result of Louvain algorithm compared with the ground truth communities. Community on the left graph (a) is the ground truth while community on the right (b) is the result of Louvain algorithm. It can be seen that communities got separated into many small communities by Louvain algorithm.

2.3 Our Algorithm

There are two main ideas in our algorithm. The first idea is that nodes that are close together should be in the same community. The second idea is that the community should be formed from the important nodes. To determine the distance among nodes, we use edge distance as the measurement. And to determine the importance of the nodes, we use node centrality. Once edge distance and node centrality are calculated, the community detection phase is employed. In this phase, all nodes are first treated as no community. Then, starting from the highest centrality node, the community is assigned to it and its closest neighbor. After

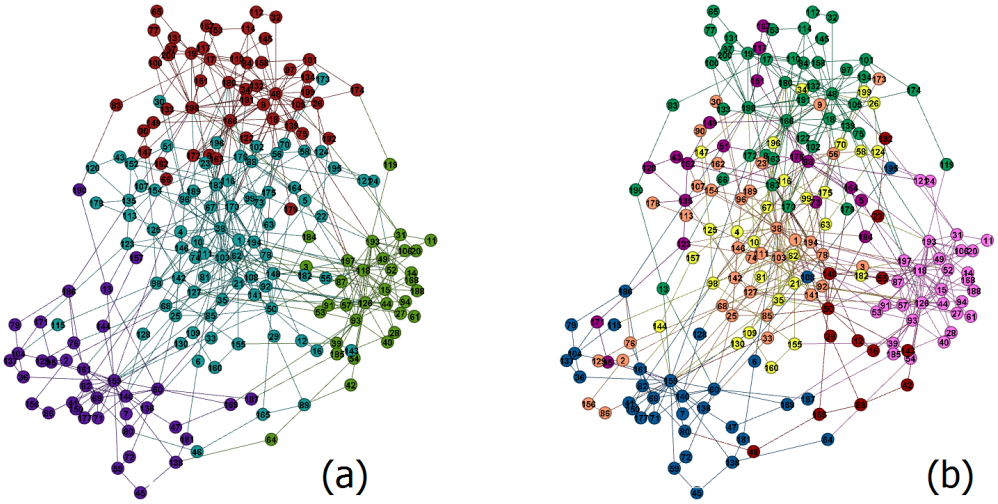


Figure 2.2: Ground truth value (a) compares with community detection result from Louvain algorithm (b).

that, the next highest centrality node and its closest neighbor are picked to be assigned the community. The algorithm is finished when all nodes have been assigned to any of the community. This section is divided into four subsections: centrality value, edge distance, community detection, and configurable variables of the algorithm.

2.3.1 Centrality Value

Node centrality is one of the most intuitive methods to identify the important nodes in a network. The early researches about node centrality can be dated since 1950s by Bavelas and Leavitt (Estrada (2011)). In general, node centrality represents its ability to communicate with other nodes, or its closeness to other nodes. The most basic form of centrality is the degree centrality. The main idea of this centrality is that the node is more central than others if it connects to more nodes. In short, degree centrality value is equal to node degree. The main flaw of degree centrality is that it only accounts the neighbor nodes. To correctly represent the centrality of nodes, the nodes that farther away should also be included into the calculation. Leo Katz (Katz (1953)) introduced his version of centrality measure-

ment which is later known as Katz centrality. To calculate the degree centrality of node i, we count the nodes that are one hop away from node i (thus this value is equal to the degree of node i). However, in Katz centrality, nodes that are n hops away from node i are also counted. The nodes that are n hops away can be found by the n-th power of the adjacency matrix. However, nodes that take fewer hops to reach should provide more centrality score than the nodes that are farther away. Therefore, the attenuation factor α is introduced. Katz centrality can be expressed as the equation below:

$$K_i = [(\sum_{k=1}^{\infty} \alpha^{k-1} A^k) I]_i,$$

where K_i is the Katz centrality of node i, A^k represents the power k-th of the adjacency matrix, α is the attenuation factor and I is the column vector where all values are 1.

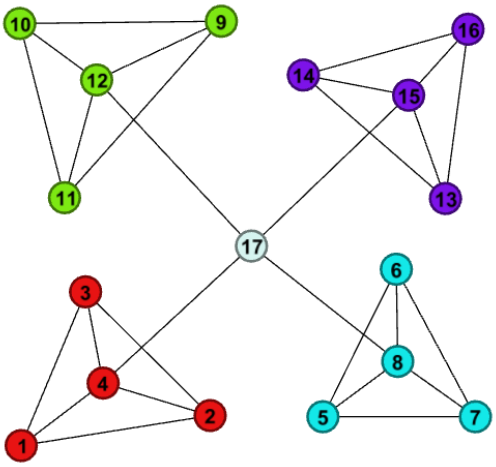


Figure 2.3: Network that Katz centrality cannot identify hub nodes.

Consider normal scale-free networks; Katz centrality can identify hub nodes correctly since hubs normally have high degree. Even in the case where hubs do not have the highest degree itself, nodes from the second and third length hops are usually enough to increase the centrality score of hub nodes to a noticeable level. However, the problem is the node that connects at least two hubs together. This node will have a high centrality score from connecting to hubs and sometimes it may seem more important than the hub nodes. The example of this problem is

shown in Figure 2.3. In this small network, node 17 has the highest Katz centrality score. Without using subgraph centrality, the whole graph will merge into one community, while modularity-based algorithms will give 4 communities. To solve this problem, we have decided to add another centrality value to complement the Katz centrality.

Subgraph centrality was introduced in 2005 by Estrada and Velázquez (Estrada and Rodríguez-Velázquez (2005)). Similar to Katz centrality, subgraph centrality also uses the hops length increment method, starting from 0 hop to n hops. The main difference is that only the hops that complete the closed walk are counted. Closed walk of 0 hop (0 length) is the node itself. Closed walk of 1 hop (length 1) is not available (since it cannot hop back to the starting node to create a closed walk). Number of closed walks of length 2 is equal to its degree, the node hops to its neighbor nodes and hops back to itself to create closed walks. Closed walk of length three is a triangle shape closed walk. The length three closed walk from node i involving node j and k would form a triangle shape; node $i \rightarrow$ node $j \rightarrow$ node $k \rightarrow$ node i. Subgraph centrality score of length l on node i can be calculated by this formula:

$$f_i(A) = (\sum_{l=0}^{\infty} c_l A^l)_{ii},$$

where f_i is subgraph centrality of node i, l is the length of closed walks, c_l is the coefficient of length l (similar to attenuation factor in Katz centrality), and A^l represents the l-th power of the adjacency matrix. Since the first meaningful value of start at hop length 3, we will use c_3 equal to 1. After applying subgraph centrality into graph in Figure 2.3, node 12, 4, 8, and 15 become more important than node 17. The number of communities now becomes 4.

In scale-free networks, length three subgraph centrality score can clearly differentiate hub nodes from non hub nodes. This can be seen in Figure 2.8(a) where node 1, a hub node, has 18 length three closed walks, the highest number in this network. This can solve the problem of Katz centrality on nodes that connect at least two hubs together. This is because those nodes do not participate in many length three closed walks. Since Katz centrality and Subgraph centrality contain many overlapping parts, we choose to combine only the length three subgraph centrality with Katz centrality to create our centrality score. The formula of the

centrality score is shown below:

$$K_i = [(\sum_{k=1}^{\infty} \alpha^{k-1} A^k) I]_i + (A^3)_{ii}.$$

Another point for discussion is how many hops should be used. In both Katz centrality and subgraph centrality, the number of hops is set to infinite. Katz centrality's attenuation factor and subgraph centrality's coefficient are selected so that the infinity converges. This is reasonable because centrality value should be calculated from the involvement of every node in a network. However, in our case, we only want the centrality value to identify the hubs. Therefore, it is not necessary to involve the nodes that are too far away. We observed from our experiment that length of three hops is the most appropriated value because it provides the best compromise between accuracy and computation time. However, it is possible to use higher hop number to increase accuracy at the cost of the computation time.

2.3.2 Edge Distance

Edge distance is calculated to determine the distance between two nodes. Edge distance can be calculated from distance matrix. Let A be the distance matrix, the shortest distance or shortest path from node i to node j is equal to A_{ij}. However, communication between nodes does not always use the shortest path. For example, if we want to send a large amount of data from node i to node j, we would use all the paths that are available to maximize the data flow. Edge distance is the value that used to measure this maximum data flow. Edge distance can also be regarded as a measure of how fast it will take to spread information from node i to node j. Our distance is developed based on the Katz centrality algorithm. Edge distance matrix can be calculated by using the formula shown below:

$$D_{ij} = c_1 d_{ij1} + c_2 d_{ij2} + ... + c_l d_{ijl}.$$

D_{ij} is the edge distance between node i and node j. d_{ijl} is the total number of paths that has length of l that connect node i and node j. c_l is the coefficient value. l is the number of hops or lengths. The higher the edge distance score between node i and node j is, the closer they are.

Similar to Katz centrality, this value can identify the closeness between nodes very well, but it creates a problem in scale-free networks. The problem occurs on the nodes that are not the part of the community but are connecting at least two hubs together. The distance score on those nodes can be very high, resulting in a merging of the two scale-free communities. The solution to this problem is similar to centrality case, which is to add the length three closed walk of subgraph centrality. Let node i connects to node j and node k. If node j and node k are connected, the length three closed walk is formed. Then, the distance score from node i to node j and node i to node k will be increased.

Probably the most beneficial factor and the reason we use this distance is that it is a byproduct of the centrality value. In fact, the sum of the distance value from node i to other nodes in graph is equal to our centrality value:

$$K_i = \sum_{j=0}^{\infty} D_{ij}.$$

Therefore, the calculation needs to be done only once to get both centrality score and distance score. The l value of distance is also similar to the l value used in centrality calculation.

There are other distance values available such as shortest path and resistance distance (Klein and Randić (1993)). In our algorithm, we need to calculate the distance among adjacency nodes. In shortest path, the distance among adjacent nodes are all equal to 1. Therefore, shortest path is not suitable for our algorithm because it cannot differentiate the importance among adjacency nodes. On the other hand, resistance distance and our edge distance can differentiate the distance among adjacency nodes correctly. Figure 2.4 shows the inverse version of resistance distance values and our edge distance values of each edge. It can be seen that the values are almost identical. This is because our edge distance and resistance distance are both based on the information spreading concept. The reason that we use our own distance instead of using resistance distance (even the values are almost identical) is because our edge distance is a byproduct of centrality values calculation, which means we do not need to calculate distance values separately. Our distance is better in term of resources and calculation time when compared with resistance distance.

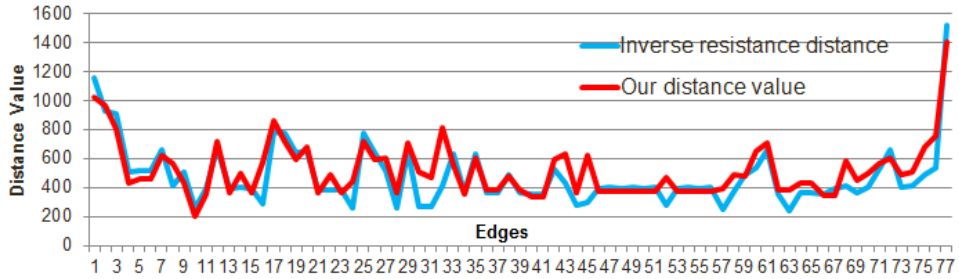


Figure 2.4: Our distance values and inverse resistance distance values of each edge on the same network.

2.3.3 Community Detection

At this stage, we already obtained centrality value of each node and the edge distance matrix of the network. To start the community detection phase, we treat all nodes as no community. Then we pick the node that has the highest centrality value as our main node and its closest node as the secondary node. The closest node refers to the node that has the highest distance score from the main node. This node can be easily looked up from our distance matrix. Then we have to check if our main node or secondary node already has a community or not. There are 4 possibilities; 1) if the main node already has a community but the secondary node does not, the secondary node will join the same community as the main node; 2) if the main node does not have community but the secondary node already has the community, the centrality score will be used to determine the outcome of this situation. If the main node's centrality score is higher than the secondary node's, a new community will be created and assigned to the main mode. But if the main node's centrality score is lower than that of the secondary node's, the main node will be assigned to the same community as the secondary node; 3) if both nodes do not have any community assigned, a new community will be created and assigned to both nodes; 4) if both nodes already have community, nothing will be done. This process starts from the highest centrality node to the lowest centrality node. A constraint relaxing parameter called laxing value is also used in this algorithm when assigning communities. This parameter is explained in the next section. The pseudo code of this algorithm is shown in Algorithm 1.

```
list_of_node L = sort(nodes, centrality, hightolow);
for node i in L do
    node j = find_closest_node(i, laxing_value);
    switch do
        case has_community(i) and !has_community(j)
            assign_community(j,get_community(i));
        end
        case !has_community(i) and has_community(j)
            if (get_centrality(i)¡get_centrality(i)) then
                assign_community(i,get_community(j));
            else
                c = new community();
                assign_community(i,c);
            end
        end
        case !has_community(i) and !has_community(j)
            c = new community();
            assign_community(i,c);
            assign_community(j,c);
        end
        case has_community(i) and has_community(j)
            //both nodes have groups, do nothing
        end
    endsw
end
```

Algorithm 1: Community detection algorithm of node centrality approach.

The characteristic of this algorithm is similar to a flow of water. When we visualize a network in 3D by using force layout algorithm (in this case, we used force atlas 2 (Jacomy (2011))) and assign the z axis value as our centrality value. The scale-free community becomes a mountain shape structure with the hub node as its peak. Figure 2.1 is the 3D representation of our algorithm on the Zachary's karate club network. It can be seen that there are two scale-free communities in this network (two mountains). The peaks of those mountains are Mr. Hi and John, the leader of each group. The communities are spreading from these 2 nodes. The result is that we have 2 communities that have Mr. Hi and John as the center of each community. Also, similar to the water that it will not flow up, the condition 2 in our pseudo code is representing this characteristic.

2.3.4 Configurable Parameters

There are several configurable parameters in this algorithm, which are the coefficient value or the attenuation factor α, the depth of the algorithm or the maximum number of hops l, and the constraint relaxing parameter called laxing value. The effect of each of these values will be explained.

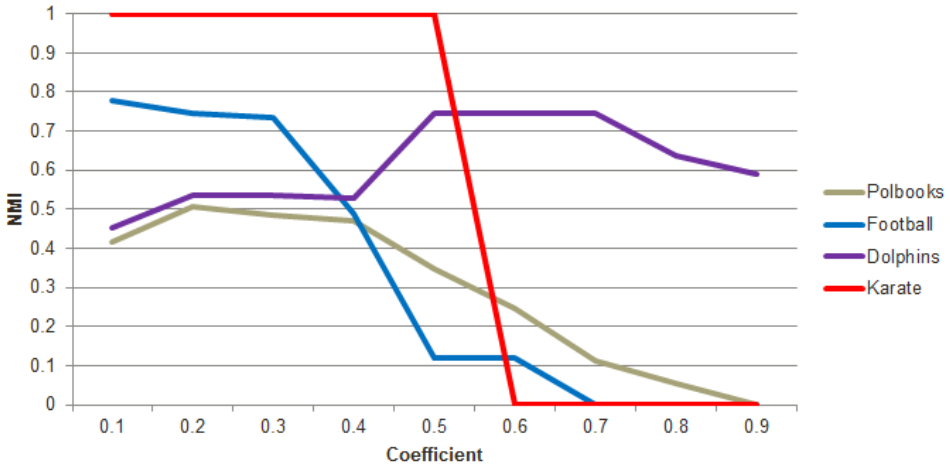


Figure 2.5: NMI scores from various coefficient values on 4 real-world datasets. Y axis is NMI score (higher the better). X axis is coefficient value.

