Abstract

This project is concerned with finding local community structure in real network. Although much structured data can be presented as network, such as World Wide Web, social network, citation network and biological network, but there exist a common property that many of them possess: the network community structure, which means that there exits an area which have more edges between vertices within a group than outside.

Recently, detection community structure in network has attracted much research attention. However, most approaches require the entire knowledge of the network structure. This is a problem due to some networks are too large and too complex to even be known completely, for example, World Wide Web. In this paper, we implement a community detection algorithm for large network that starting the particular vertex then finds the community that vertex belongs.

Chapter two is the background information. Chapter three describes some essential information about local algorithm implementation. Chapter four and five analyses the performance from real network and synthetic network for evaluation the algorithm. The last chapter compares the performance of local algorithm and breath first searching algorithm for identify its property of community detection.

Acknowledgement

Firstly I would like to thank my supervisor Steve Gregory for his guide and give me the chance to do this project. It is good training both for the programming and problem solving skills. I am very grateful for his continuous help and support.

I would also like to thanks Bowen and other teachers for their guidance and helpful discussions provided when problems were encountered in this project.

Much gratitude is paid to my parents for their love and support, without their financial support I wouldn't have been able to put so much time and effort in to my studies.

This project is dedicated to Jia Liu, with my love.

Contents

List of Figures

Pages

Chapter I Introduction

1.1 Aims and Objectives

The aim of this work is to write a piece of software that can detect the community structure within a large complex network (myspace network) and simultaneously construct a real network.

This work aims to complete three main objectives below.

Objective1. To implement the software that can fetch information from web and simultaneously represent the relevant data.

The author will create a graphical user interface application for detection community structure in the real network. The crawler will be applied to find the adjacent information, and the algorithm for the local community finding will be used to agglomerate vertex in the community. Besides, the relevant data should be dynamically visual in the user graph interface thus the researcher can observe how the network constructed.

Objective2. To construct the synthetic network for testing the accuracy of the algorithm.

Computer-generated graphs are the standard method to test community finding algorithm. This synthetic network has full known community structure. In this case, 64 vertices have been divided into four equal-sized communities of 16 vertices each. In addition, each node has degree $Z = Z_{in} + Z_{out} = 8$ (Z_{in} means the number of edges are placed with intra-community and Z_{out} is represent number of edges inter-community). Therefore, we can apply the program to explore these graphs and observe how good the structure found is.

Objective3. To compare the different community detection algorithms

In this task, the breadth first searching will be used to compare the local algorithm. The performance of these algorithms will be exhibited respectively. Furthermore, the different terminal condition will be discussed due to it may directly affect the accuracy of community detection.

1.2 Motivation

Community involves a lot of useful information, such as knowledge sharing and personal preference. It is clear that exploring the network community structure has a significant effect on commercial activity and scientific research. Therefore, it widely applies to biological, physical, and sociological fields. Moreover, in terms of e-commerce, the technology of local community finding not only provides an appropriate model to improve the sales service from

buyer, but also may enhance the customer's loyalty.

Furthermore, a majority of researchers pay attention to finding all communities in a complete network. However, the disadvantage of this work is hardly getting the entire knowledge of the network, especially with World Wide Web. Therefore, how to detect the local community structure efficiently is a key point in this research aspect. In addition, how to measure the quality of the community is worth for further research. In this paper, researcher will describe the standard approach of testing the accuracy of the algorithm.

This work may be considered as useful since it could fetch information from web. However, the difficulty within this work is the sample network which has many limitations. For example, some personal pages have been defined as private information thus the program cannot access them. In other words, some potential relationships probably are ignored. It may affect the performance of community detection. Thus, it is vital to choose an appropriate starting vertex to avoid some areas which have not community structure exist.

Chapter II Background Information

2.1 The Different Perspective of Community

This part states the fundamental impression of community and collects different opinions from researcher, and then followed with types of the virtual community.

Community has various forms as culture or religion, so it may be difficult to give the exact definition. In particular, Worsely (1987) had provide the three general description and present below.

1. It can refer to the regional community, such as the residential area.
2. It can be expressed as some networks which have been connected each other.
3. It involves the special relationship, such as the community's spirit and feeling.

[11].

People who are located within community can be called members. They often have the similar ideas or individual interests in the special area. For example, people those want to be together because they like to play Go, so in this community they can discusses the problem occurred in their own game and easily find the adversary. Therefore, the relationship could be easily established among them than outsiders. Furthermore, virtual community has the similar property comparing the community in real life. In this aspect, such as the internet, people can use some software to communicate to each other. Meanwhile, they can share their knowledge, provide the technical support and launch news mutually.

There are some opinions about the community.

Rheingold (1993) believes that the virtual community is a group of people who used the internet to exchange the information and threat the other member as the friend or family member [19].

In contrast, Wellmn (1999) presented a different idea. He declared that the community is just a tie that can connect the many social actors. As for the network, such as the internet or intranet, although it provides a platform to exchange the information, it is just a kind of connection type [19].

Moreover, Romm (1999) states that virtual community is a fresh phenomenon, the member in that area often have the high loyalty, and sharing or exchanging the knowledge or personal opinion [19].

2.1.1 The Types of Virtual Community

The first virtual community can be found in early 1980s, called USNET. That is a kind of network which can connect the different computer center in the corresponding university for information exchange [19]. Besides, it also allows people who establish the news group in the network and comment or post the information on them [19]. It is clear that the various news groups will lead to the emergence of relevant communities because members in certain group

may have not adequate interesting with outside. Nowadays, since the internet technology steadily progress, every person can use internet to exchange the information and share the experience. Therefore, many types of community have been found, for example, some communities are nested. It means one community can involve another [11]. Other type of community can be called overlap. That is the node can belong with more than one community in the network [11].

Summary

As described above, so many forms of community caused the difficulty in establishing the exactly mathematic definition. This research will concentrate on finding local community structure, the following parts focus on some strategies of detection community within large complex network.

2.2 Finding Community in Complex Network

This part aims to introduce community structure and algorithm of community finding. The traditional approach could be divided into two parts. One is clustering; the other can be called division.

2.2.1 The structure and property of community

Real network can be deemed as the vertex and corresponding edges constructed. The simplest one is the unweighted and undirected graph. Normally, a community is taken to be a group of vertices which have more neighbors inside than outside. These structures have important practical application if researcher wants to interpret the property or understand the network system. For example, a forum within World Wide Web may have a center topic and based on this information to construct the community.

In addition, the structure of real network can be divided into two basic types.

1. Coexistence: In this structure, the position of different communities can be regarded as equivalence [21].
2. Hierarchy: The community within this structure often includes some small communities [21].

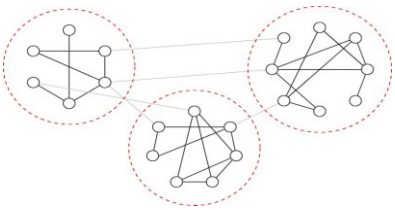


Figure 2.1 Community Structure

Taken from Ref [14]

This figure illustrates three communities denoted by the circle. It is clear that the amount of the edges in the community is more than outside.

Furthermore, the main task of finding the local community is to extract the accurate information of community structure from large network based on limited information. Since Newman converts the community finding into the physical problem or mathematic problem, the relevant topics has become a research hotspot [5]. Current community finding algorithms almost based on the clustering such as the hierarchy clustering algorithm, G-N algorithm, and likelihood clustering algorithm [5, 7]. However, there are two main problems of the community finding. Firstly, it cannot estimate how many communities within the network before explore it [12], which is typical. The other is the overlap problem. The node's position is uncertain. However, this problem has been solved by Gregory, Steve [33].