In Katz centrality, coefficient or attenuation value must not be set to the eigenvalue of the network. This is because it is required that the matrix to converge when number of hops is set to infinite. However, since we choose not to use the infinite number of hops, this restriction does not apply. Coefficient value is the value that acts as a score reduction on nodes that are farther away. This value is the same on both edge distance and node centrality. For example, suppose coefficient value is set to 0.1. Let node i connects to node j, and node i has a degree of 1. Node j connects to 3 nodes or has a degree of 3. The final score of node i is 1 * 0.1^0 + 3 * 0.1^1 = 1.3. Different coefficient value can change the outcome of the final score. For example, comparing two nodes, let node i connects to 2 nodes that are one hop away and 2 nodes that are two hops away, while node j connects to 1 nodes that are one hop away and 7 nodes that are two hops away. If coefficient is 0.1, node i score is 2.2 (2 * 0.1^0 + 2 * 0.1^1) while node j score is 1.7 (1 * 0.1^0 + 7 * 0.1^1). However, if coefficient is 0.4, the scores will become 2.8 (2 * 0.4^0 + 2 * 0.4^1) and 3.8 (1 * 0.4^0 + 7 * 0.4^1). This can drastically change the outcome of the algorithm.

The first problem of coefficient value is that if the value is set too high, non hub node that connects to many hub nodes can have higher centrality value than hub nodes. The result is that all communities will be merged into one large community. On the other hand, since our centrality value is consisted of 2 parts, if the coefficient value is too low, subgraph centrality part will become too dominant. This can result in breaking of a community when there are low degree nodes that have high subgraph centrality in graph. These 2 effects of too low and too high coefficient values can be seen in Figure 2.5, where NMI values drop when coefficient approaches 1 in all networks or when coefficient approaches 0 in polbooks and dolphins networks. To fix the coefficient value within this range of not too low and not too high, we defines our coefficient value as:

$$\alpha = 1/d,$$

where α is coefficient value, and d is an average degree of the graph. If this coefficient value is used, the condition for communities to merge into one large community is true when:

$$d^2 + d > 2\Delta_H,$$

where Δ_H is subgraph centrality of hub nodes. This formula is valid only when there is a node that connects to many hub nodes at the same time, which is quite rare on scale-free networks. The formula is derived from our centrality formula which is shown in centrality value section, on the assumption that the worst case occurs (none hub node A connects to many hub nodes). In this case, the centrality of node A is:

$$C_A = d + d.\alpha.(C_H - \Delta_H) + \Delta_A,$$

where C_A is centrality of node A, α is the Katz coefficient, Δ_A is subgraph centrality of node A, and Δ_H is subgraph centrality of hub node H. Using $\alpha = 1/d$ and assume that Δ_A has maximum value (worst case scenario), the formula becomes:

$$C_A = C_H - \Delta_H + (d^2 + d)/2.$$

In the case of all communities to be merged into one large community, C_A must be higher than C_H, which results in the condition mentioned earlier.

Using similar calculation from the case of graph merging, The following condition must be true for the case of community breaking:

$$(d^2 + 3d)/2 > 2d_H + \Delta_H,$$

where d_H is the degree of hub nodes. This case is also very unlikely to happen because scale-free networks have power law degree distribution where degree of hub nodes is high when compare with average degree. Moreover, subgraph centrality of hub nodes are also very high. These two values can be seen in Figure 2.6.

To show the values of these parameters in actual networks, we use 5 synthetic networks based on preferential attachment model - model that is used for scale-free networks - and 5 real-world Facebook networks from ego-Facebook datasets (McAuley and Leskovec (2012)). The values are shown in Figure 2.6. It can be seen that in all networks, none of the condition is satisfied for causing communities to merge or to break. However, in some cases such as tree networks where Δ_H is 0, the merge or the break can still occur.

The number of hops or l also plays an important role in this algorithm. It is noted that l value must be more than three in order to use the length three closed walk as a part of the calculation. In the case when l is equal to one, centrality will

Network	1	2	3	4	5	FB1	FB384	FB1648	FB1912	FB3437
Size (nodes)	200	200	200	400	500	333	229	765	747	534
d	4.98	5.18	4.86	4.61	5.66	15.12	28.5	4.5	80.38	18.06
$(d^2 + d)/2$	14.8902	21.1862	19.0998	17.5411	24.5078	136.987	448.875	16.875	3351.04	190.172
Δ_H	25	33	22	31	59	970	1713	136	16573	1129
$(d^2 + 3d)/2$	19.8702	21.1862	19.0998	17.5411	24.5078	136.987	448.875	16.875	3351.04	190.172
$2d_H + \Delta_H$	77	85	80	133	177	1124	1911	408	17159	1333

Figure 2.6: Values from 5 synthetic networks and 5 ego-Facebook networks.

become degree centrality and edge distance will be 0 (not connected) or 1 (connected) only. The number of hops greatly affects the complexity of the algorithm. This will be explained in discussion section.

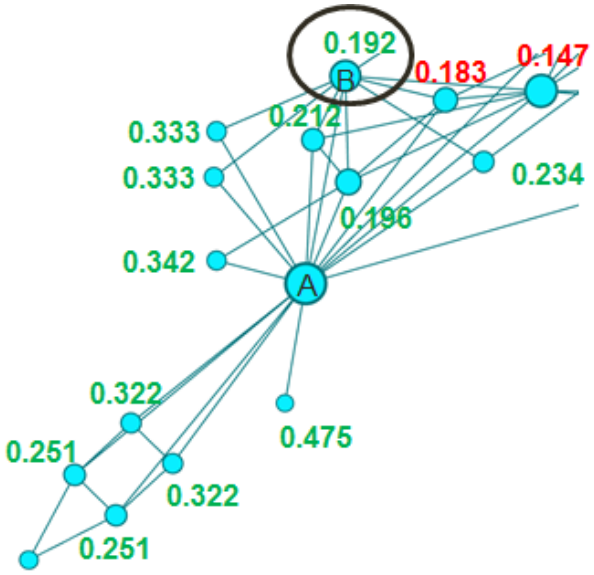


Figure 2.7: Node B is joining node A with ratio value of 0.192. Nodes that have higher ratio than B that connect to node A also join A at the same time, while nodes that have ratio lower than ratio of B are not joining.

The last configurable parameter is laxing value. Laxing value is used in community detection phase. The role of laxing value is to make a group of nodes that

have similar distance scores to act as one node. In this case, we use the ratio of distance from current node and the sum of all distance values from this node to other nodes in graph. For example, let node B is joining node A and node B has distance to A equals to 5 and sum of distance values from B to other node in graph is 10, B has ratio value of 0.5. Node C also connects to node A while C has distance value equals to 3 and sum of distance values from C to other node in graph is 5, C has ratio value of 0.6. In this case, if B is joining A with ratio of 0.5, C will also join A at the same time because C has higher ratio then B. The example of the use of laxing value is shown in Figure 2.7.

Another configurable aspect that should be mentioned is that the method for calculating centrality value and node distance value can be substituted. There are many centrality calculation algorithms such as eigenvector centrality (Bonacich (2007)), closeness centrality (Borgatti (2005)), betweenness centrality (Freeman (1977)) and more. All of these centrality values can be exchanged with our centrality value to create another community detection results. Similar to centrality value, our distance value algorithm can be replaced with other algorithms as well, such as the inverse value of resistance distance (Klein and Randić (1993)).

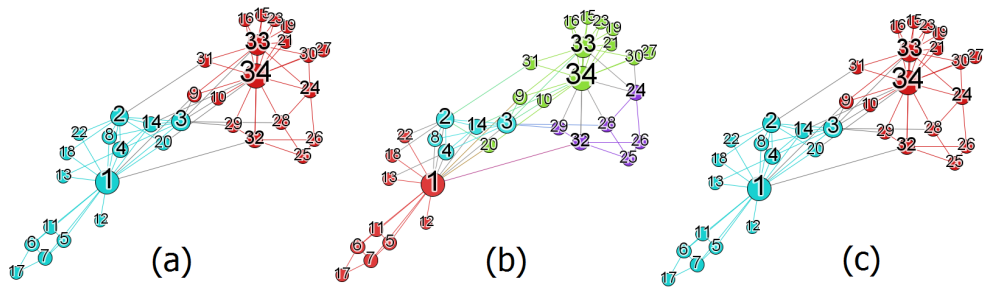


Figure 2.8: Zachary's karate club, (a) truth value (b) Louvain algorithm - NMI = 0.337 , (c) our algorithm - NMI = 1.0

2.4 Experiments

One of the indexes for measure the goodness of community detection is normalized mutual information or NMI. The NMI is used to compare two sets of par-

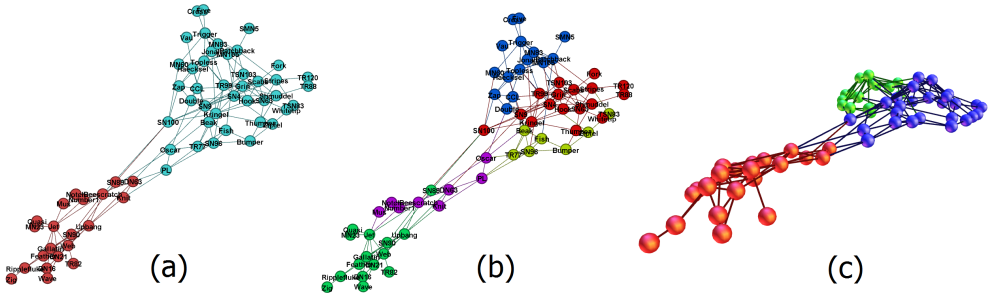
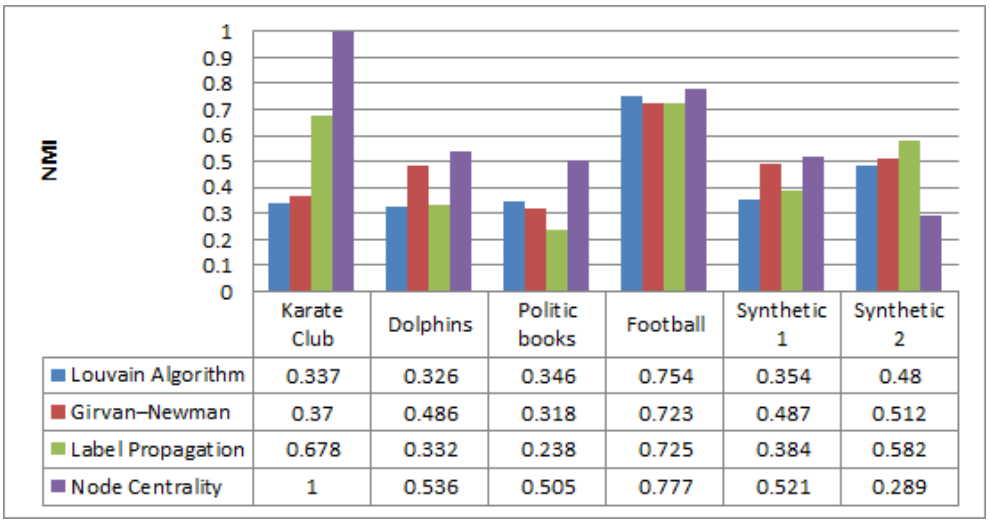


Figure 2.9: Dolphins network, (a) truth value, (b)Louvain - NMI = 0.326, (c) our algorithm (in 3D visualization) - NMI = 0.536



	Karate Club	Dolphins	Politic books	Football	Synthetic 1	Synthetic 2
Louvain Algorithm	0.337	0.326	0.346	0.754	0.354	0.48
Girvan–Newman	0.37	0.486	0.318	0.723	0.487	0.512
Label Propagation	0.678	0.332	0.238	0.725	0.384	0.582
Node Centrality	1	0.536	0.505	0.777	0.521	0.289

Figure 2.10: NMI scores on 4 real-world datasets and 2 synthetic datasets.

titioning results. The value is high when two results are similar. NMI is normally used when the ground truth of the network (the correctly partitioned set) is available. This severely limits the use of NMI since many networks do not have ground truth, such as friends networks or communication networks. However, NMI can be used on synthetic networks or some well-known networks that ground truth communities are available. Comparing the results of the partitioning with ground truth partition by using NMI is probably the most accurate method to grade the result of the partitioning. Although many NMI algorithms are available, we use generalized normalized mutual information developed by Lancichinetti et

al. (Lancichinetti et al. (2009)) in this research. There are some famous real-world networks where ground truth values are available and often used in testing the validity of the community detection algorithms (Fortunato (2010)). Networks that are used in this paper are Zachary's karate club (Zachary (1977)), Doubtful Sound dolphins network (Lusseau et al. (2003)), American college football network (Girvan and Newman (2002)), and political books network (Krebs (2004)). In addition to the known famous networks, synthetic networks are other options. In order to generate scale-free networks, we use the method describe in "Benchmark Graphs for Testing Community Detection Algorithms" by Lancichinetti et al. (Lancichinetti et al. (2008)). Since we aim only to partition unweighed and undirected graphs, the basic version of Lancichinetti's work is used. To test with more real-world datasets, we used ego-Facebook datasets provided by Stanford Large Network Dataset Collection (McAuley and Leskovec (2012)). However, the ground truth values for ego-Facebook datasets is not available. To validate the accuracy of our method, we compared our results with modularity optimization based algorithm called Louvain algorithm or as known as fast unfolding. It is shown in their research that it surpassed other modularity optimization methods. Other methods that are used as based line methods are Girvan-Newman algorithm (Newman and Girvan (2003)), and label propagation algorithm (Xie and Szymanski (2011)).

The truth value of Zachary's karate club network is extracted from the node attributes table presented in "An Information Flow Model for Conflict and Fission in Small Groups" by Wayne W. Zachary (Zachary (1977)). There are five categories of factions: Mr. Hi - Strong, Mr. Hi - Weak, John - Strong, John - Weak, and None. We then assign John - Strong and John - Weak to community 1 and Mr. Hi - Strong and Mr. Hi - Weak to community 2. Nodes without faction are neglected. The degree distribution of this network is shown as almost straight line in log-log scale. Power-law degree distribution can be observed. As shown in Figure 2.8(b), a network got separated into 4 communities by Louvain method while our algorithm provided 2 communities similar to the truth value. In result, the NMI value of our algorithm is significantly higher. Also by looking at Figure 2.1, it is shown that our algorithm can detect 2 mountain shape structures. Each mountain represents a community with scale-free property.

The bottlenose dolphins of Doubtful Sound network is another network that is frequently used by community detection researchers. Unlike Zachary's karate club, the known truth value is not available. However, in consensus, this dolphins network is believed to be divided into 2 communities as shown in Figure 2.9(a). With Louvain algorithm, the network was split into 5 communities (NMI = 0.326) while our algorithm provided 3 communities (NMI = 0.536), and the truth value contains 2 communities. Our algorithm also produced higher score in NMI. In 3D visualization, our algorithm detected 3 mountain shape structures, suggesting 3 scale-free communities.

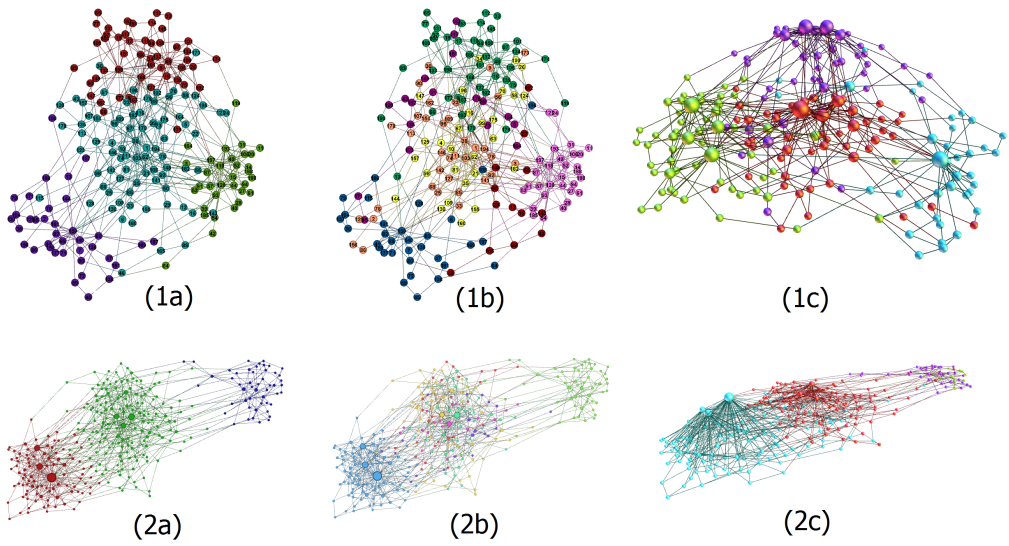


Figure 2.11: Synthetic network 1 and 2, (1a) truth value , (1b) Louvain algorithm - NMI = 0.354 , (1c) our algorithm - NMI = 0.521 , (2a) truth value , (2b) Louvain algorithm - NMI = 0.48 , (2c) our algorithm - NMI = 0.289

We used Lancichinetti's work to create benchmark graphs (Lancichinetti et al. (2008)). The graph follows the power law on both node's degree distribution and community size. The program can be downloaded directly from Fortunato's website (Fortunato (2011)). Two synthetic networks are shown in Figure 2.11. The sizes of the networks are 200 and 300 nodes respectively. In network 1, our algorithm can identify the number of communities correctly which is 4, and provided

higher NMI value. However, in network 2, our algorithm identified the left and middle communities correctly but did not perform well on the right community. The reason for this is because the right community does not possess a scale-free attribute. Therefore, our algorithm failed to identify it as one community. This can be seen in the 3D visualization (Figure 2.11(2c)) as a flat area on the far right. This problem will be discussed further in the discussion section.

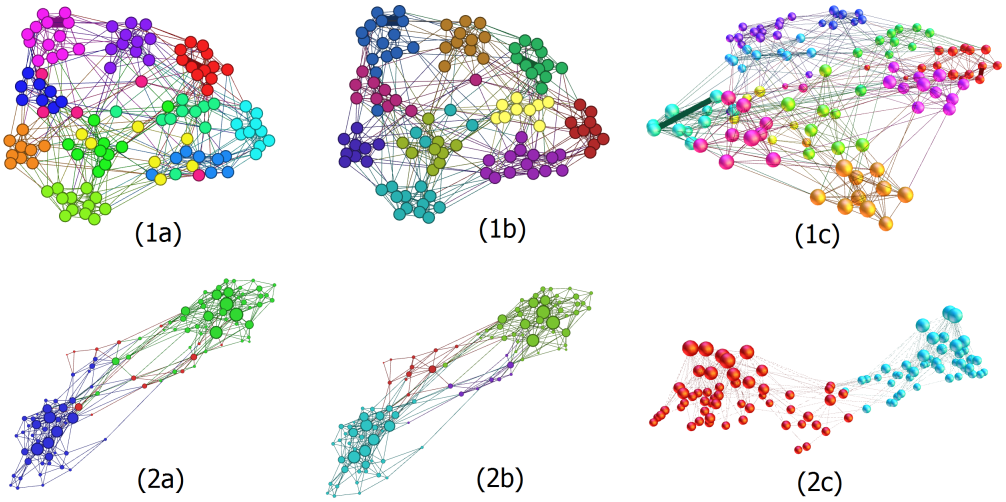


Figure 2.12: American college football network (1) and US political books network (2), (1a) truth value , (1b)Louvain - NMI = 0.754 , (1c) our algorithm - NMI = 0.777 , (2a) truth value , (2b) Louvain - NMI = 0.346 , (2c) our algorithm - NMI = 0.505

Scale-free networks are only a portion of total networks. For networks that have poisson or normal distribution such as random networks or small-world networks, modularity optimization is probably the best method to detect communities. Modularity score also has a great advantage over NMI because modularity score does not require a ground truth value. Since our algorithm is aimed for scale-free community detection, it does not perform well on non scale-free networks. However, in order for our algorithm to be used on non scale-free networks, we have to modify our configurable parameters so that our algorithm can perform community detection effectively. Two non scale-free datasets are used in this sec-

tion, American college football network (Girvan and Newman (2002)), and US political books network (Krebs (2004)). By using our auto-generated coefficient value and laxing value, the result is that our algorithm can perform well on non scale-free networks.

To show the result of our algorithm on real scale-free social networks, we used a portion of ego-Facebook datasets provided by Stanford Large Network Dataset Collection (McAuley and Leskovec (2012)), namely network number 0, 384, 1648, and 3473. The result can be seen in Figure 2.13. Results of network 0 and network 384 showed one huge scale-free community and a few smaller communities. Results of network 1648 showed two large scale-free communities with a few smaller communities. Results of network 3437 showed one large scale-free community and several medium/small sizes communities. It should be noted that there are contradictions between our algorithm and Louvain algorithm in every graphs, especially in the largest community. Louvain algorithm tends to separate the largest community into several smaller communities, while our algorithm detects only one scale-free community. In 3D visualization, the mountain shape structures can be seen in all of the Facebook graphs.

The final experiment is the comparison between our algorithm and another scale-free algorithm. During the earlier years, SCAN by Xu et al.(Xu et al. (2007)) are regarded as the most accurate algorithm for scale-free networks. In 2010, Khorasgani et al. (Khorasgani et al. (2013)) proposed top leader algorithm. Top leader algorithm works in a similar way of our algorithm, which means communities are defined based on hub nodes. Top leader achieve higher accuracy than SCAN but the number of communities must be defined beforehand. On the other hand, our algorithm can automatically define the number of communities. In this experiment, we tested both algorithm on 10 synthetic scale-free networks based on preferential attachment model. The comparison between NMI results of our algorithm and top leader are shown in Figure 2.14.

The average of our algorithm NMI is 0.461 while average NMI of top leader is 0.402. This means our algorithm is even more accurate than top leader. Moreover, our algorithm does not require the number of communities beforehand.

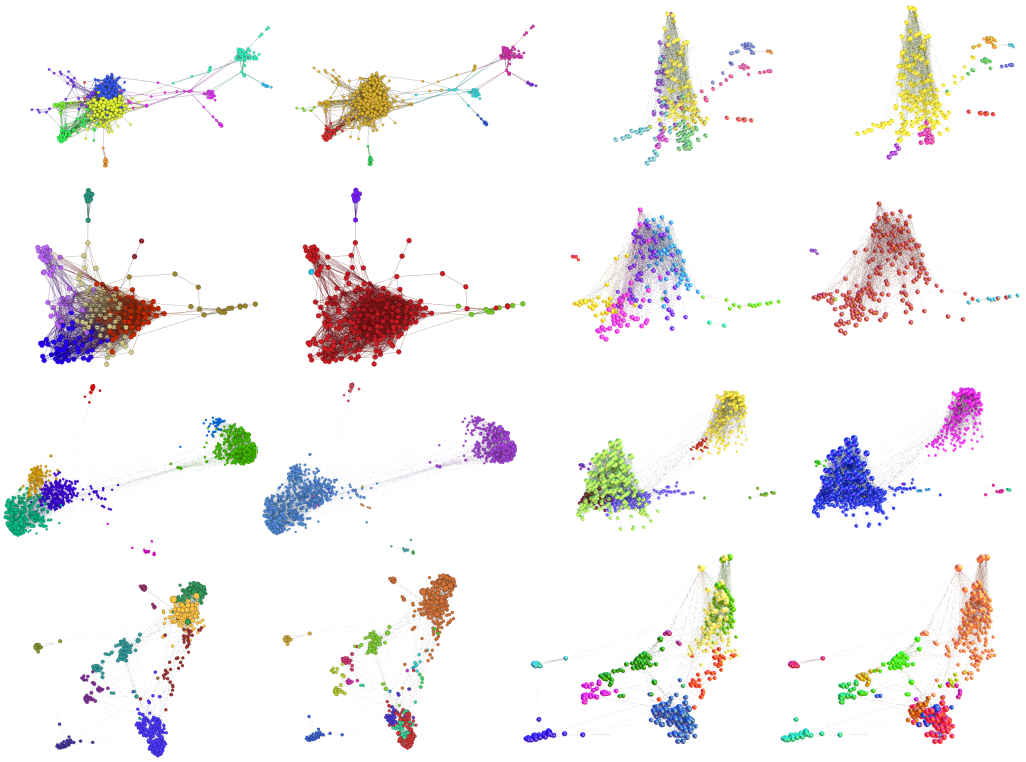


Figure 2.13: Figures of facebook networks number 0, 384, 1648, and 3437. Row - from top to bottom: 0, 384, 1648, 3437. Column - Louvain algorithm, our algorithm, Louvain algorithm in 3D, our algorithm in 3D

2.5 Discussion

It can be observed from our experiment that modularity optimization based algorithms can sometimes break a community with scale-free attribute into many communities, such as, the communities on Zachary's karate club (Figure 2.8(b)), the communities on dolphins network (Figure 2.9(b)), and communities on synthetic networks (Figure 2.11(1b) and 2.11(2b)). Considering the definition of modularity, the breakdown of these scale-free communities can obtain higher modularity score. This phenomenon occurs when low degree nodes form a tightly connected group within a scale-free community. Not every scale-free networks have this phenomenon. The reason behind this occurrence is that modularity based algorithms

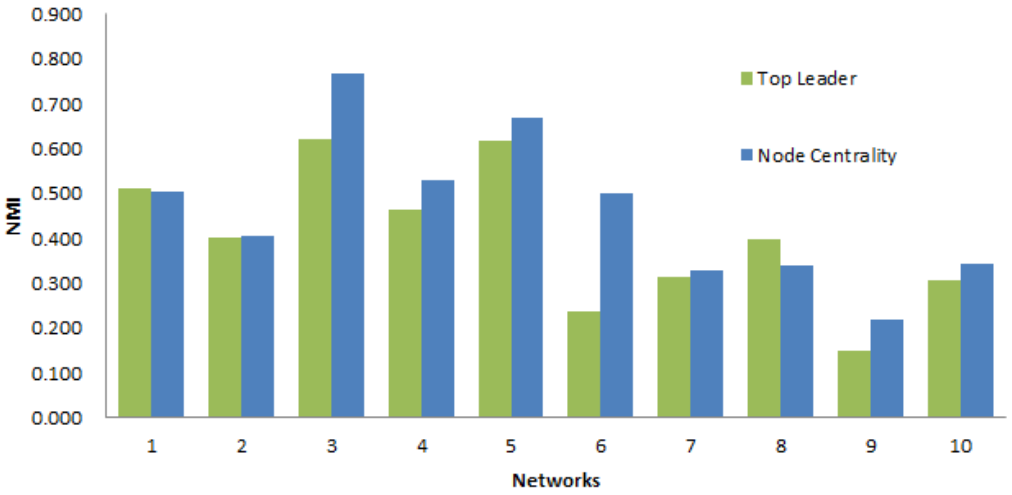


Figure 2.14: NMI results of our algorithm compared with top leader.

considered all edges and nodes as equal. This characteristic makes modularity optimization based methods less suitable for scale-free networks. In our opinion, hub nodes and edges that generated from hub nodes should be treated with more important than other nodes and edges. Our algorithm is implemented based on this idea and the result is presented in this chapter.

The flaw of our algorithm is shown in the synthetic network 2 (Figure 2.11(2c)), especially on the right most community, which does not contain a scale-free property. We have shown that our algorithm can work on non scale-free networks in previous section by using auto-generated parameters. However, what happens when there are scale-free communities and non scale-free communities in the same network? With our current implementation, parameters are automatically generated based on network's properties. It is possible to manually set the parameters so that the result is more accurate. However, this solution is still not satisfactory and it is solved in our second contribution of this thesis.

Another discussion that should be mentioned is the complexity of the algorithm. Originally, our algorithm's complexity is at $O(n^3)$. This is the complexity of the most time consuming part in our algorithm, the Katz centrality. However, in 2001, Foster and Muth (Foster et al. (2001)) introduced "A Faster Katz Status Score Algorithm" which can reduced the complexity to $O(n + m)$, by using list instead of matrix. After applying the idea from Foster and Muth research, the

complexity of our algorithm became $O(mk^{l-1})$ where m is the number of edges, k is the average degree, and l is the number of hops. Since we use l equals to 3, the effective complexity is $O(mk^2)$. The current largest dataset which was tested with this algorithm contains 317,080 nodes and 1,049,866 edges. The total computation time is 26 minutes on a laptop computer using Intel i7 with 8 GB of RAM. The result is shown in Figure 2.15.

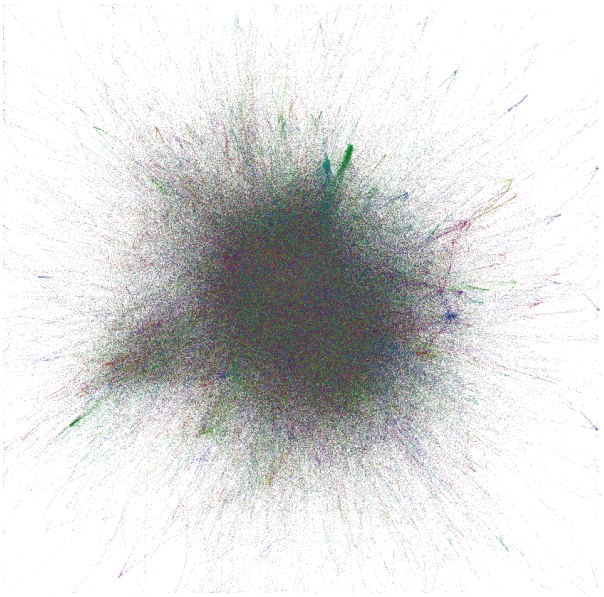


Figure 2.15: Result of our algorithm on large scale network with more than 300,000 nodes and 1,000,000 edges. Total computation time is 26 minutes.

2.6 Conclusions

In this contribution, we have introduced a community detection algorithm based on node centrality and edge distance value. The algorithm works well on scale-free networks, and can score higher NMI scores when compared with modularity optimization based algorithms. The first main idea of this algorithm is that the community should be formed with the nodes that play important roles in a network; the hub nodes in this case. Centrality value is used to find the hubs of

each community and these hub nodes are used as the community starting points. Edge distance is used to determine the closest connected nodes. These values are used in our second main idea of our algorithm; nodes which are closely connected together should be in the same community. Moreover, if we use the centrality value calculated in our algorithm as the z axis and visualize the graph in 3D, the community with scale-free property can be seen as a mountain shape structure.