2.2.2 Clustering

Clustering is a set of assignment that can divide the different element into corresponding categories [16]. The element within subset often emerge the similar feature. Clustering is a method of unsupervised study; it is a classic problem in machine learning aspect. In the modern society, clustering has been widely applied in data mining, pattern recognition, and finding community in social network analysis [7]. In fact, clustering is an efficient approach to find the community, especially within hierarchy clustering.

As for the clustering, there are two main features of the each element in the certain cluster.

- The element within different cluster always emerge the different features and vice versa.
- The distance between two elements which have located in the same cluster always less than located in different cluster.

[1]

The distance is often applied to represent the similarity of two elements. There are two main approaches to measure it.

Oji Distance: $d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+...+(x_{in}-x_{jn})^2}$ (2.1)

Man Distance: $d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{in}-x_{jn}|$ (2.2)

[16].

2.2.3 Overview about the Clustering Algorithm

Clustering involves many algorithms that can be divided into partition, hierarchy, model or density.

- Partition Clustering:

Suppose we have the data set D including n elements and the amount of categories Y are require to be established, the partition approach will divide the n elements into the K categories based on its similarity.

One famous algorithm called K-means clustering. First of all, it randomly chooses K elements,

so each element can be defined as the center point within relevant categories [16]. Then, to calculate the distance between the center points to other elements and allocate them into the nearest group [16]. In addition, calculate the new mean within categories until convergence. This equation can represent this value [16].

$$E = \sum_{i=1}^{k} \sum_{p \in G_i} (| p - m_i |)^2 \qquad (2.3)$$

- Hierarchy Clustering:

This method is used to divide the dataset into the tree called dendrogram [6]. Based on the direction from up to bottom or opposite, the hierarchy clustering can be classified with agglomerative hierarchy clustering and divisive clustering.

- Agglomerative hierarchy clustering means that every element can be considered as the different categories, then merge the similar elements based on its likelihood until every element join the relevant group or achieve the terminal condition [7].

- Divisive clustering is opposite comparing the previous one, all element can deem as the one cluster, and divide it again and again until every element become a single cluster or achieve the terminal condition [7].

In this aspect, the famous algorithm can be called chameleon. The each process describes below.

First of all, it divides the complete graph into some small segments, then merges the similar cluster as we described above. This picture can present the each process.

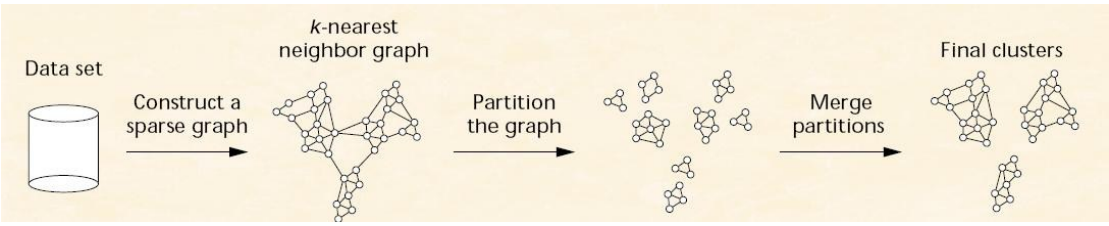


Figure 2.2 Chameleon Algorithm

Taken from the book: Chameleon, Hierarchical Clustering Using Dynamic Modeling

Within chameleon algorithm, cluster's similarity is based on its connectedness and closeness [6]. That is also to say, if the two categories have high connectedness and very close, thus these two groups will be merged [6]. In other words, chameleon can provide a dynamic approach for adaptive changing within network.

- Density Based Clustering

The approach of density clustering can find any shape of the cluster [4]. In the sample space, the high density area is separated by the low density area. The famous algorithm named DBScan (Density based- Spatial Clustering of Application with Noise).

DBScan require two parameters, ε and $MinPts$. ε refers to the radius of the object [4]. If (p, q) $\in \varepsilon$ and p is the central point within object, thus the point q can be deem as the directly density reachable from the point p [4]. The basic process starts a random choosing a point p that has never been visited [4]. If the amount of neighbors upper or equal the $MinPts$, then the point p and its neighbors can become a cluster [4]. DBScan constantly do these work until all of the points has been allocated within cluster.

Summary

Clustering is an efficient approach to detect the community structure, especially with the hierarchy clustering. In fact, this approach has been widely used by many sociologists to analyze the property of social network. However, as described above, some problems of community detecting have not perfect solution. Following part will describe some famous algorithm of community finding and it also based on the clustering.

2.3 The algorithm of community detecting

This part firstly pays attention to some famous algorithms for community finding in the large complex network, for example, K-L, G-N, and Radicchi. Then, it will focuses on we local modularity algorithm in details due to it will be used for community finding in the subsequent chapters.

2.3.1 Kernighan-Lin algorithm

It is a heuristic algorithm that can be used for overcome graph partitioning problem [8]. The running time approximately O (n^2) [8], where n is the number of the vertex within network.

To run the K-L algorithm must define two sub-graph first, so it need to create the efficiency function denoted by Q, then using the amount of edges intra-subgraph subtract the amount of edges inter-subgraph. This process can be divided into two steps.

1. $Q_a = I_a - E_a$. I_a is the internal amount of edges and E_a is the outsider amount of edges [8].
2. Exchanging the point pair, then calculate the reduction as the $\Delta Q = Q_a + Q_b - 2C_{a,b}$. The C can be denoted the cost between a and b [8].

The advantage of K-L algorithm is the running time. The disadvantage is the two sub-graph's size should be defined before calculation the relevant parameters. That is the reason K-L algorithm is difficult to apply in the real network [2].

2.3.2 G –N algorithm

This method is given by Girvan and Newman. The traditional approach of community detecting tries to find the partitioned information (boundary) but not require the pre-knowledge of the community's size [5]. G-N algorithm can satisfy these properties.

According to the hierarchy clustering, it is based on the weight edge and allocated them into the initially network, then starts from the strongest edge to the weakest edge. That is the reason the edge in the center of the network has the biggest weight and vice versa. However, The G-N algorithm is opposite. The edge betweenness has been introduced to measure the information flow [12]. The vertex betweenness we described above, the definition of edge betweenness is similar with that, the amount of shortest path between pairs of vertices that run through it [5]. That is also to say, if it is possible to find the edge which has been undertaken the large communication among the community, that edge is probably connect between two communities. Therefore, remove them can easily get the community. This process starts to calculate the every edge's betweenness and remove the highest one, until no edge remains [12].

However, there are two drawbacks with the G-N algorithm.
Firstly, it cannot measure how many communities involved in the network [18].
Secondly, inefficient, the worst case of running time is O (m^2, n) [12]. Where m denotes the amount of edges and n denotes the amount of vertices [18].

2.3.3 Radicchi Algorithm

This approach is similar with the G-N algorithm but not base on the edge betweenness. The new index called edge clustering coefficient have been introduced. Comparing running time, this algorithm is faster than G-N approach, O (m^4/n^2) [18]. This algorithm considers the triangle loop, three edges and the path are closed. The reason is the amount of edges which connects the different communities is significant fewer than connects the vertex within community.

The edge clustering coefficient can be defined as $C_{ij} = z_{ij} / \min(k_i - 1, k_j - 1)$ (2.4) [18].

k_i, k_j is the vertex degree of i and j, Z_{ij} denotes the amount of the triangle loop involved these edges within the network. The fundamental idea with this algorithm is remove the edge which has the lowest clustering coefficient and iterative this step until no edge remains [18].