Chapter 3

Edge Weight Approach

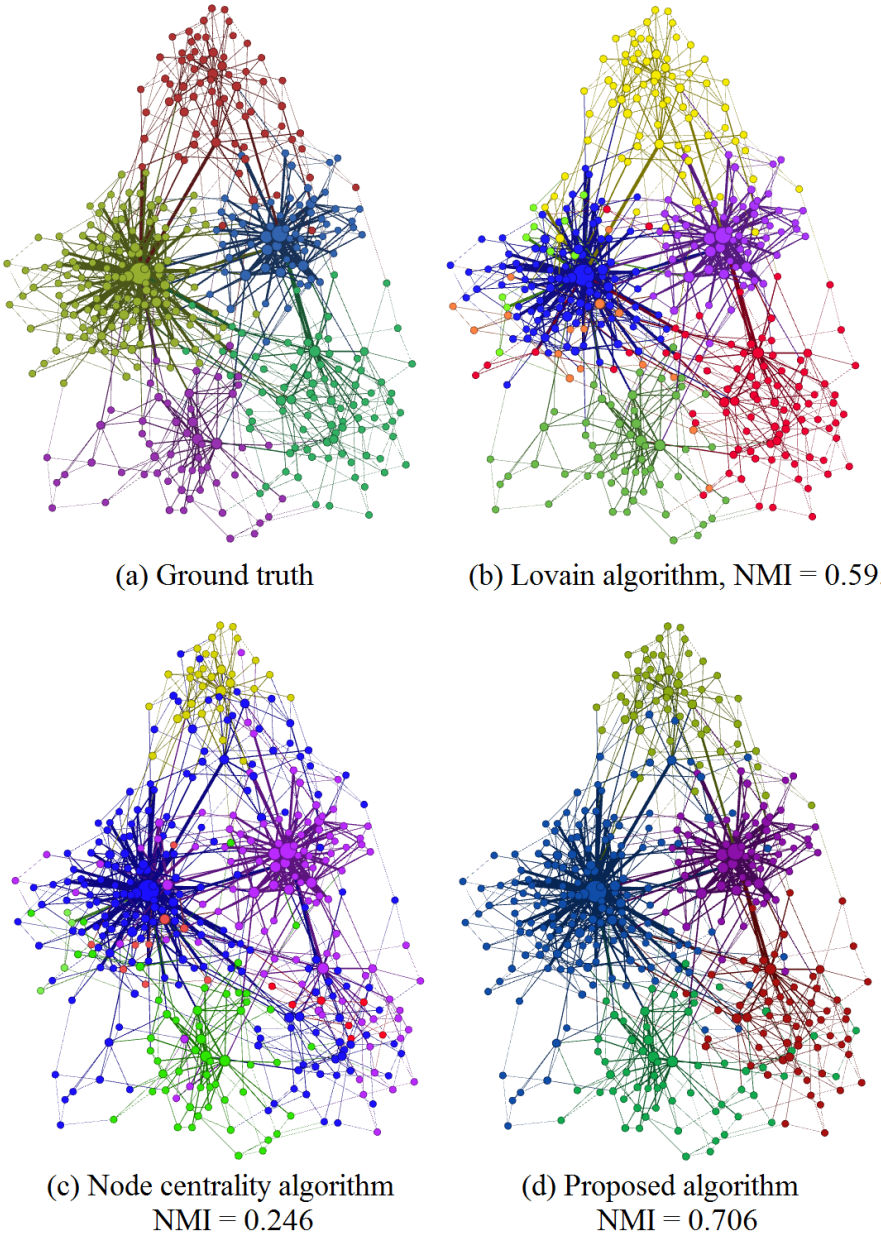
3.1 Introduction

During our own experiments on scale-free networks, we encountered another subtype of scale-free networks that does not fully follow the characteristics of scale-free, which we call "mixed scale-free network". In this type of scale-free networks, some communities have hub nodes and node degree follows a power law, while some communities do not have hub nodes and node degree follows normal distribution. Mathematically, scale-free graphs can be generated by using Barabási Albert model (Barabási and Bonabeau (2003)), which can be described with this formula:

$$p_i = \frac{k_i}{\sum_j k_j},$$

where p_i is the probability that a new node is connected to node i, k_i is a degree of node i, and j represents other nodes in the graphs. This means new nodes tend to connect to existing nodes that have high degree. However, mixed scale-free network does not directly follow Barabási Albert model. From the observation on real-world datasets, some closely connected groups of nodes are based on scale-free model, such as Barabási Albert model, while some groups follow other models, such as Watts Strogatz model - a random graph generation model that produces graphs with small-world properties (Watts and Strogatz (1998)). Therefore, it is possible to generate mixed scale-free networks by using network models such as Watts Strogatz model along with scale-free based model such as Barabási

Albert model while maintaining the overall characteristics of scale-free networks.



(a) Ground truth

(b) Lovain algorithm, NMI = 0.595

(c) Node centrality algorithm NMI = 0.246

(d) Proposed algorithm NMI = 0.706

Figure 3.1: Ground truth value and community detection results of synthetic network_m2.

For mixed scale-free networks, modularity optimization based methods will

have difficulties because some communities contain scale-free property. At the same time, scale-free based methods will have difficulties because some communities have nodes whose degrees follow a normal distribution. In this contribution, we propose a community detection algorithm that can work on networks that contain both types of communities at the same time.

Figure 3.1 shows the result of our propose algorithm compared with Louvain algorithm and algorithm from our first contribution, node centrality algorithm (Jarukasemratana et al. (2013)). Since node size is correspondent to node degree, the bigger the node size, the higher the node degree. The higher the NMI means the result is more similar to the ground truth (better). Figure 3.1(a) is the ground truth community of the network. In Figure 3.1(b), Louvain algorithm failed to extract the largest community (blue color in Figure 3.1(b)). The node degree of this community follows a power law. The result contains 3 communities instead of 1 (blue orange and green). On the other hand, In Figure 3.1(c), node centrality algorithm failed to extract the bottom right community. This is because the algorithm is based on finding hub nodes in order to start the community, but there is no hub node in bottom right community (communities whose node degree follows normal distribution do not have hub nodes). What our algorithm do is to take the advantages of both approaches and combines it into one result. The result is that our algorithm receives the highest NMI (Figure 3.1(d)) which means the result is the most similar to the ground truth.

Mixed scale-free networks are often observed in social networks. Figure 3.2 is one of the networks from Ego Facebook datasets (McAuley and Leskovec (2012)) that contain both communities with normal distribution and power law distribution. Red community's node degree follows normal distribution, while blue community's node degree follows power law distribution. Red community on the left does not contain any hub nodes and most nodes have almost the same size (almost the same degree), while community on the right (blue) contains very large nodes and very small nodes similar to scale-free networks. The results of comparing our proposed algorithm with Louvain algorithm and node centrality algorithm can be seen in Figure 3.15.

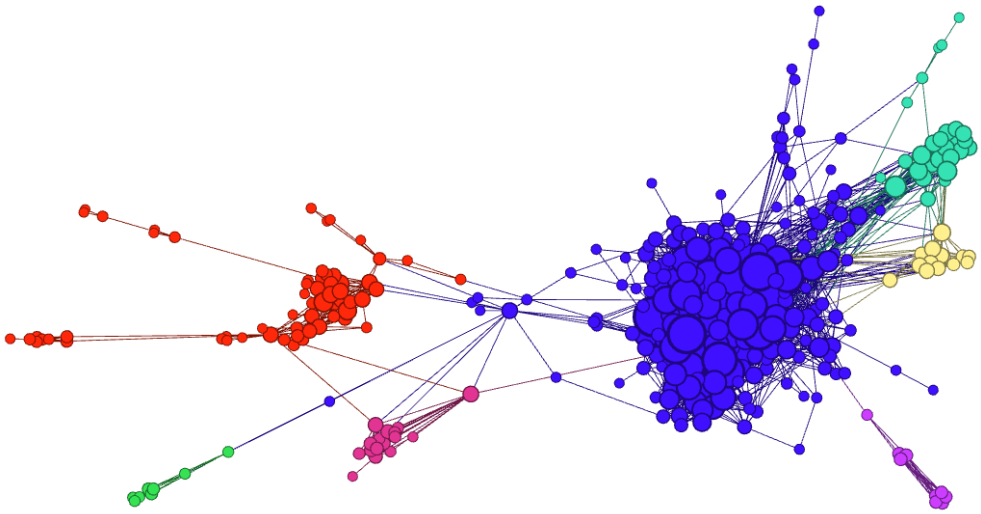


Figure 3.2: Facebook0 network from Ego Facebook datasets (McAuley and Leskovec (2012)).

3.2 Previous Works Related to Our Approach

Our algorithm consists of many steps and contains many ideas from previous works in this field. This section is dedicated to those previous works. First of all, in the first step of our algorithm, we have to calculate the weight of each edge. This concept is similar to the famous work by Girvan and Newman's edge betweenness algorithm (Newman and Girvan (2003)). The difference from our algorithm is that our weight is not directly coincide with the betweenness, but is a mixture of centrality and local clustering coefficient. Moreover, Girvan-Newman is a top down approach, which treats all nodes as one community and then divides them up into several communities. Our algorithm, instead, is a bottom up approach. Each node has its own community at first, and then combines them into bigger communities.

There are many community detection methods that aim for scale-free networks. One of them is the first contribution of this thesis, node centrality algorithm. The method starts with the calculation of node centrality and edge distance. Node centrality is used to find hub nodes and edge distance is used for determining the communities. The weighting scheme of edge weight approach is the improved

version of this edge distance. The largest difference between our first contribution and the second contribution is that our second contribution's algorithm is focused on edges instead of nodes. By using edge weight, scale-free communities can be extracted in the earlier step while communities with normal distribution can be extracted in the latter step. Node centrality algorithm also fails to extract communities that do not have hub nodes, which normally appear in mixed scale-free networks.

The latter part of our algorithm is modularity optimization based method. Since our method is a bottom up approach, it is similar to the second iteration of Louvain algorithm (Blondel et al. (2008)). In Louvain algorithm, each node is randomly selected and paired with other node in order to create the highest modularity gain pair. Then, the current network is transformed into a new network where nodes in the network are the communities of the current network. After that, the first and the second steps are repeated until the maximum modularity is gained. Therefore, in our algorithm, the starting network for Louvain algorithm is a processed network created by our scale-free part. Another difference from normal Louvain algorithm is that edges in our algorithm have weights. This greatly affects the outcome of the community detection.

3.3 Our Algorithm

We believe that, each edge is not equally important in scale-free networks. Therefore, the main idea of our algorithm is to weight each edge according to its importance, then community detection is performed base on the weighted network. The summary of our algorithm is explained here. First of all, weight of each edge is calculated by using centrality method with attenuation factor along with local clustering coefficient. Once we have all edges weighted, community detection phase is performed. The first step is to assign each node its own community. Then, for each node, we create a list that stores all communities that connect with this node. The next step is to calculate the total weight from each community that connects to this node. If the highest total weight community is not this node's current community, the node will change its community. This step is repeated several times until all nodes are balanced or the threshold is met. Then, we apply Louvain

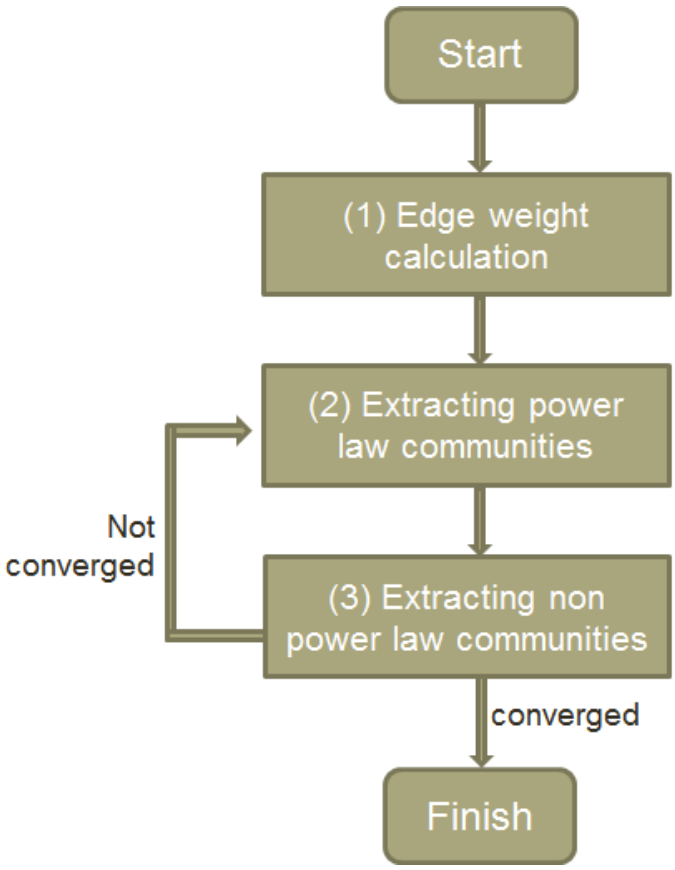


Figure 3.3: Flowchart of Edge Weight Algorithm.

algorithm with weighted edges to our result from step one, which is to treat each community as a node and try merging it until the total modularity of the graph is maximum. Finally, step one and two are repeated iteratively until all nodes are balanced. The flow chart of our algorithm is shown in Figure 3.3.

3.3.1 Edge Weights Calculation Phase

The purpose of calculating edge weight is to determine whether the nodes that are linked by this edge should be in the same community or not. If the weight is high, it means that nodes that are connected by this edge should be in the same community. Therefore, the weight of edges that connect nodes inside same community should be high. On the other hand, if the weight is low, these two nodes should not be in the same community. Edges that connect nodes from different communities

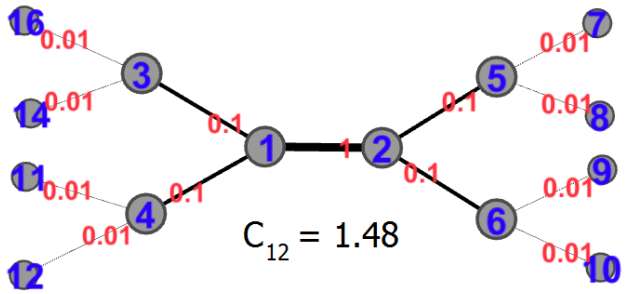


Figure 3.4: The example of edge weight calculation.

should have low weight. To explain with metaphor, edges in our algorithm are similar to rubber bands that connecting nodes together. Edge weight represents the thickness of the rubber bands. The higher the weight, the more pulling power between connected nodes. To calculate this weight, we apply the node centrality method base on Katz centrality (Katz (1953)), but we use it on edges instead. The formula for edge centrality is shown as follow:

$$C_{ij} = \sum_{l=0}^{n} \alpha^l E_l,$$

where C_{ij} is the centrality of edge that connects node i and node j, n is the maximum depth, E_l is the number of edges that are located l hops away from edge i. If the network is weighted, E_l is equal to the sum of all weight that are located l hops away from edge i, respectively. Figure 3.4 shows an example on how to calculate this value. Our target is the weight of an edge that connecting node 1 and 2. Let attenuation factor α is set to 0.1. An edge itself weight 1.0. There are 4 edges that are located 1 hop away from this edge, resulting in α times 4 = 0.4. There are 8 edges that are located 2 hops away, resulting in α^2 times 8 = 0.08. Therefore, using l = 2, this edge's weight is 1+0.4+0.08 = 1.48. If the edge connects nodes with high degree, the centrality of that edge is high. This centrality value can separate edges that connect to important nodes (such as hub nodes in scale-free networks) from edges that connect to less important nodes.

However, in some cases, the centrality value alone is not enough. Edges that connected two hub nodes from different communities normally have very high centrality, which results in a merging of two communities. The example of this

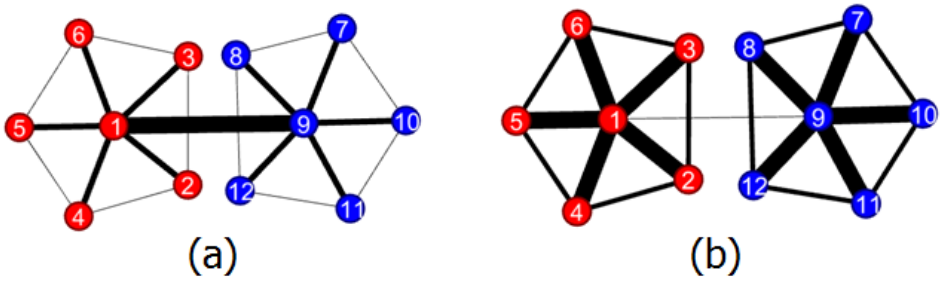


Figure 3.5: Edge weight without(a) and with(b) local clustering coefficient.

scenario is shown in Figure 3.5(a), where edge that connects between two hub nodes (node 1 and node 9) has more weight than other edges. To prevent this from happening, we added local clustering coefficient into edge weight calculation. Local clustering coefficient can differentiate edges inside the same community from edges that link communities together. Edges that bridge two communities together normally have low local clustering coefficient. Figure 3.5(b) shows the edge weight result after including local clustering coefficient into the formula. Local clustering coefficient can be counted by the number of triplets (3 nodes and 3 edges form a triangle). For each triplet that the edge participated in the weight is increased by 1 (in case of unweighted networks). For weighted networks, the weight is increased by its original weight. The number of triplet can be calculated by using the power of 3 of the adjacency matrix.