However, the drawbacks within this algorithm are also distinct, because it always depends on the triangle loop in the network. Fortunately, within social network or real network, the number of this triangle is not fewness comparing with other type of network. But within large complex network, the number of edge is significant higher. That is the reason why this is not efficient approach due to the running time is not desired.

Summary
These algorithms which described above have some disadvantages such as the pre-knowledge requirement, for example, the size or amount of the communities, especially within large complex network, these information is unknown and hardly to get. In the following part, it will describe a local modularity algorithm that can overcome these problems and its running time is also acceptable.

2.4 Finding Local community

2.4.1 Modularity

As described above, current approach cannot solve the problem of pre-knowledge, such as the number of communities within large network. Within G-N algorithm, Newman defines modularity to measure the community. It depends on the number of edges within community to minus the expected value of within random network which has the similar quantity.

Normally, the modularity is an important index to measure the quality of the community. For example, if we can divide the network into K communities, then the systemic matrix E as $k \times k$ could be gained. Each of the elements E_{ij} means the number of vertex's edge from the community i to the community j divided by number of all edges within network. Thus, this type of graph is indirection. It can obtain the equation such that: $Tr\ e = \sum_i e_{ij}$ [12]. TR e is the number of edges within the community divided by the total number of edges within network [12]. It is clear that if the TR e is significant high, the quality of community is also well. However, this index is not appropriate because once all of the vertices belong with one community, the TR e is 1. Therefore, it cannot get information from the structure of community.

Therefore, Newman redefines the measurable approach:

$$Q = \sum_i e_{ii} - a_i^2 = TR\ e - \| e^2 \| \qquad (2.5)$$

Where $\|e^2\|$ denotes the amount of element within matrix e, if the number of edge in the community approximately equals to the number of edge in randomly network, the Q equal 0 [12]. Other condition, if the Q achieves 1, then it reflects the strongest structure of community [12]. In fact, many of the value Q fall into the 0.3 to 0.7 [12]. The high value is rare [12].

However, although this method is a standard approach for measuring the quality of the community, it also has some limitations. For example, the pre-knowledge of the entire network requirement, considering the World Wide Web, which is too large and too dynamic to ever be known fully. Thus, local modularity has been introduced to measure the quality of the network.

2.4.2 Local Modularity

Due to some drawbacks within previous approach, this method can overcome the problem due to prevent exploring the entire network.

Given a network denoted as G, the C denotes some vertices and we have favorable knowledge of the community "C's" connection, the whole network lacks some necessary information. Based on C, a series set of neighbor vertices denoted by U could be obtained. Then an

assumption that access the area C only through the set U (his neighbor) is made to get the adjacency matrix [3]. Therefore, the only way to gain additional information about graph G is to explore some neighbor's vertices in U. As a result, vertex $v_i \in u$ will be removed and become the member in C. This one vertex at one step work can cooperate with web crawling to explore the WWW [3]. Figure 1.3 illustrates this process below.

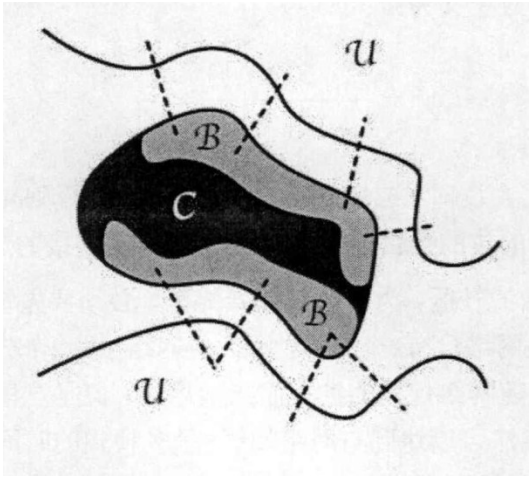


Figure 2.3 Community Structure

Taken from Ref [3]

Considering $A_{i,j}$ denotes the element within adjacency matrix, there have the information present below.

$$A_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are connected,} \\ & \text{and either vertex is in } C \\ 0 & \text{otherwise.} \end{cases}$$

Fig 1.4 Adjacency Matrixes Taken from Ref [3]

If the set C can be deemed as the community, thus to measure the community we need only calculate the proportion of adjacency vertex within C. The relevant index can be quantified by the equation shows below [3].

$$\frac{\sum_{ij} A_{ij}\xi(i,j)}{\sum_{ij} A_{ij}} = 1/2m^* \sum_{ij} A_{ij}\xi(i,j) \qquad (2.6)$$

With the function $\xi(i,j)$, if both i and j belong with the C, then it equals 1, otherwise, it equals 0 [3]. Besides, $m^* = 1/2\sum_{ij} A_{ij}$ denotes the number of edges within the adjacency matrix [3]. It is clear that when C has numerous connections with inside and the connection of outside is opposite, the corresponding value is high. Especially with the condition of ||C||>||U||, the partition often

appear to be good [3].

If concentrating on the vertex which belongs with C but at least one connection to the set U thus this is the boundary vertex of set C. If the quality of community is significant high, that the connection with inside should more than outside. Based on this information, an adjacent matrix of edge can be defined as below.

$$B_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are connected,} \\ & \text{and either vertex is in } \mathcal{B} \\ 0 & \text{otherwise.} \end{cases}$$

Fig 1.5 Boundary-adjacency matrix Taken from Ref [3]

Therefore, the local modularity could be represented as below.

$$R = \frac{\sum_{ij} B_{ij}\xi(i,j)}{\sum_{ij} B_{ij}} = \frac{I}{T} \qquad (2.7)$$

When $v_i \in b, v_j \in c$, so $\xi(i,j)$=1, otherwise is 0. I is the number of edge which has no connection with set B, and T is the number of edge which has connection with set C [2, 3].

2.4.3 Problem Definition

From previous part, the community in the large complex network was discussed. The previous algorithms often concern the global community structure in network [3]. However, it requires the graph be completely known [3]. As well-know, community has many types. Such as the overlapped community, it means the vertex belong with more than one community. Furthermore, in the real network, a large community may include many of the small communities, which are nested. Finding local community means to detect the community to which particular vertex belongs. It is useful to understand the property of real network. Another example can be presented as the Dunbar's number. It refers to the people's relationship in the real society that has stable quantities [22]. The number is 150, it means that one person within society normally has 150 relationships to the other and this relation tends to constant [22]. About the large complex network, this phenomenon also exists. That is also to say, a group of vertices has stable edge and its structure may not be affected by the network changing because dynamic is usual case within real network. In addition, about search engine system, if indexing or clustering web pages depends on the text content without considering its linking structure, the result is probably bad for queries [3].That is the reason why measure local community is worth to research.

2.4.4 The Algorithm for finding local community

This algorithm has been developed by Aaron Clauset. Comparing with the G-N algorithm based on the partition graph, this approach depends on the agglomeration. The time complexity is O (k^2, d) [3]. Where k is the number of vertex and d refers to the mean degree

[3]. That is the reason this method is more suitable than the previous methods to detect the community in large complex network.

The relevant process begins starting vertex which will be added it into set C. Furthermore, add the neighbor vertex into set U. Besides, within each step, it add the vertex from set U to set C that can leads the local modularity increase until the community scale is adequate or no vertex remains [3]. The corresponding steps presents below [3].