The last step is to normalize the value. For each edge weight, we divide it by the average value of all edge weight. This process is needed because in some extreme cases, the difference in node degree is very high. For example, in a very dense area of the graph, the weight of each edge is extremely high. When community detection phase is performed, communities from sparser area will merged into communities from denser area. The final step is to add our normalized weight to current edge weight. In case of unweighted graphs, each edge weight is one. The final formula is shown as follows:

$$W_i = w_i + (C_i + T_i)/(\overline{C} + \overline{T}),$$

where W_i is the weight of an edge i. w_i is the original weight of an edge i, C_i is the centrality of an edge i, T_i is the number of triplets that an edge i participated in,

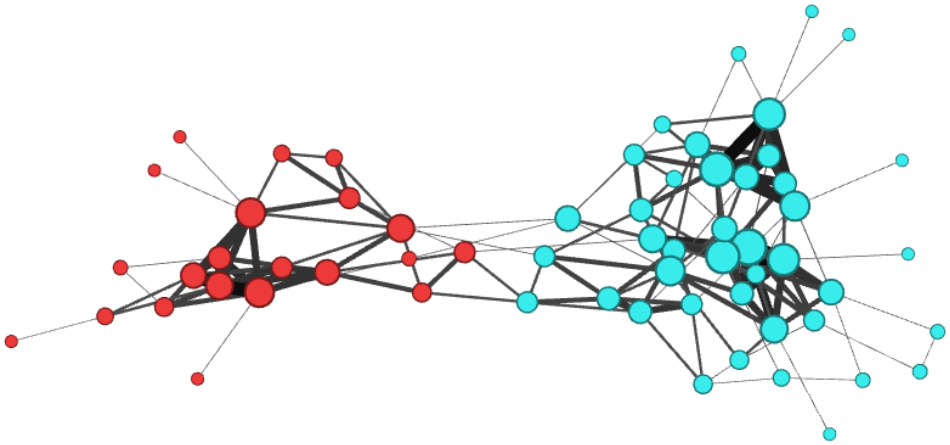


Figure 3.6: Doubtful Sound dolphins network (Lusseau et al. (2003)) after edge weights calculation.

$\overline{C}$ is the average value of edges centrality, and $\overline{T}$ is the average number of triplets that edges participated in. If the network is weighted, w_i is equal to the original weight of an edge i. In case of unweighted edges, w_i is equal to 1. From our experiments on unweighted networks, edge weight normally ranges from 1 to 4 in the case of scale-free networks. The higher values are from edges that connect to hub nodes. Also, the number of edges that has weight less than 2 is significantly larger than the number of edges that weight more than 2. This is according to the structure of scale-free networks that node degree follows power law distribution. On the other hand, in networks that node degree follows normal distribution, most weight values are around 2. An example of edge weight from real-world dataset can be seen in Figure 3.6. Node's size refers to node's degree, the bigger the higher. Edge's thickness refers to its weight, the thicker edges weight more than the thinner edges. It can be noticed that edges that connected to high degree nodes (hubs) have considerably higher weights. Moreover, communities are divided where edge weight value are low (thin edges).

3.3.2 Community Detection Phase

There are 2 steps in community detection phase. The first step is aimed to extract communities with power law distribution (scale-free), while the second step is aimed for communities with normal distribution.

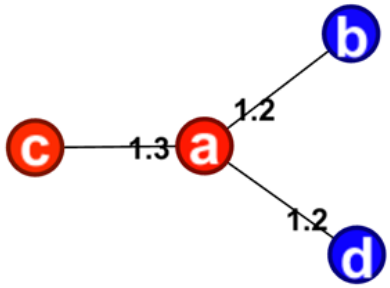


Figure 3.7: Node a is going to change from red community to blue community because the total weight from blue community is 2.4, which is more than total weight of 1.3 from red community.

Step 1

Since we assume that nodes that are connected by high weight edge should be in the same community, edges with higher weight should be processes first. This step starts with sorting all edges from the highest weight to the lowest weight and assigns each node a unique community. For each node, we create a list that stores all communities that connect with this node. The next step is to calculate the total edge weight from each community that connects to this node. In the first iteration, since each node has its own community, the highest community will always come from the highest weight edge. If the highest weight community is not this node's current community, the node will change community to the community with the highest total weight instead (this node got pulled to the highest weight community). This process is repeated iteratively until all nodes are balanced or until the threshold is met. The example scenario is shown in Figure 3.7. The pseudo code of this step is shown in Algorithm 2.

The communities that follow scale-free are extracted almost completely in

this step. This is because in scale-free community, edges that connected with hub nodes tend to have very high weight. Edges around hub nodes will be processed first. If this process is observed step by step, it can be noticed that the community is extracted starting from hub nodes and then expanding to the peripheral area of each community. The order of execution is the main reason that this method is more suitable for scale-free than other methods such as Louvain algorithm or label propagation. It is possible that non scale-free methods to break large scale-free community into smaller communities because those methods randomly choose the node for starting the community. It is possible that hub nodes are selected later in the random order and many small communities are already formed, which resulted in the breaking of large scale-free community into several smaller communities. However, with our ordering method, this problem can never occur.

On the other hand, in the communities with normal distribution, each edge has almost the same weight. Therefore, nodes in communities with normal distribution are randomly grouped up into small sub-communities without any significant meaning during this step.

```
node.sort(HighestEdgeWeight);
foreach nodes do
    foreach node.adjacency do
        map.add(adjacency.community,adjacency.weight);
    end
    highestWeight = FindHighestWeight(map);
    highestCommunity = map.get(highestWeight);
    if (node.community == highestCommunity) then
        # no need to do anything.
    else
        node.community = highestCommunity;
    end
end
```

Algorithm 2: Community Detection Phase - Step 1

The example result of step 1 is shown in Figure 3.8. This network contains 3 communities. The right most (red) and middle (blue) communities are scale-free

communities, while the left most community (pink) follows normal distribution. After step 1, right most and middle communities are extracted (red and blue color). However, left most community got separated into 2 small sub-communities (black and pink). Also there are some nodes at peripheral area of each community that are still given wrong communities.

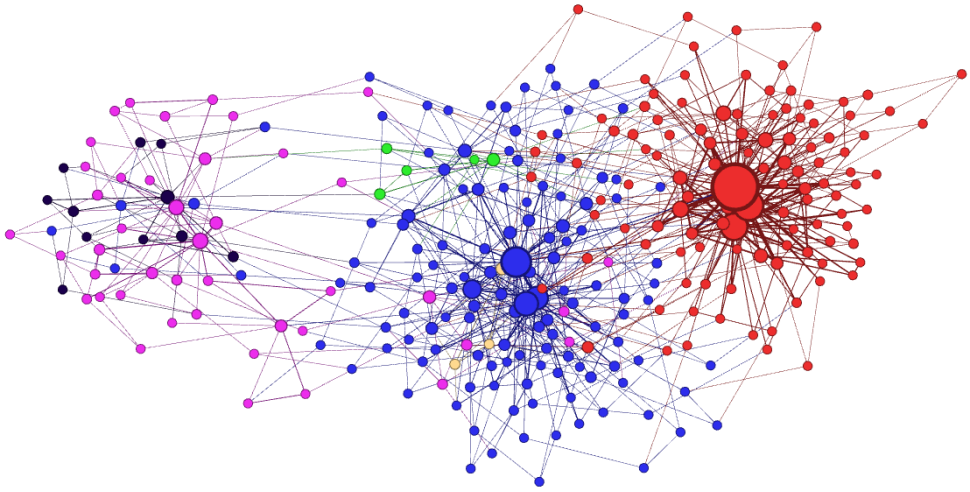


Figure 3.8: An image of synthetic network_m1 network after step 1 of community detection phase.

Step 2

The goal of this step is to extract communities whose node degree follows normal distribution. It has been proved that modularity optimization based method is excellent at this task. Therefore, in this step, we applied the modified version of one of the most popular modularity optimization algorithm, Louvain algorithm (Blondel et al. (2008)). There are 2 steps in Louvain algorithm, the first step is to randomly select nodes and try to pair it with other nodes in order to obtain sub-communities of the highest modularity. When all nodes are processed, this step is done. The second step is to aggregate communities generated in the first step into a node. With this method, the network obtained in this step is much smaller from the original network. This is the end of the first iteration. Step 1 and step 2 are

repeated iteratively until the maximum modularity is gained.

In our case, our scale-free communities' detection phase is equal to the first step of Louvain algorithm. In step 2 of our algorithm, we start with the aggregation of our current network into a new network where each community is represented as a node. After that, Louvain algorithm is applied until the maximum modularity is obtained. Another difference from using normal Louvain algorithm is that we have edge weight. Using our weighting scheme results in a better NMI than non-weighted version. This is further explained in discussion section.

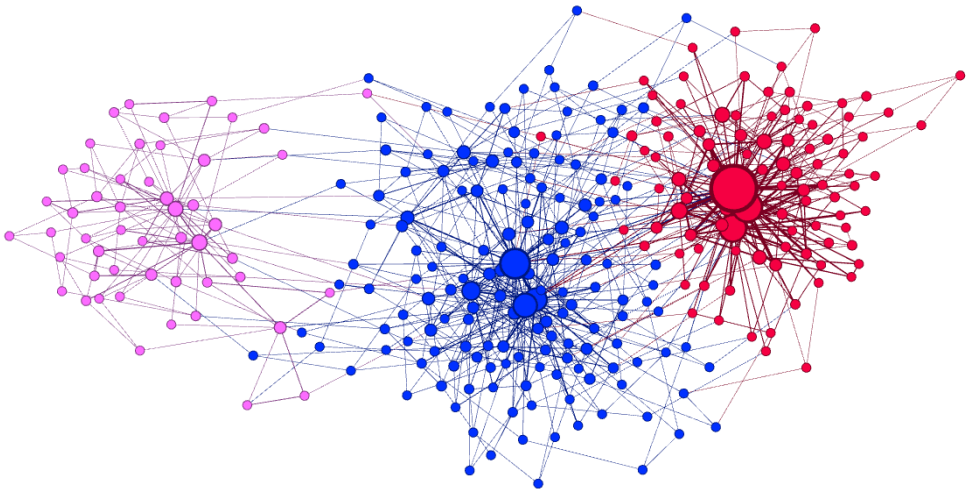


Figure 3.9: The final result of synthetic network_m1.

After step 2 is done, if there are merges of communities in step 2, it is possible that some nodes at the peripheral area of the merged communities should be changed to other communities from the effect of step 1. Therefore, if the change occurs in step 2, we have to perform step 1 again. And if there is any change in step 1, step 2 is needed to be performed again as well. The algorithm is finished when both step 1 and step 2 are at balanced state (nodes no longer change communities in both steps) or the threshold is met. The example of the final result is shown in Figure 3.9. Pink community is a community that node degree follows normal distribution. Blue and red communities are communities that node degree follows power law distribution. The NMI of this result is 0.958, which means the result is almost similar to the ground truth.

The summary of the community detection phase is shown in Figure 3.10.

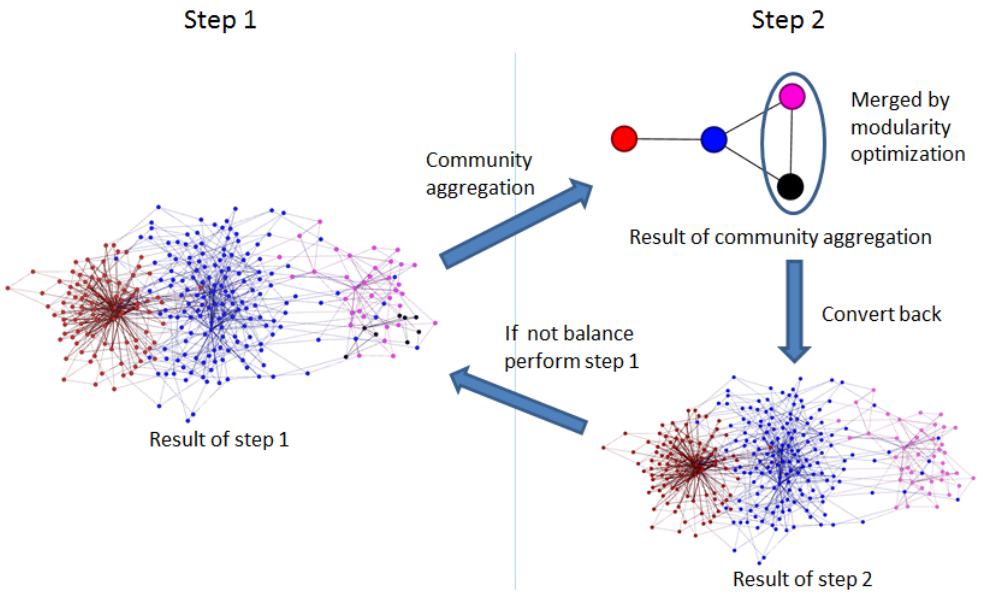


Figure 3.10: The loop between step 1 (scale-free method) and step 2 (non scale-free method).

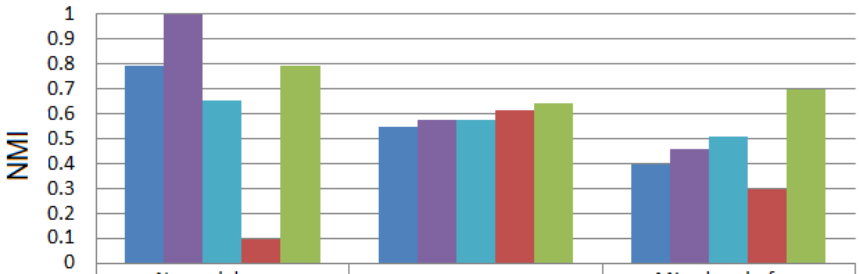
Configurable Parameters

There are two configurable parameters in this algorithm, attenuation factor and the depth l. Both are used when calculating edge weight. The term attenuation factor is used in Katz centrality (Foster et al. (2001)). Attenuation factor, normally symbolized by α, is used for penalizing the connections that are farther away. This value should be set to range from 1 to 0. Setting this value more than 1 will result in farther away edges are more important than the edge itself (switch from penalizing to rewarding edges for locating farther away). In our algorithm, α = 1/d is employed, where d is the average degree. This value is selected because it is shown in chapter 2 that this value gives the most balanced result. Similar to the previous contribution, coefficient value in this algorithm affects the outcome of the community detection result. However, the effect is not equally severe because our centrality is balanced out by average ratio, which makes the value does not significantly affect the outcome. Another configurable value is depth l. The higher the depth is, the more accurate the weighting scheme is. In our case, the minimum

l is 2. This is because we need three edges to form the triplets to calculate the clustering coefficient. From our observation, $l = 2$ (minimum value) provides the best trade-off point for calculation time and performance. However, in small networks, we used $l = 3$ in our experiments to increase the accuracy.

3.4 Experiments

In this contribution, we categorized networks into three categories; networks with normal distribution, networks with power law distribution (scale-free networks), and networks with mixed distribution. NMI is used to determine the similarity between two partitioning results. The higher the NMI is, the closer the result of community detection is to the ground truth. The maximum NMI score is 1.0 which means the result is exactly the same with the ground truth. All figures of networks in this research are displayed by Gephi (Bastian et al. (2009)) using Yihan Fu and ForceAtlas 2 layout algorithm.



	Normal degree distribution networks	Scale-free networks	Mixed scale-free networks
Louvain Algorithm	0.793	0.547	0.398
Girvan-Newman	0.996	0.572	0.456
Label Propagation	0.652	0.572	0.509
Node Centrality	0.095	0.615	0.294
Edge Weight	0.792	0.643	0.697

Figure 3.11: NMI results on normal degree distribution networks, scale-free networks, and mixed scale-free networks.

The first experiment is to test our algorithm on three categories of synthetic networks by comparing the results with modularity optimization based algorithm (Louvain algorithm), scale-free based algorithm (node centrality algorithm), edge

Table 3.1: The results of our proposed algorithm compared with other algorithms

Network Name	*Louvain Algorithm*	*Node Centrality*	*Proposed Algorithm*	*Edge Beweeness*	*Label Propagation*
Zachary karate club	0.359	1.0	1.0	0.370	0.678
American college football	0.765	0.789	0.765	0.723	0.725
Doubtful Sound dolphins	0.328	0.603	0.580	0.486	0.332
Political books	0.349	0.504	0.449	0.318	0.238

betweenness algorithm (Newman and Girvan (2003)), and label propagation algorithm (Raghavan et al. (2007)).