Input: graph G, start vertex v_0
Output: community
Add v_0 to C
Add all neighbor of v_0 to U
Set B=v_0
While (|C|<K) do

For each $v_j \in u$ do

Compute ΔR_j

End for

Find v_j such that its ΔR_j is maximum

Add that v_j to C
Add all new neighbor of that v_j to U
Update R and B
End while

$$\Delta R_j = \frac{x - Ry - z(1-R)}{T - z + y} \tag{2.8}$$

Where T is the number of edges in the boundary region, x is the number of the edge within T which has been terminated by the v_j. y is the number of increasing edge when v_j has been added into T. z is the number of edge which should be removed from the T [3].

This method's running time is $O(k^2 \times d)$ [3]. Where k is the number of vertex which must be visited during community detection; d is the mean degree of vertex. About the sparse graph, the running time is $O(k^2)$ [3].

2.5 The history of social network analysis

This part describes the history of the social network analysis and illustrates some milestones.

2.5.1 What is the Social Network Analysis

Social network has a long time history. In 1968, Allen Barton, a professor of Columbia

University, describes the main research area of social network analysis. He wrote that "*for the last thirty years*, *empirical social research has been dominated by the sample survey. But as usually practiced, using random sampling of individuals, the survey is a sociological meatgrinder, tearing the individual from his social context and guaranteeing that nobody in the study interacts with anyone else in it* "[10]. Barton's statement is justified even today. The main area of social network analysis is concentrate on the clustering on the behavior of individuals [10]. That is also to say, Social network indicates the social actors and their relationship. Some researchers focus on the relationship between each node rather than the nodes itself. This kind of research that examines connection between each objects of study is named structural [10].

2.5.2 The landmark of social network analysis developing

In the early of 1920s, Gustave LeBon described the phenomenon of crowd behavior. He states that if the individuals join the crowd then they will lose their own identities [10]. Within the crowd, people often like imitate those around them, and theirs idea or behavior also may diffuse over that area [10]. Furthermore, about the graphic imagery, the earliest research concentrates on the kinship [10]. This picture can show the relationship of kinship with ancient roman, data collected by Lewis Henry Morgan, although the researcher has not collect an adequate data, but it point out the position of equivalent relatives. The corresponding graph presents below.

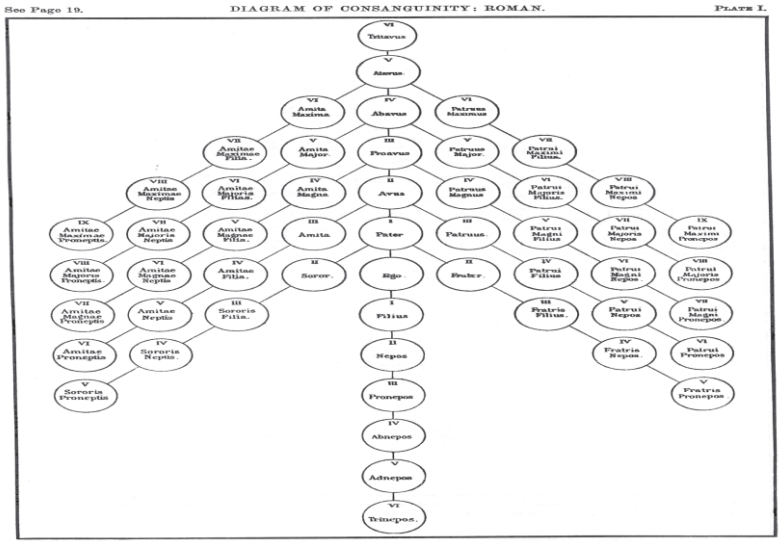


Figure 2.4 Morgan's Descent System of Ancient Rome

Taken from Ref [10]

In 1930, Whyte, a student of Harvard University, write a book named "street corner society" [10]. He states the social structure of the community by observation interaction patterns within its citizens, and it fully describes that structure [10]. In this book, he used some graphs to depict the structural position [10]. The relevant picture shows below.

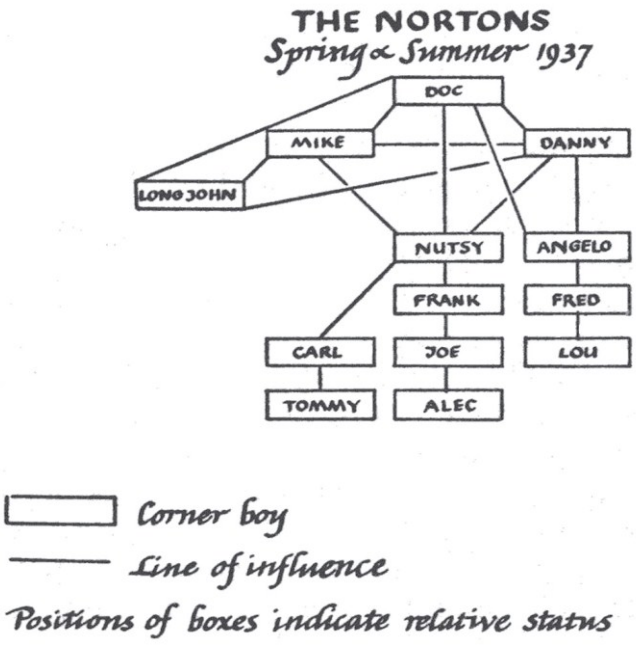


Figure 2.5 the Structure Position of People's Status

Taken from Ref [10]

In the period of 1940s to the end of 1960s, Bavelas and his research group developed a formal model to express the social structure and collect the experienced data [10]. Meanwhile, the graph theory has been introduced within social structural analysis. This graph can represent the pattern of communication.

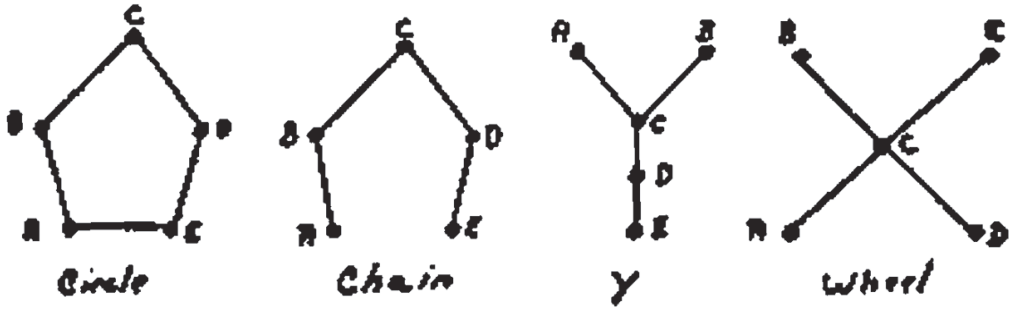


Figure 2.6 the Experimental Patterns

Taken from Ref [10]

In the end of 1950s, as the structural research has been constantly developed, some new research groups have rapidly established during this period. The famous people within this group named Claude Flament, he worked at the laboratory of experimental psychology at the Sorbonne in Paris [10].

In 1963, he published a book named "application of graph theory to group structure" [10]. The book combines two approaches to both communication research and structural balance and include graph theoretic and algebraic models of structural phenomena [10]. That is the

reason why Flament's theory can be prevailed even until today.

After the 1970, Harrison Colyer White made a fantastic contribution with this research aspect. His work has significant effect with the modern social network analysis. However, his greatest contribution is a teacher due to introduce a whole generation of Harvard student with a structural perspective [10].

2.5.3 Different Perspectives of Social Network Analysis

Nowadays, social network analysis is a distinct research area between social and behavior science. The reason is that social network analysis is based on the assumption of importance of relationships among individuals [21]. For example, in the biological area, social network can be represented by the different biotic population and theirs relationship. To study this structure can improve the reorganization of ecosystem in the current area.