Mixing parameter μ of synthetics networks in this experiment is set to 0.2. Mixing parameter is equal to noises in networks, the higher the value, the more edges that connect between two communities. The results are shown in Figure 3.11. It can be seen that our method has about the same NMI as modularity optimization based method on normal distribution networks, better NMI than both node centrality and Louvain algorithm on power law distribution networks, and best NMI on mixed scale-free networks.

This is because in mixed scale-free networks, it is not necessary that each community is following power law distribution. Node centrality algorithm focuses solely on the characteristic of scale-free networks such as hub nodes can make mistakes when some communities do not follow scale-free characteristics. The example of this phenomenon is shown in Figure 3.1.

Edge betweenness algorithm received the highest score on normal distribution networks and received the second best score on scale-free networks. However, the most disadvantage point of edge betweenness algorithm is the time complexity. Because the algorithm needs to recalculate all the centrality values in every iteration, this algorithm is only suitable for small size networks. Another possibility is

that this network synthesizer is over-fitting with this benchmark graph generator. On the other hand, the result of label propagation is mediocre on both types of networks. However, label propagation algorithm is extremely fast. The authors claimed that the complexity is almost linear (Raghavan et al. (2007)).

The second experiment is to test our algorithm on real-world datasets that have truth value. Real-world networks used in this experiments are Zachary karate club network (Zachary (1977)), Doubtful Sound dolphins network (Lusseau et al. (2003)), American college football network (Girvan and Newman (2002)), and political books network (Girvan and Newman (2002)). The results are shown in Table 3.1, Figure 3.12, and Figure 3.13. In this experiment, edge betweenness algorithm did not receive the highest NMI scores in any networks, which may ensure that the synthetic networks used in the first experiment are over-fitting with edge betweenness algorithm.

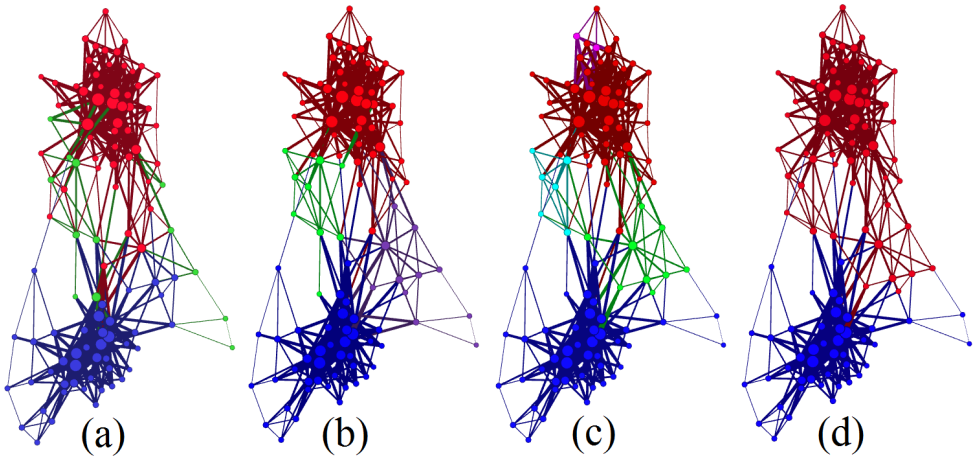


Figure 3.12: Political books network; (a) Ground truth , (b) Louvain algorithm - NMI = 0.349 , (c) Node centrality - NMI = 0.504 , (d) Proposed method - NMI = 0.449

The third experiment is to test our algorithm on synthetic scale-free networks but the mixing value is set from 0.1 to 0.4 by increasing mixing value by 0.05 per step. The result is shown in Figure 3.14. It can be seen that our proposed algorithm starts to fall off at mixing value 0.25. After that, both algorithms cannot maintain solid community detection result anymore.

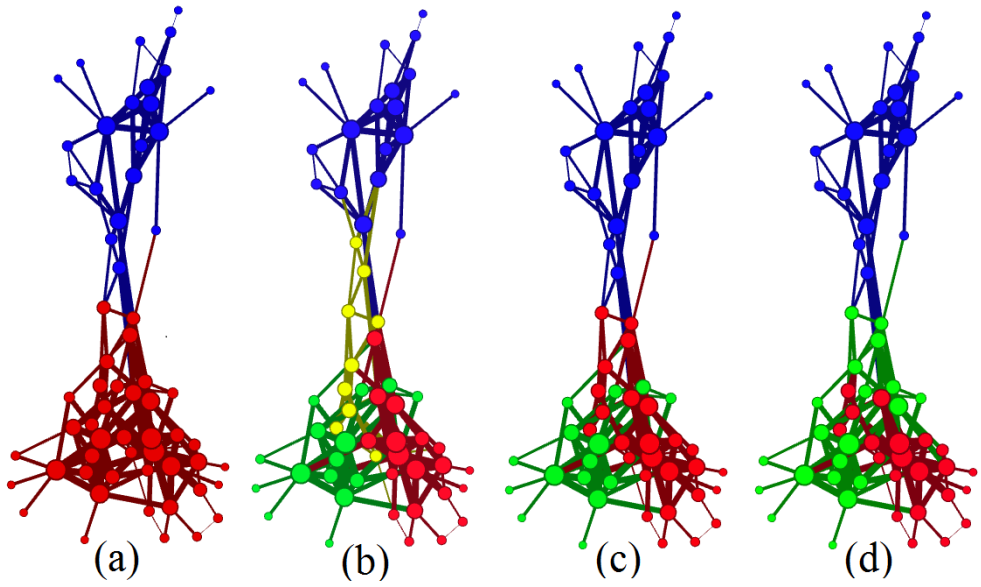


Figure 3.13: Doubtful Sound dolphins network; (a) Ground truth , (b) Louvain algorithm - NMI = 0.328 , (c) Node centrality - NMI = 0.609 , (d) Proposed method - NMI = 0.580

The last experiment is to apply our algorithm on real-world networks that do not have truth value. Four ego Facebook networks (McAuley and Leskovec (2012)) are selected to be displayed in Figure 3.15. The networks shown on the left column in Figure 3.13 are the results from Louvain algorithm. Networks shown in middle column are the results from node centrality algorithm, and our proposed method results are shown at the right column. Node size in Figure 3.15 represents the degree of that node. The bigger the size is, the higher its degree is. Communities with scale-free property can be seen as a group of nodes that has high degree node or nodes in the middle, surrounded by less degree nodes. On the other hand, normal distribution community is a group of nodes that have about the same size. Modularity optimization method sometimes tends to break scale-free community into many sub-communities from our previous experiments (Jarukasemratana et al. (2013)). This can be observed in the left column of Figure 3.15. On the other hand, normal scale-free based method tends to separate normal distribution community into many sub-communities as well (middle column -

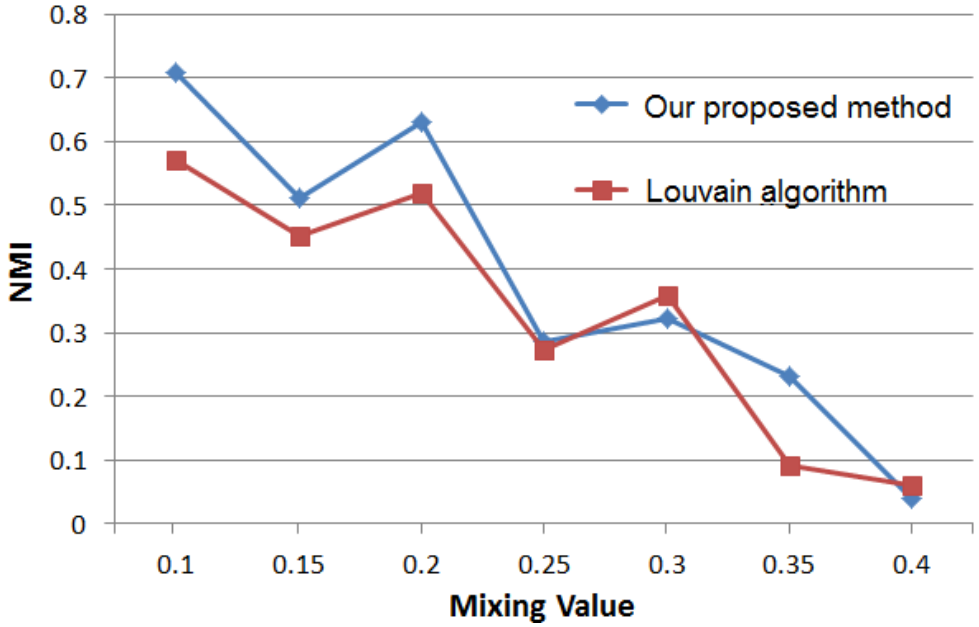


Figure 3.14: Results from the third experiment. Mixing value is equal to noises in networks.

Figure 3.15). What our method does is to take the advantages of both approaches and combines it into one result (right column - Figure 3.15).

3.5 Discussion

3.5.1 Experimental Results

For the first experiment, the first type of networks is the networks with normal distribution. Our algorithm achieves about the same NMI with modularity optimization method, receives better NMI when compared with scale-free and label propagation methods, and performed worse when compared with edge betweenness method. Next type of networks is the networks with power law distribution or scale-free networks. Our algorithm achieves about the same NMI with scale-free method on many networks. On mixed scale-free networks our proposed algorithm performs better. Our proposed algorithm also received better NMI when

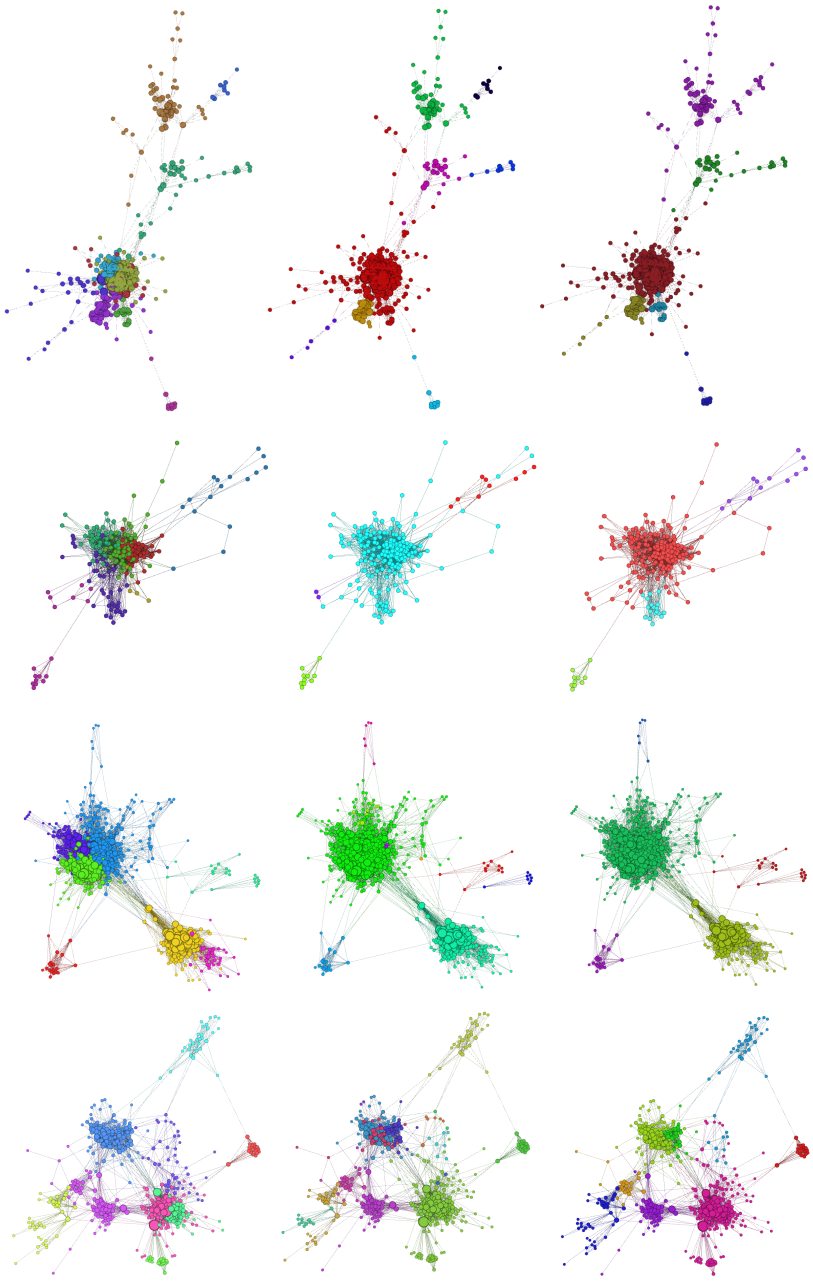


Figure 3.15: Results from the fourth experiment, a set of four Facebook networks. Results of Louvain algorithm, Node centrality algorithm, and our proposed algorithm.

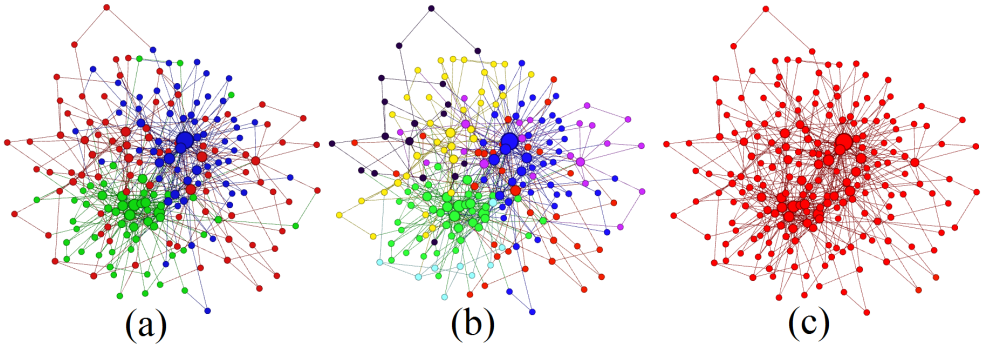


Figure 3.16: Synthetic network_m3, mixing value = 0.25, (a) Ground truth , (b) Louvain algorithm - NMI = 0.254 , (c) Proposed method - NMI = 0.0

compared with modularity optimization, edge betweeness, and label propagation methods. The result is as expected because our method is the combination of scale-free based and modularity optimization based method. However, the biggest disadvantage of our algorithm is the vulnerability against noises.

As shown in the third experiment, our algorithm is tested on scale-free networks where mixing parameters are ranging from 0.1 to 0.4. It can be seen that the average NMI of our algorithm drops very fast. Figure 3.16 shows the result of our proposed algorithm compared with ground truth and Louvain algorithm on synthetic network_m3. It can be seen that our algorithm considered the whole graph as one huge community, while the truth is not. The reason is because in the first step of our community detection phase, we merge nodes together from the highest edge weight to the lowest edge weight. In case of low noise situations, edges that connect between communities normally have low weight. These edges are processed in the later part of the algorithm where most communities are already formed up. However, when there are a lot of noises, it is possible that these noise edges will have high weight. Communities that are connected by the high weighted noise edges have a chance to merge together. In the worst case such as shown in Figure 3.16, the whole graph is merged into one large community. This phenomenon does not happen in modularity optimization based algorithm because before merging of communities, the modularity must be calculated first,

which means it is not possible for all communities to merge into one large community. We can use this idea to improve our algorithm in the future researches.