From the sociology perspective, social network is consisted by a set of people and theirs relationships. As the computer science begin to rapidly develop, the method of data mini has been introduced within social network analysis. From this perspective, social network is represented by the graphic which consisted by a set of data. In addition, the entity often be represented by the node and its interaction have been expressed by the line within the graph [21]. Normally, graphic theory, probability and statistical often have been used to analyse social network [10].

Recently, these approaches have become the standard research areas; That can be defined within four classes below [10].

- It relies on the ties link to each social actor.
- It relies on the research' experience or systematic empirical data.
- It depends on the grapy theory.
- It depends on the mathematic or computational model.

2.5.4 Famous experiment of social network

Social network can reflect the phenomenon of small world. In 1967, Stanley Milgram, a sociologist in Harvard University, conducted a famous experiment that tracked the chain of acquaintance in United States [10]. In his experiment, 160 people have been randomly chosen to give the packages and require them to forward to a friend whom they thought would bring the packages near the target people. Finally, approximately half of the mails send to the individual successfully [10]. Based on these data, Stanley Milgram has established the theory called "six degree of separation" [12]. It means that every people can find a stranger at most six steps on earth. Figure 2.4 can present the structure described above.

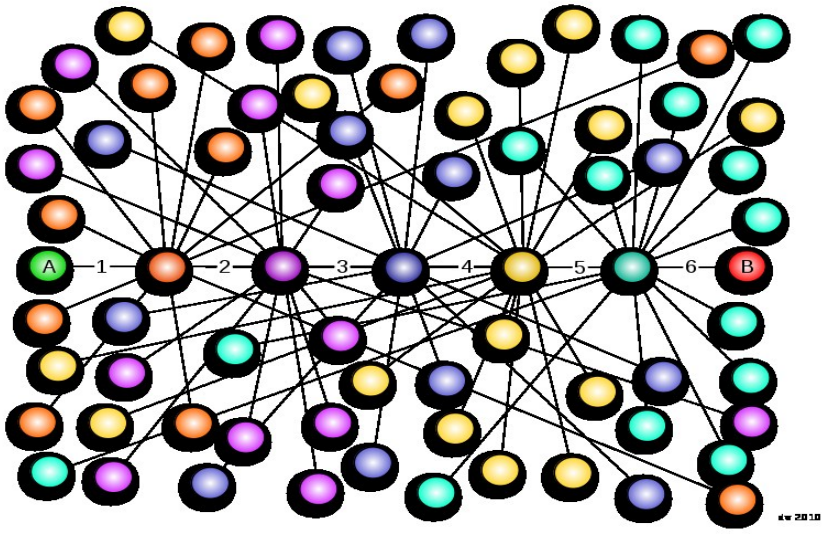


Figure 2.7 Six Degree Separation

Taken from Gurevich, M (1961) The Social Structure of Acquaintanceship Networks, Cambridge, MA: MIT Press

However, Milgram's theory received many criticisms. For instance, other researcher unconvinced that the degree merely achieve six due to he cannot track adequate amount of packages and as a result [20]. Besides, although the six degree separation reflects the relationship between people but it cannot consider the weight of each node [20]. It means each node has their own importance within social network because people will make many friends in our life. But some of them is not important and vice versa. That is the reason why merely consider the links is not accurate to measure the property of node in the network, especially with the human relation network. Furthermore, in Milgram's experiment, consider the process of letter transferring, whether everyone still likes to do that work, or the cost during the letter transferring have been ignored [20]. On the other words, there may have some people who do not like to forward the mail due to their own problem, such as lost the address or alienated friendship. This phenomenon can be called damping [21]. Nonetheless, damping also has some advantages for information flowing. It could think about that if there is not damping during information exchanging, that is also to say people can exchange the information to every person based on the six degree separation. That is a definitely horrible situation because people cannot undergo the huge information requirement. That is the reason the six degree separation has some drawbacks because it gives the exceeded assumption of equivalent importance of each node and the cost has been ignored.

Summary

From this part, it describes some important periods of development of social network analysis. For example, the period from end of the 1950 to 1970, many of the relevant concepts have been formed during this time. Besides, it is also illustrates many of the social network analysis approaches and relevant perspectives because some of parts will be used in the following research. Finally, it interprets the "small word" phenomenon and describes the theory of six degree separation because it has some relations with the community finding which will be described below.

2.6 The application of social network analysis in modern society

This part describes the three typical applications within social network analysis. They are antivirus application, terrorist detection and international relationship evaluation.

2.6.1 Social network application within medic science and computer virus

In the model medic science, the one research hotspot is finding the route for virus diffusion and its structure, especially with some unknown virus. Therefore, due to track and identify theirs behavior, to establish the platform based on social network analysis is an appropriate method [20]. Meanwhile, it is also benefit to detect the importance of each node within network. For example, through the social network analysis, it could provide information of virus's diffuse route and its scale in the crowd. Therefore, medical staff can vaccinate to the group of people which have been defined as the virus susceptible. Thus, experts can make some strategies for reducing the diffusion of virus. It is clear that these problems probably can be overcome by social network analysis.

Furthermore, in terms of computer virus detection, social network analysis may give the new approach to detect and prevent them. For example, in the zombie network, hacker can issue the command from control node and attack the object through the zombie network, such as Slapper, a kind of DDOS attack based on the P2P framework [22]. However, some traditional approach is inefficient to prevent that due to the P2P network has not center node or relevant structure [22]. That is also to say the node within P2P network is random distribution. That may lead an enormous barrier to detect or defend attack. However, based on the social network analysis, that problem may be overcome. The fundamental idea is finding some important node, bridge node and key connections in zombie network, then monitor or remove them if it is necessary. Finally, the researcher can divide this network into many small fragments for prevention the attack to arrive the target [22].

2.6.2 Detection the relationship among Terrorist

After the September 11 attack, the network which can link each terrorist has been concentrated by many researches [17]. Although to clearly understand the whole network structure is impossible during a short time, but once it can reduce the sample space, the clear relationship can be extracted.

Figure 2.5 presents the four terrorists and theirs relationship. Based on social network analysis, researcher can find the key entity within network, and then try to monitor and destroy it because it may lead network disruption. In addition, according to the dynamic changing of the network, researcher can forecast the potential problem in the network, such as the people who may become the terrorist [17].

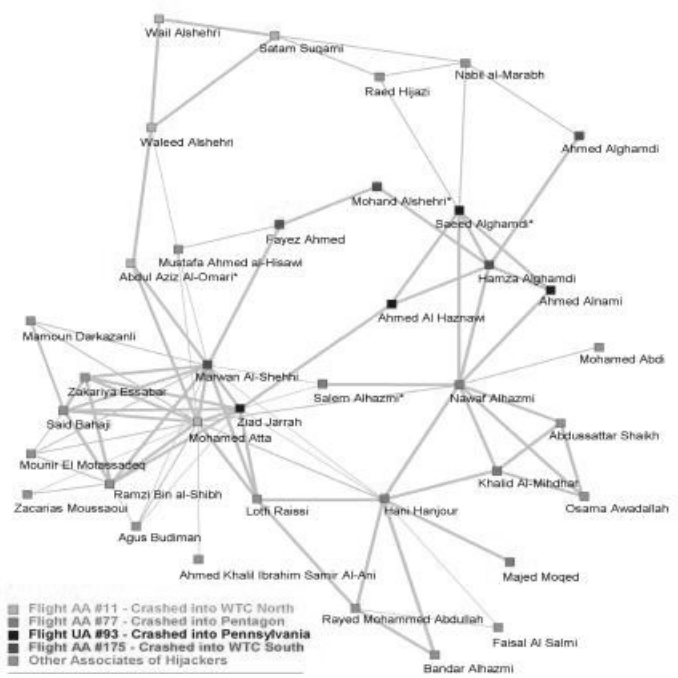


Figure 2.8 the Network of Relationships among Terrorists in 911

Taken From Ref 17

However, this type of network is a little difference compared with the normal one. First of all, as described above, the network is dynamic, not static. The second one is the fuzzy boundary due to the node is hardly belonged within a certain community of the network. The last one is incompleteness because there also have many nodes is invisible so it probably not be involved in the network [17].