3.5.2 Time Complexity

Table 3.2: The computational time on large networks

$Networks$	$Nodes$	$Edges$	$Weighting$	$CommunityDetection$	$Time(seconds)$
GRQC	4158	13422	24	3	27
HepTh	6301	20777	26	1	27
Gnutella8	8638	24806	19	12	31
PGP	10,680	24,316	43	12	55
DBLP	317,080	1,049,866	3029	3161	6190

Another aspect of the algorithm that needs to be mentioned is the complexity or computational time. The trade-off for combining two algorithms into one is that our method has many steps, which is costly. In edge weight phase, the complexity is $O(mk^l)$, where m is the number of edges, k is the average degree, and l is the depth of the algorithm. Minimum value of l is 2. It should be noted that in this phase, if the number of edges is the same, networks with more nodes are faster. This means that our algorithm is suited for sparse networks than dense networks. The first step in community detection requires a run through all nodes once per iteration. The number of iteration required is unknown, but from our experiments the number of iterations never exceed 10. Thus complexity in this step is $O(nk)$, where k is the average degree and n is the number of nodes. The second step has the complexity of Louvain algorithm with slight difference because the number of nodes is the number of communities aggregated from previous steps. Therefore, the complexity of this step is $O(n \log n)$, where n is the number of communities aggregated from previous steps, which is normally small. Based on our test on various networks, in most case, more than 70% of the time is spent on edge weight phase. Therefore, we like to assume that the complexity of our algorithm follows the complexity of edge weight step, which is $O(mk^l)$, where m is number of edges, k is average degree, and l is the depth of the algorithm.

We test our algorithm on real-world datasets of various sizes to measure our algorithm computational time. The networks used are Arxiv General Relativity and Quantum Cosmology collaboration network (GRQC) (Leskovec et al. (2007)), Gnutella peer to peer network 8 (Ripeanu et al. (2002)), Arxiv high energy physics theory citation graph (HepTh) (Leskovec et al. (2005)), Pretty Good Privacy network (PGP) (Boguña et al. (2004)), and DBLP collaboration network (Yang and Leskovec (2012)). The size and the computational time of each network are shown in Table 3.2.

3.5.3 Weighted Modularity

Since NMI is usable only when ground truth is available, this makes using NMI with real-world datasets almost impossible. For example, in Facebook datasets, there is no ground truth. When ground truth is not available, usually, modularity (Girvan and Newman (2002)) is used to determine the goodness of the community detection results. However, it is shown that modularity is not suitable for scale-free networks (Jarukasemratana et al. (2013)). To come up with better evaluation score, we tried to improve modularity with our edge weight technique. In unweighted networks, values in adjacency matrix are either 0 or 1 depending on if there is an edge between corresponding nodes or not. However, when applying our edge weighting scheme, the value in each cell is modified to match with the weight of each edges. By using these modified adjacency matrix, we can calculate new modularity value.

We performed an experiment to compare the maximum modularity, maximum weighted modularity, and our proposed method. Maximum modularity and maximum weighted modularity is obtained by using Louvain algorithm (Blondel et al. (2008)). The results are shown in Table 3.3.

From Table 3.3, it can be seen that maximizing weighted modularity yields better NMI results than normal modularity. However, our proposed method still yields the best NMI results. One of the benefits for choosing maximizing weighted modularity is the time complexity. The complexity of normal Louvain algorithm is $O(n \log n)$. The complexity for edge weight calculation is $O(mk^l)$, where m is the number of edges, k is the average degree, and l is the depth of the algorithm. In case of extremely large networks, where our proposed method is not feasible,

Table 3.3: The results of Louvain method, Louvain method with weighted edges, and proposed algorithm.

$Network\ Name$	$Modularity\ (NMI)$	$WeightedModularity\ (NMI)$	$ProposedAlgorithm\ (NMI)$
Zachary karate club	0.359	0.432	1
American college football	0.765	0.765	0.765
Doubtful Sound dolphins	0.328	0.412	0.580
Political books	0.349	0.356	0.449

it is possible to use weighted modularity for a tradeoff between accuracy and computation time. It is also possible to tweaking the edge weight scheme so that when maximizing the weighted modularity, the NMI results are better.

Moreover, by using our weighting scheme with modularity optimization method makes our algorithm more robust to noises. Figure 3.17 shows the result of Louvain method, weighted Louvain method, and our proposed algorithm on synthetic networks.

However, weighted Louvain receives NMI almost equal to Louvain method when performing on mixed scale-free networks. This is because scale-free part is removed from the algorithm. Therefore, weighted Louvain cannot identify scale-free communities correctly.

3.6 Conclusions

In this contribution, we propose an algorithm for community detection that specifically aim at scale-free networks which contain both communities that node degree

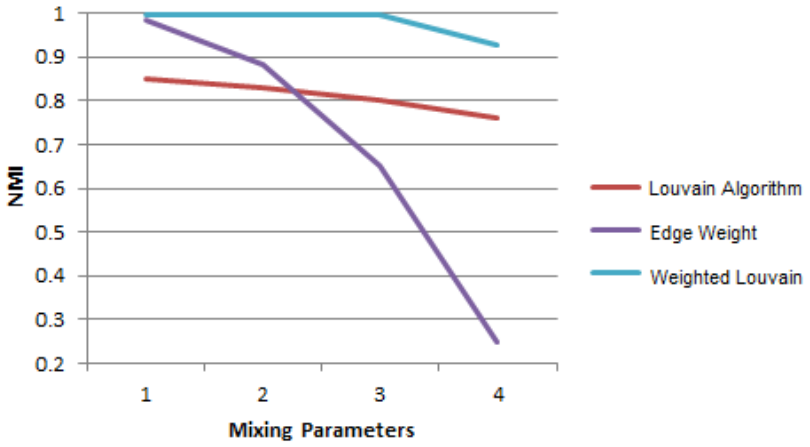


Figure 3.17: NMI results on 4 noises level on scale-free networks with 3 algorithms; Louvain method, edge weight method, and weighted Louvain.

follow normal distribution and communities that node degree follows power law distribution, which we call “mixed scale-free network”. When this algorithm is applied to networks with normal distribution, the result will be similar to those from modularity optimization based algorithm. If this algorithm is applied to scale-free networks where every community follows scale-free properties, the result will be similar to those from scale-free based method. And if this algorithm is applied to mixed scale-free networks, the result will be superior when compared with other methods. The key concept is that, we perform scale-free based community detection method to extract scale-free communities and then perform modularity optimization based method to extract normal distribution communities. We perform experiments to test our method by comparing the results with modularity optimization based, scale-free based, edge betweenness, and label propagation methods. We used famous real-world datasets which have ground truth available and synthetic datasets. NMI is used for evaluation. Our method performs equally to the previous approaches on networks with normal distribution. On scale-free networks, our method is superior. Therefore, if the type of the networks is unknown, our method should be used because the result is equal or better than previous methods. The disadvantage of our approach is the vulnerability to noises

which can result as a whole network becomes one large community. Our further research is to make this algorithm more robust to noises and receive better accuracy and computational time.

Chapter 4

Discussions

In this thesis, we introduce two community detection algorithms for scale-free networks. The first algorithm, node centrality algorithm, is aimed for scale-free networks by using hub nodes, which is one of the most important characteristics of scale-free networks. The second algorithm, edge weight algorithm, is aimed specifically for sub-type of scale-free networks that some communities do not follow scale-free characteristics, which we call "mixed scale-free networks". The advantages, disadvantages, and differences between these two algorithms are discussed in this chapter.

4.1 Computational Time

Computational complexities of both algorithms are $O(mk^l)$, where m is the number of edges, k is the average degree, and l is the adjustable depth of the algorithm which is normally set as 2. This complexity belongs to the most time consuming parts of each algorithm, node centrality calculation and edge weight calculation respectively. However, in real practices, edge weight algorithm uses more time than node centrality algorithm. In node centrality algorithm, more than 90% of the calculation time is spent on centrality calculation. After the centrality calculation is done, the community detection step can be finished in one run through all edges. This means the total computation time of node centrality algorithm is $O(mk^l) + O(m)$. On the other hand, edge weight algorithm is much more complex. Community detection phase of edge weight algorithm consists of two steps,

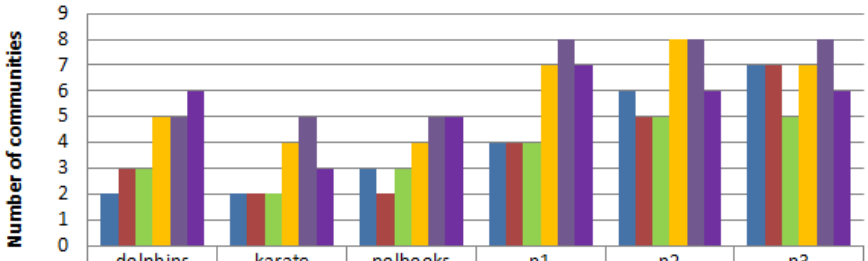
Table 4.1: The number of nodes, number of edges, and computational time on various networks of node centrality and edge weight algorithms

$Networks$	$Nodes$	$Edges$	$NodeCentrality(seconds)$	$EdgeWeight(seconds)$
GRQC	4158	13422	4	27
HepTh	6301	20777	8	27
Gnutella8	8638	24806	17	31
PGP	10,680	24,316	11	55
DBLP	317,080	1,049,866	1560	6190

one for scale-free communities and one for non scale-free communities. Moreover, these two steps must be done iteratively until the result is converged or the threshold is met. Rough estimation of total computational time of edge weight algorithm is $O(mk^l)+t(O(nk)+O(n\log n))$, where n is number of node and t is the number of times until convergence which is set to maximum at 10. The exact running times of both algorithms on the same networks are shown in table 4.1. It can be seen that if the graph is large, node centrality algorithm can provide much faster result than edge weight algorithm.

4.2 Accuracy versus Computational Time

It is shown in chapter 3, experiments section, that edge weight algorithm can achieve better NMI scores than node centrality algorithm on non scale-free networks and mixed scale-free networks, while received similar NMI on scale-free networks that do not contain non scale-free communities. More importantly, edge weight algorithm is extremely flexible because the algorithm incorporates with Louvain algorithm. This means that edge weight algorithm can handle non scale-free graphs, scale-free graphs, or mixed scale-free graphs effectively. However, edge weight algorithm is slower when compared to node centrality algorithm. In real practice, if the size of the network is not too large, the edge weight algorithm is recommended.



	dolphins	karate	polbooks	n1	n2	n3
Truth value	2	2	3	4	6	7
Node Centrality	3	2	2	4	5	7
Edge weight	3	2	3	4	5	5
Louvain	5	4	4	7	8	7
Edge betweeness	5	5	5	8	8	8
Label Propagation	6	3	5	7	6	6

Figure 4.1: Number of communites of 6 scale-free networks from various community detection algorithms.

4.3 Accuracy of the Number of Communities

In community detection, apart from the accuracy, another important result is the number of communities. In some algorithm such as top leader (Khorasgani et al. (2013)) the number of communities is required beforehand. However, both of our algorithms can generate the number of communities on the fly. The number of communities of scale-free networks from our algorithms, Louvain algorithm, Girvan-Newman algorithm, and label propagation are shown in Figure 4.1.

Node centrality and edge weight algorithm both receive 3 correct numbers of communities, which is more accurate than traditional methods. The reason is because those methods tend to break large scale-free community into smaller communities. This is why the number of communities from Louvain method, label propagation and Girvan-Newman edge betweenness are usually higher than the true value of the number of communities.

4.4 Centrality Parameter and Parameters of Scale-free Networks

In our algorithm, coefficient of our Katz centrality equals to

$$\alpha = 1/d,$$

where α is coefficient value, and d is average degree of the graph. It is shown in chapter 2 that this value can prevent the case of too high or too low coefficient value, which prevents the graph merging and graph breaking. This section is intended to show the relation between α and others parameter of scale-free networks. As shown in chapter 1, degree distribution graphs of scale-free networks can be written as

$$y = \beta x^{-k} + \varepsilon,$$

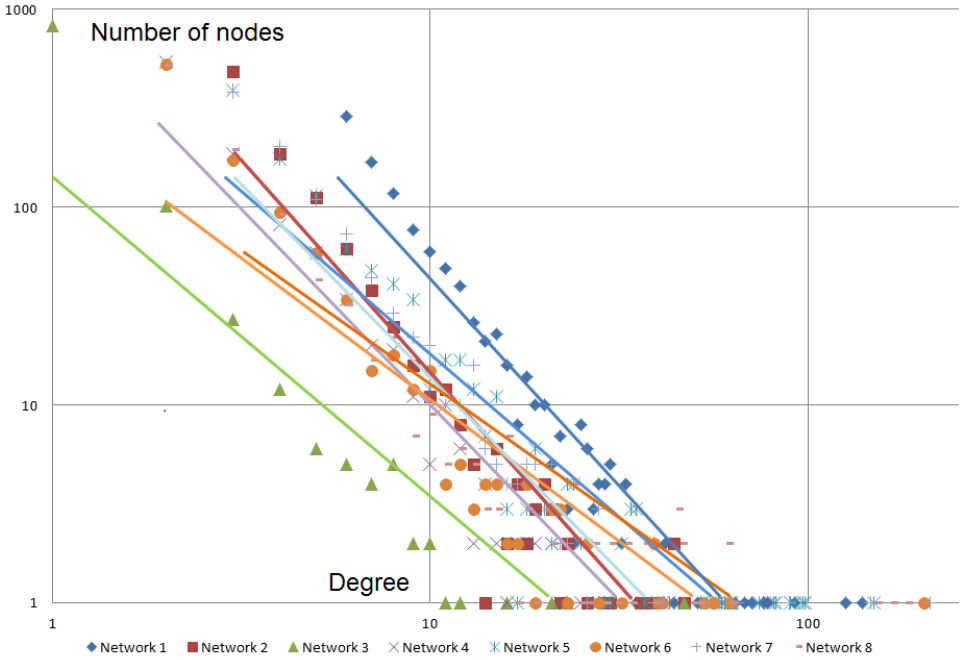
where β is a constant value, k is a coefficient of scale-free, and ε represents a deviation of the formula. Normally, k is within a range between 2 and 3. This degree distribution follows a power law distribution. One characteristic of this degree distribution is that when it is plotted in log-log scale, the graph will become a straight line, which is represented as the following formula:

$$\log y = -k \log x + \log \beta,$$

which will become similar to the formula of straight line where $-k$ is slope and $\log \beta$ is Y intercept.

To show the relations between our Katz coefficient and degree distribution of scale-free networks, we use 8 synthetic networks that all have number of nodes equal to 1000 but with different average degree and different scale-free coefficient. Degree distribution graph in log-log scale is shown in Figure 4.2(a) and details of each network are shown in Figure 4.2(b).

Since k (the scale-free coefficient) refers to the slope of the node degree distribution line, the higher the k value, the faster the y value reaches to 0. If two networks have similar Y intercept values, the higher k results in the less degree of the highest degree nodes. This results in the lower of average degree, and increase Katz coefficient used in our experiments. For example, network 1 and network 2 have similar Y intercept values (3.564 and 3.419 respectively), but network 1



(a)

Network	Number of nodes	Number of edges	Average degree	Scale-free coefficient	β	Y intercept (log β)
1	1000	10668	10.668	1.970	3667.80	3.564
2	1000	4996	4.996	2.269	2625.30	3.419
3	1000	1520	1.520	1.596	138.77	2.142
4	1000	3634	3.634	1.932	807.20	2.906
5	1000	6858	6.858	1.576	602.42	2.779
6	1000	4534	4.534	1.394	254.47	2.405
7	1000	7806	7.806	1.231	214.16	2.330
8	1000	5208	5.208	1.110	110.92	2.045

(b)

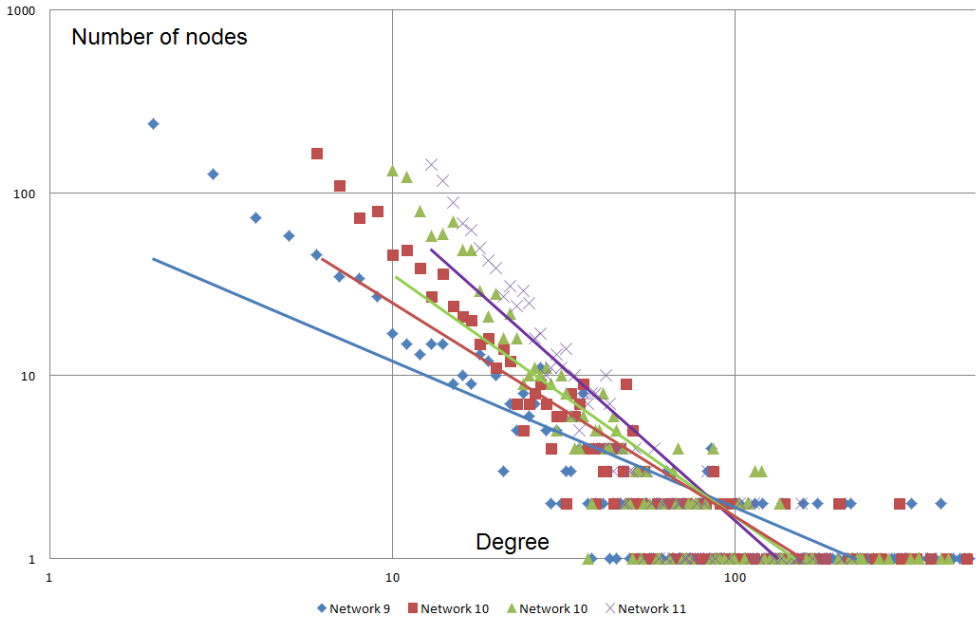
Figure 4.2: (a) degree distribution graph of 8 synthetic networks. (b) details of each network.

has lower scale-free coefficient (1.97) than coefficient of network 2 (2.269). This makes average degree of network 1 higher than network 2 (10.668 and 4.996).

Our Katz coefficient, which is based on average node degree, is based on two variables, scale-free coefficient k and Y intercept $\log \beta$. This means that it is possible for many networks to have the same Katz coefficient values even when k and $\log \beta$ are not the same in each network. We created synthetic networks where the average degrees and number of nodes are the same but k and β are different. Degree distribution graph in log-log scale is shown in Figure 4.3(a) and details of each network are shown in Figure 4.3(b).

From Figure 4.3(b), it can be seen that when average degree is about the same value for all networks, Y intercept $\log \beta$ shows the direct relation to the scale-free coefficient k and vice versa. This is because according to the formula $-k$ is acting as a slope of the graph. To maintain the same average degree, if Y intercept value is high, the scale-free coefficient need to be high too.

The conclusion of relations between our coefficient of Katz centrality and parameters of scale-free networks are as follows. If the coefficient of Katz centrality is fixed, scale-free coefficient and Y intercept values are in a direct relation with each other, which is shown in Figure 4.3. If Y intercept is fixed, the higher ceofficient of Katz centrality results in the reduction of scale-free coefficient, which means these two parameters are in an inverse relationship. And if scale-free coefficient is fixed, Y intercept and Katz centrality value are also in an inverse relationship.



(a)

Network	Number of nodes	Number of edges	Average degree	Scale-free coefficient	β	Y intercept (log β)
9	1000	26280	26.281	0.869	92.72	1.967169
10	1000	25464	25.464	1.139	309.13	2.490141
11	1000	26474	26.474	1.296	659.17	2.818997
12	1000	25126	25.126	1.704	3673.21	3.565045

(b)

Figure 4.3: (a) degree distribution graph of 4 synthetic networks with about the same average degree. (b) details of each network.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis is devoted to community detection algorithms for scale-free networks. In chapter 1, two main related topics of this thesis are introduced, community detection and scale-free networks. Community detection, so far, has been studied by researchers from many fields such as biology, physics, and network science. Therefore, there are many researches about community detection algorithms. On the other hand, the term scale-free networks was invented not so long ago, thus there are many aspects left to be researched. This includes community detection algorithms for scale-free networks. In this thesis, we combine the knowledge from various community detection researches and scale-free networks researches, to create community detection algorithms that befitted scale-free networks.

In chapter 2, we introduced our first contribution, community detection algorithm for scale-free networks based on node centrality and edge distance. In this approach, we focused on one of the most important characteristic of scale-free networks which are hub nodes. There are two main ideas in this algorithm. The first idea is that community should be formed from hub nodes. To identify hub nodes, we use centrality value. We developed our own centrality value based on two previous research which are Katz centrality (Katz (1953)) and subgraph centrality (Estrada and Rodríguez-Velázquez (2005)). By using the strength of both centrality approaches our centrality value can distinguish hub nodes from other nodes. The second main idea is that nodes that are closely connected to each other

should be in the same community. To determine if the nodes are closely connected together or not, we calculate edge distance value. Our community detection begins with a node with the highest centrality. Using edge distance, community is spread from the highest centrality node. Then, the node with the next highest centrality value is processed. Algorithm is finished when all nodes are labeled with community. Our algorithm achieved higher NMI scores when compare with other community detection methods on scale-free networks. However, there is a problem when some communities in scale-free networks do not have hub nodes. Since our algorithm defines communities based on hub nodes, our algorithm cannot perform correctly when communities do not have hub nodes. The solution to this problem is offered in our second contribution.

In chapter 3, we introduced our second contribution, community detection algorithm for scale-free networks based on edge weight. We call scale-free networks that some communities do not have scale-free properties "mixed scale-free networks". This contribution is aimed for this sub-type of scale-free networks. The main idea of this contribution is that each edge in scale-free networks is not equally important. Edges that connect to hub nodes should be more important than edges that connect nodes in peripheral area. To obtain the importance of each edge, we modified our centrality approach which was used in our first contribution to use on edges instead of nodes. Each edge is weighted according to its centrality value. The higher the weight means the more the importance of this edge. To cope with mixed scale-free networks, we divided our community algorithm into 2 steps. The first step is to extract communities that have scale-free properties. This step is performed by grouping nodes into the same communities based on the weight of edges that connect them together. Communities that have scale-free properties are extracted almost completely in this step. The second step is to extract communities that do not have scale-free properties. In this step we apply weighted Louvain method to extract communities that do not have scale-free properties. Since our second contribution contains both scale-free and non scale-free approaches, it achieved the same NMI scores as non scale-free methods on non scale-free networks, same NMI scores as scale-free methods on scale-free networks, and superior NMI scores on mixed scale-free networks.

5.2 Current Problems and Future Works

One of the biggest concerns for community detection algorithms is time complexity. Large datasets such as WWW datasets can easily exceed tens or hundreds of millions of nodes and ten times or more edges. It is a challenge for every algorithm to be able to cope with time complexity and space complexity. In our case, both of our algorithms are running at $O(mk^2)$, where m is the number of edges, k is the average degree, and 2 is the adjustable depth of the algorithm which in this case we use 2. This suggests that both of our algorithms are suitable for sparse graphs. Another method to improve the calculation time is to run on parallel platform or run on multiple CPUs. Currently, our algorithms can only partially run on multiple cores, which will create a bottle neck on the process that cannot be run in parallel.

Another problem of our approaches is the vulnerability to noises. The foundation of our community detection result is based largely on the centrality value calculation. Nodes and edges that are processed first are usually the one with more weight or higher centrality than others. However, if the amount of noises in the network is high, it is possible that some noises will be treated as very important nodes or edges. In this case, our algorithms will fail since the most important nodes or edges are misclassified. To solve this problem, the most important step that needs to be improved is the centrality calculation step. Currently, our approaches rely on Katz centrality (Katz (1953)) and subgraph centrality (Estrada and Rodríguez-Velázquez (2005)). Despite the fact that the combination of Katz and subgraph centrality can identify the important nodes and edges correctly, these two methods are vulnerable to noises. There are many other centrality method such as betweenness centrality that may possible to replace Katz and subgraph centrality to improve the robustness.

Another topic worth further investigation is the improvement of modularity based on our weighting scheme from our second contribution. Modularity (Girvan and Newman (2002)) is probably the easiest way to determine the goodness of the community detection results and it does not require ground truth value. However, it is shown that modularity is not suitable for scale-free networks in our thesis. In our second contribution, we have shown that using our weighting scheme to transform unweighted networks to weighted networks then apply

modularity optimization algorithm can increase the correctness of the community detection. With this knowledge and further researches and experiments, it is possible that the new modularity value that can work on both scale-free and non scale-free networks can be discovered.

Bibliography

Amine Abou-Rjeili and George Karypis. Multilevel algorithms for partitioning power-law graphs. In *Proceedings of the 20th international conference on Parallel and distributed processing*, IPDPS'06, pages 124–124, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 1-4244-0054-6. URL http://dl.acm.org/citation.cfm?id=1898953.1899055.

Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 33–42, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1228-8. doi: 10.1145/2380718.2380723. URL http://doi.acm.org/10.1145/2380718.2380723.

Albert-László L. Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):60–69, May 2003. ISSN 0036-8733. URL http://view.ncbi.nlm.nih.gov/pubmed/12701331.

Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), July 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/P10008. URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

Marián Boguña, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Phys. Rev. E*, 70:056122, Nov 2004. doi: 10.1103/PhysRevE.70.056122. URL http://link.aps.org/doi/10.1103/PhysRevE.70.056122.

Phillip Bonacich. Some unique properties of eigenvector centrality. *Social Networks*, 29(4):555–564, October 2007. ISSN 03788733. doi: 10.1016/j.socnet.2007.04.002. URL http://dx.doi.org/10.1016/j.socnet.2007.04.002.

Steve Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, January 2005. ISSN 03788733. doi: 10.1016/j.socnet.2004.11.008. URL http://dx.doi.org/10.1016/j.socnet.2004.11.008.

Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Grke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity – np-completeness and beyond, 2006.

Derek J. de Solla Price. Networks of Scientific Papers. *Science*, 149(3683):510–515, July 1965. ISSN 1095-9203. doi: 10.1126/science.149.3683.510. URL http://dx.doi.org/10.1126/science.149.3683.510.

Paul Erdös and Alfréd Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959. URL http://www.renyi.hu/~p_erdos/Erdos.html#1959-11.

Ernesto Estrada. *The Structure of Complex Networks:Theory and Applications: Theory and Applications*. OUP Oxford, 2011. ISBN 9780199591756. URL http://books.google.co.jp/books?id=pCAaYAAACAAJ.

Ernesto Estrada and Juan A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 71(5 Pt 2), May 2005. ISSN 1539-3755. URL http://view.ncbi.nlm.nih.gov/pubmed/16089598.

Santo Fortunato. Community detection in graphs. *Physics Reports 486*, pages 75–174, January 2010. URL http://arxiv.org/abs/0906.0612.

Santo Fortunato. Benchmark graphs to test community detection algorithms, https://sites.google.com/site/santofortunato/, 2011. URL https://sites.google.com/site/santofortunato/.

Kurt Foster, Stephen Muth, John Potterat, and Richard Rothenberg. A Faster Katz Status Score Algorithm. *Computational & Mathematical Organization Theory*, 7(4):275–285, December 2001. ISSN 1381298X. doi: 10.1023/A:1013470632383. URL http://dx.doi.org/10.1023/A:1013470632383.

Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977. ISSN 00380431. doi: 10.2307/3033543. URL http://dx.doi.org/10.2307/3033543.

Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy*

of Sciences of the United States of America, 99(12):7821–7826, June 2002. ISSN 0027-8424. doi: 10.1073/pnas.122653799. URL http://dx.doi.org/10.1073/pnas.122653799.

Mathieu Jacomy. Force atlas 2 layout, http://gephi.org/2011/forceatlas2-the-new-version-of-our-home-brew-layout/, 2011.

Sorn Jarukasemratana, Tsuyoshi Murata, and Xin Liu. Community detection algorithm based on centrality and node distance in scale-free networks. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 258–262, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1967-6. doi: 10.1145/2481492.2481527. URL http://doi.acm.org/10.1145/2481492.2481527.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, March 1953.

Reihaneh Rabbany Khorasgani, Jiyang Chen, and Osmar R. Zaane. Top leaders community detection approach in information networks. In *in Proceedings of the 4th Workshop on Social Network Mining and Analysis, 2010. ISSN : 2319-7323*, page 228, 2013.

Douglas J. Klein and Milan Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, December 1993. ISSN 0259-9791. doi: 10.1007/BF01164627. URL http://dx.doi.org/10.1007/BF01164627.

Valdis Krebs. Political book networks, http://www.orgnet.com/divided.html, 2004.

Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80: 056117, Nov 2009. doi: 10.1103/PhysRevE.80.056117. URL http://link.aps.org/doi/10.1103/PhysRevE.80.056117.

Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 78(4), 2008.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3), March 2009. ISSN 1367-2630. doi: 10.1088/1367-2630/11/3/033015. URL http://dx.doi.org/10.1088/1367-2630/11/3/033015.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 177–187, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081893. URL http://doi.acm.org/10.1145/1081870.1081893.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217301. URL http://doi.acm.org/10.1145/1217299.1217301.

David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003. doi: 10.1007/s00265-003-0651-y. URL http://dx.doi.org/10.1007/s00265-003-0651-y.

Julian McAuley and Jure Leskovec. Discovering Social Circles in Ego Networks, October 2012. URL http://arxiv.org/abs/1210.8182.

Mark E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104+, July 2006. ISSN 1539-3755. doi: 10.1103/physreve.74.036104. URL http://dx.doi.org/10.1103/physreve.74.036104.

Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113+, August 2003. ISSN 1539-3755. doi: 10.1103/PhysRevE.69.026113. URL http://dx.doi.org/10.1103/PhysRevE.69.026113.

Rong Qian, Wei Zhang, and Bingru Yang. Community detection in scale-free networks based on hypergraph model. In *Proceedings of the 2007 Pacific Asia conference on Intelligence and security informatics*, PAISI'07, pages 226–231, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71548-1. URL http://dl.acm.org/citation.cfm?id=1763599.1763625.

Usha N. Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. 76(3):036106, September 2007. doi: 10.1103/PhysRevE.76.036106.

Matei Ripeanu, Ian Foster, and Adriana Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6:2002, 2002.

Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998. ISSN 0028-0836. doi: 10.1038/30918. URL http://dx.doi.org/10.1038/30918.

Andrew Y. Wu, Michael Garland, and Jiawei Han. Mining scale-free networks using geodesic clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 719–724, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014146. URL http://doi.acm.org/10.1145/1014052.1014146.

Jierui Xie and Boleslaw K. Szymanski. Community detection using a neighborhood strength driven label propagation algorithm. *CoRR*, abs/1105.3264, 2011.

Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 824–833, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281280. URL http://doi.acm.org/10.1145/1281192.1281280.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *CoRR*, abs/1205.6233, 2012.

Wayne W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4): 452–473, 1977. ISSN 00917710. doi: 10.2307/3629752. URL http://dx.doi.org/10.2307/3629752.

Publications Related to This Thesis

Sorn Jarukasemratana, Tsuyoshi Murata, "Community Detection in Scale-Free Networks using Edge Weight and Modularity Optimization Method", To be appeared in Transactions of the Japanese Society for Artificial Intelligence, Vol.30, No. 1, 2015

Sorn Jarukasemratana, Tsuyoshi Murata, Xin Liu, "Community Detection Algorithm based on Centrality and Node Closeness in Scale-Free Networks", Transactions of the Japanese Society for Artificial Intelligence, Vol.29, No.2 p. 234-244, 2014, $10.1527/tjsai.29.234$

Sorn Jarukasemratana, Tsuyoshi Murata, "Edge Weight Method for Community Detection in Scale-Free Networks", Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics, Thessaloniki, p. 1-9, Greece, 2014, $10.1145/2611040.2611065$

Sorn Jarukasemratana, Tsuyoshi Murata, Xin Liu, "Community Detection Algorithm based on Centrality and Node Distance in Scale-Free Networks", Proceedings of the 24th ACM Conference on Hypertext and Social Media p. 258-262, 2013, $10.1145/2481492.2481527$

Other Publications

Sorn Jarukasemratana, Tsuyoshi Murata, "Web Caching Replacement Algorithm Based on Web Usage Data", New Generation Computing, Vol. 31, No. 4 p. 311-329, 2013, $10.1007/s00354-013-0404-z$

Sorn Jarukasemratana, Tsuyoshi Murata, "Recent Large Graph Visualization Tools : A Review", Computer Software (JSAI) , Vol. 30, No. 2 p. 159-175, 2013, $10.11309/jssst.30.2_159$

Sorn Jarukasemratana, Tsuyoshi Murata, "Improved Web Cache Replacement Policy using Web Usage Data", Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence, Japan, 2012

Sorn Jarukasemratana, Tsuyoshi Murata, "Visualizing Web Structure based on Browsing Sessions", Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction (APCHI2012), p. 89, Aug. 2012