2.6.3 The Application of Social network analysis within International Relations

In the research of social network analysis, a country can be deemed as the node, and the interaction among them from the politic, economic and culturals' cooperation can be regarded as the tie. That can construct a network to represent the relationship among them. This picture illustrates the relation among fifty-one countries below.

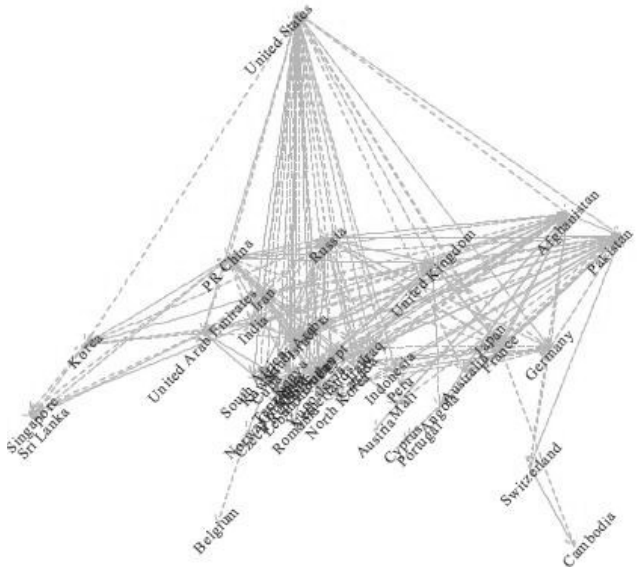


Figure 2.9 International Relationship among Fifty-one country

Taken from Taken from Ref [11]

Through this network, it could be clearly observed that the closeness among these countries. In addition, based on this information, government can make a strategy of economic or political aspect to other countries. Furthermore, the potential connection also may be found within this graph. That is the reason why social network analysis has significant effect with international relation.

Summary

Researcher states three applications in the current situation. Through this description, it is clear that social network analysis has been widely used in our life and worthiness for further studying.

2.7 The Object of Social network analysis

This part describes the two main approaches with social network analysis. Depends on the different perspectives, it can be divided into the relation based approach and position based approach [21].

2.7.1 Relation based approach

This type of approach based on the relationship among actors in the network. There are some important properties that can be used to measure the network.

Between-ness: This parameter refers to the degree an individual lies between other individuals in the network [15]. In other words, it can be deem as the node which locate between other nodes in the network but there have not directly connection to each other [9].

Closeness: This parameter refers to the degree an individual is close to other individuals in a

network [15]. It can be calculated by the inverse of the sum of the shortest distance between each individual in the network [21].

Density: Density can be defined with the proportion of ties with group of actors in network relative to the total number of ties [15]. That is also to say, the more amount of relations the group has, the high density degree the group emerges, and vice versa.

Direct and Indirect: Direct refers to the relationship between social actors [15]. Indirect refers to establish the connection that needs at least one middle actor to forward the information [15]. It is clear that if there have many middle actors that the closeness between them is weak.

Centrality: It refers to amount of ties from itself [15]. Normally, if the actor has many links to the other thus it is more important within network. The more links the actor has, the more importance he possess. However, the ties are not only index to measure the network property because the actor may locate anywhere. Especially in the edge of the network, for example, even if some nodes have more links to the other, but it may less important than bridge node. That is the reason why position based approach also should be seriously considered.

2.7.2 Position based approach

In comparing with the relation approach, this approach concentrates on the structure of network. If the relation based approach can be deeming as the social cohesion, then the position based approach can be considered as the structure similarity [15]. There are some important indexes that can measure the network.

Structure Equivalence: If there are at least two or more than two groups of actors, and their structures are similar with the third one that can be called as the structure equivalence [15]. In the social network, the two equivalent nodes must have the same structure [15].

Position: It refers to a group of actors or nodes that have the similarly structural position [15]. It means that this approach will ignore the effect of the signal actor in the network. In other words, it is not important to identify which node located in this position; it focuses on the situation of this position within the whole network.

Summary

Social network analysis can be described about the relationship among social actors. There have two types of approaches we described above. First one is the relation based approach, depends on this perspective, the community may have been constructed due to it has more edges between each vertex than outside it.

Secondly, position approach normally consider to the social patterning between each actor who has the same position in the network structure. It often concentrates on the structural equivalence to describe the actor's behavior.

2.8 Type for Social network analysis

This part describes two classical types of the social network analysis. First one is complete network and also be called whole network. Second one is ego network and also be called personal network [21].

2.8.1 Whole network

In this area, currently it is focus on the relationship in the small group of actors. Linton Freeman is a famous researcher within this aspect [21]. As the time begin to progress, the structure of the network or internal relationship probably be changed. Based on the data collection method, the social matrix and graph have been used to analyse whole network. For example, considering social matrix as an N*N matrix. The element within that either 1 or 0, N can represent the amount of people. Column in the matrix can express the people who need to do the experiment [21]. Row can express the people who have been chosen to do the experiment [21]. The matrix can be constructed by the both sides of people's interaction. Furthermore, about the graph, it can be constructed by the vertex (people) and edges (the result of interaction between each people) [21]. The corresponding centrality and prestige can present the property of that node (actor). Prestige refers to the amount of edges to other vertex. This method has been widely used in the area of social psychology [21]. Researcher can depend on the distribution of each node to classify them based on theirs role, such as super start, contacts, or isolates.

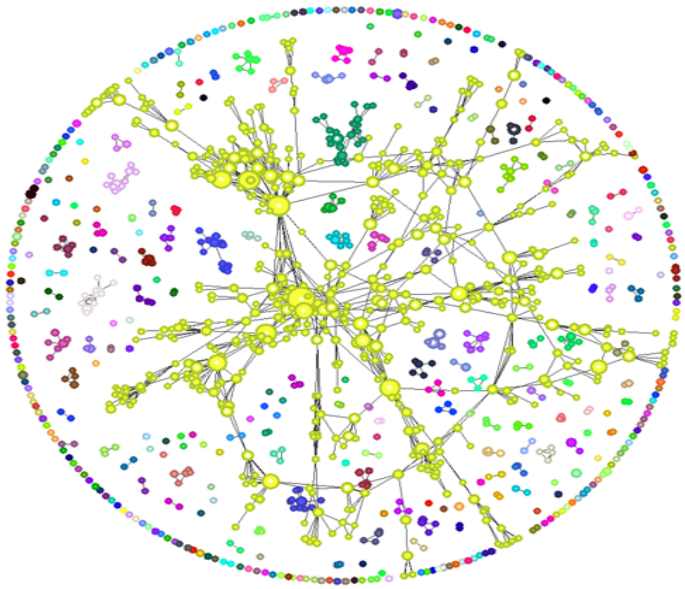


Figure 2.10 the Complete Network for Author's and Co-author's Relationship Network

Taken from Ref [21]

2.8.2 Personal network

This type of the network focuses on the individual behavior that has been affected by their social relationship. Mark Granoverter and Harrison White are two famous researchers in this aspect. According to the Van del Poel's research [21], the method of personal network

analysis can be divided into four areas, interactive method, role relationship, emotion, and social exchange. Emotion approach requires subjects point out the person who has the most closely relationship, such as the best friend. However, the disadvantage of that method is the evaluation from each person may give the different result. Social exchange theory can overcome that problem so it is widely used into current research. Especially with the people who have different background knowledge [21].

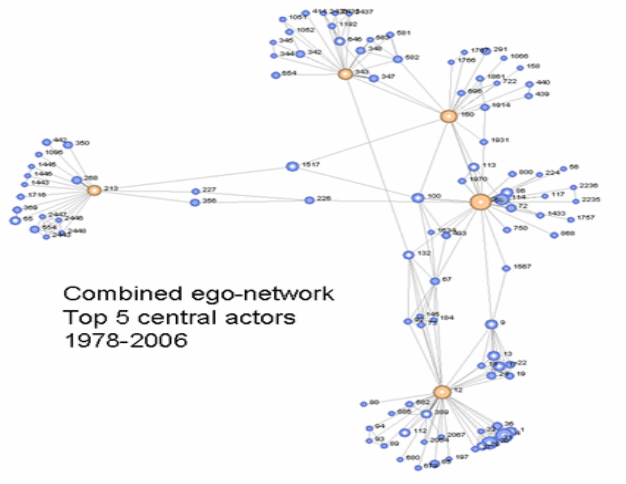


Figure 2.11 Combined Ego- networks

Taken from Ref [21]

Summary

From the previous states, the difference between whole network and personal network is the scale of the objected area. Whole network pay attention to the group of actors and alteration of theirs relationships. Comparing with the personal network, it emphasizes the individual's effect.

2.9 Mathematics in Social Network

In this field, two methods are common to measure the network. One is called sociogram and the other is called matrix algebra [15]. Furthermore, this part also states some important parameters that can be used to measure the network.

2.9.1 Sociogram and matrix algebra

Moreno, a sociologist in United States, provided a series of concepts about the sociogram [15]. Nowadays, it has been widely applied in social network analysis. Sociogram has been consisted by a vertex (actor) and its edges (relationship). A set of nodes can be represented with N, $\{n_1, n_2,,, n_g\}$. Based on the edges property, sociogram can be divided into directed graph and undirected graph. Moreover, it is clear and visual to exhibit a directed graph to express the relationship such as the personal power or loan information. For example, through the diagram below, it is simple to find that relationship between n_1 to n_2 and n_2 to n_1 is different.

This figure shows the relationship between directed graph and undirected graph and its adjacency matrix.

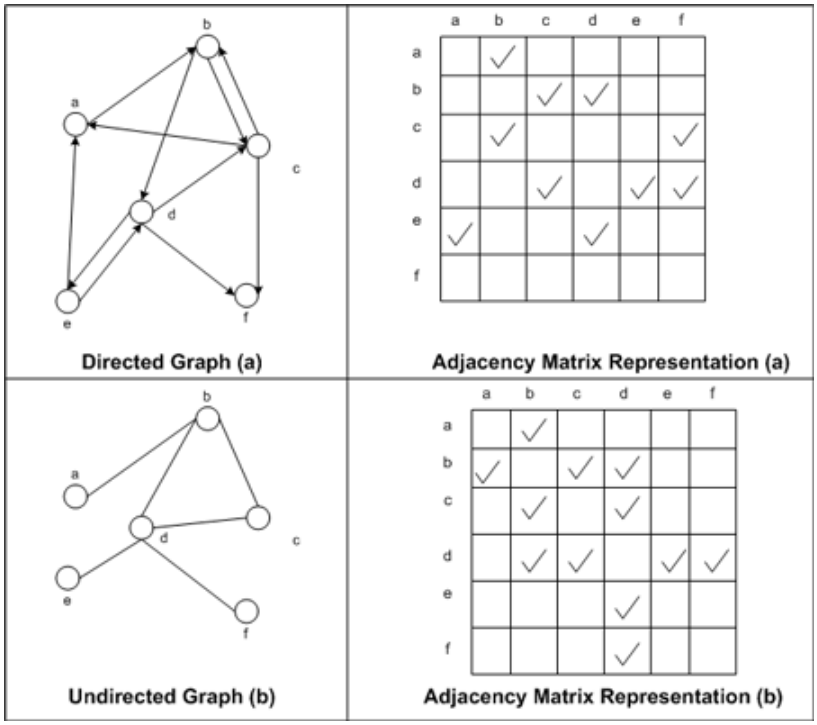


Figure 2.12 Directed Graph and Undirected Graph and its Adjacency Matrix

Taken from Ref [21]

Sociogram can clearly present the relationship among vertices. However, if the amount of vertex is too big, such as N >= 100, sociogram may not suitable to express that relation. That is the reason matrix algebra has been introduced in this aspect. It relates to the multiple relations and reflects its effect [21]. For example, researcher examines the relation between A and B is a friend, B and C is an enemy, so the relation of A to C can be represented as friend's enemy.

2.9.2 Metrics in social network analysis

- Vertex Degree

The degree of vertex in a graph is the number of edges that touch it [15]. Thus, vertex's degree can equal the amount of edges that connect it. If the two vertices have been directly connected, it called adjunct [15]. Otherwise, it named isolate [15]. Degree of vertex can reflect the actor's relationship within the network; it is also a part of the index to measure the centrality. Meanwhile, edge also has own weight, it means that the edge can represent the stronger connection or weak connection based on its effect [15]. In addition, based on the directed graph described above, the vertex degree can be divided into in-degree and out-degree. The in-degree refers to the amount of neighbor vertices which have directly connection to itself and out-degree is opposite. Therefore, within matrix, the sum of the row can be represented the in-degree, and the sum of the column can be represented the out-degree [15]. The below picture shows the relationship.

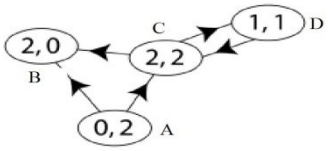


Figure 2.13 Vertex Degree within Directed Graph

In the ellipse vertex, the left number is in-degree, and the right number is out-degree. The adjacency matrix is asymmetric, means that the connection from C to D is independent with the connection from D to C [21].

$$\begin{pmatrix} (0)(1)(1)(0) \\ (0)(0)(0)(0) \\ (0)(1)(0)(1) \\ (0)(0)(1)(0) \end{pmatrix}$$

Then it can gain the out-degree $\sum_{ij} A_{ij} = \sum_{u \in V} \deg^{+}(v) = 5$ and in-degree $\sum_{ji} A_{ji} = \sum_{u \in V} \deg^{-}(v) = 5$

From this graph, the value of in and out degree is the same. Therefore, the equation could be presented as below.

$$\sum_{v \in V} \deg^{+}(v) = \sum_{v \in V} \deg^{-}(v) = |A|. \tag{2.9}$$

Normally, a vertex with $\sum_{u \in V} \deg^{+}(v) = 0$ is called source and a vertex with $\sum_{u \in V} \deg^{-}(v) = 0$ is called a sink [15].

- Geodesic

Geodesic is corresponding to the shortest path between two vertices because there probably have many paths that can touch them [15]. The distance is used to express the length of geodesic. Besides, if the two vertices have no path to touch each other, both for undirected and directed, the distance can be defined as infinite [15]. In addition, if the graph is not associated, thus it is at least a pair of vertices that length is infinite [15].

- Diameter

Normally, the graph has some geodesics. Thus the longest length is defined as diameter [15]. If the graph is associated, thus the diameter can be found and vice versa.

- Centrality:

This is the main area of the metrics in social network analysis. It describes the actor's or organization's power or their effect. In other words, it can be described as "who is the most important person in this network" [15]. However, this answer is too generally since it is

cannot tell us the exactly "important" mean.

According to A.Bavelas's research (Freeman 1979), if the social actor located within center of the network, his effect is more than located in the boundary [15]. Within the graph theory, it determines the relative importance of vertex within the graph [15]. Centrality measurement can be divided into four parts.

- Degree Centrality:

It is defined as the number of the ties that the node has been touched, so if the vertex has many edges touch to other, thus it probably has more influence [21]. That can implies a person who has more relationships tends to more power. In mathematic term, the degree k_i of vertex i can be expressed as $k_i = \sum_{j=1}^{n} A_{i,j}$. For a graph G = (V, E) with n vertices, the degree centrality $C_D(v)$ can be defined as $C_D(v) = \deg(v)/n-1$ [15] (2.10). Besides, centrality also can be used to measure the graph. Let the V has the highest degree in the graph G and G consisted by (V, E). G has the n node connection; and it can obtain the equation $H = \sum_{j=1}^{|V'|} C_D(v') - C_D(v'_j)$ [15] (2.11). H is maximizes qualities, so the graph centrality can be defined as $C_D(G) = \sum_{i=1}^{|V|} [C_D(v\ ) - C_D(v_i)]/H$ [21] (2.12). Within special case, such as the start graph, either for $\sum_{u \in V} \deg^+(v) = 0$ or $\sum_{u \in V} \deg^-(v) = 0$, where H can equals $(n-1)(1-1/(n-1)) = n-2$ [21]. Thus, the centrality of start graph is $C_D(G) = \sum_{i=1}^{|V|} [C_D(v) - C_D(v_i)]/n-2$ [21] (2.13).

- Eigenvector Centrality

Comparing with the degree centrality, the eigenvector centrality treat the edges are not equal [15]. It means the weight of edge has been considered in this index. Eigenvector Centrality measures the importance of vertex in the network [15]. In the social network analysis, it is reasonable to use the adjacency matrix to calculate that value.

Let $A_{i,j}$ denotes the adjacency matrix, x_I is the score of i^{th} node; therefore, we can get the equation: $x_i = 1/\lambda \times \sum_{j \in M(i)} x_j = 1/\lambda \times \sum_{j=1}^{N} A_{i,j} x_j$ [15]. N is the totally number of the vertices, M(i) is a set of vertices that are connected to the i^{th} vertices, λ is the constant [15]. Besides, if the equation is rewritten into matrix form, as $\lambda x = \mathrm{A} \times x$, thus x is the eigenvector and λ is the eigenvalue. Generally, the value of eigenvector centrality is positive; the amount of connection and its quality can be calculated by this parameter. In the modern society, eigenvector centrality has been widely used such as the Google Web pages rank [11].

There also have two centralities and all based on the network path. The path means traveling a series of sequence of vertices and following the edges from one to another [21].

- Betweenness Centrality:

It is associated with distance between one vertex to another [21]. For example, given the shortest path, one or more, and query for which path the vertex i lays. That is the reason this index is a part of the geodesic [15].

With the graph G: (V, E) include n vertices, the between-ness $C_B(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v) / \sigma_{st}$

(2.14), σ_{st} is the shortest path and $\sigma_{st}(v)$ is the shortest paths from s to t through a vertex v [21]. For undirected graphs is the (n-1) × (n-2)/n, and directed graph is the (n-1) × (n-2).

- Closeness Centrality:

The close-ness centrality can be described as how close between two vertices in the graph[15]. In other words, it represents the mean geodesic distance to the reachable vertex with the graph theory, $C_c(v) = 1/\sum_{t \in V \setminus v} d_G(v,t)$ [15]. In addition, in order to measure the different type of the network, Dangalchev modifies the definition for closeness that it can be used for measure the disconnected graph $C_c(v) = \sum_{t \in V \setminus v} 2^{-d_G(v,t)}$ [21] (2.15).

Summary:

This part concentrates on some mathematic theories for measuring the network. In generally, Sociogram often be used to analyze the structural equivalence and block model; and matrix algebra is able to analyze the actor's role and relationship [15].

Chapter III Implementation

3.l Program for Local Community Detection

This chapter will describe three main parts of implementation, crawler, local modularity algorithm and network reconstruction, writing program in java due to its property of platform independence.

3.1.1 Overall Design

The program has developed in three stages. First of all, crawler program starts the particular vertex (personal page in myspace) to explore the graph (myspace network) for detection the adjacent matrix which consisted by the personal friends. Secondly, local modularity algorithm will decide which candidate vertex can be agglomerated in the community. Finally, to present that vertex in graph interface for reconstruction the real network. This picture illustrates each step of agglomeration below.

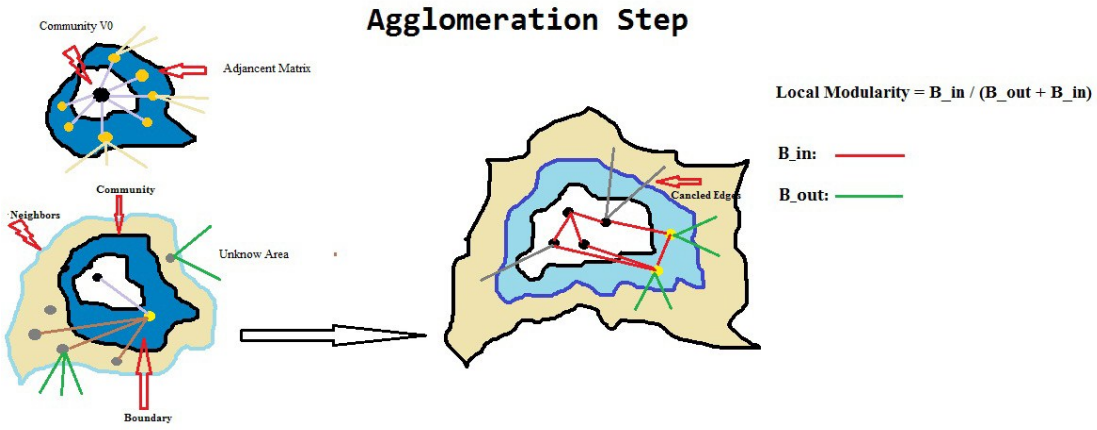


Figure 3.1 Agglomeration Step of Local Community Detection

3.2 Three Stages of Implementation

This part will describe some detail information within implementation step and analyze some problems which have been encountered.

3.2.1 Crawler and searching algorithm

Network crawler (spider) is a program that can fetch information from web based on a series of regulations. It is an essential technology of web search engine. Staring from the particular web page, crawler will constantly fetch relevant pages and store them until the terminal condition has been satisfied. Based on the similar idea, the crawler also needs to store

information from web but only focus on the information of personal friends. Therefore, regular expression will be used to filter useless information.

About the searching algorithm, crawler can be divided into two methods, breadth first search and depth first search. Based on current situation, breadth first has been chosen to embed with crawler. The reason is the requirement of detecting all information of the adjacent matrix that starting vertex belongs. Further information will be described in chapter 6.

3.2.2 Local modularity algorithm and boundary detection

Local modularity algorithm assumes the graph is unweighted and undirected. Therefore, the network as WWW, the property of direction should be ignored when implementing the program to explore it. In addition, local community detection algorithm is quite different contrasted with other regular method which required a global knowledge of the graph. Normally, local algorithm focuses on the agglomeration and other methods concentrate on the division. Figure 3.2 shows a simple structure of local community.

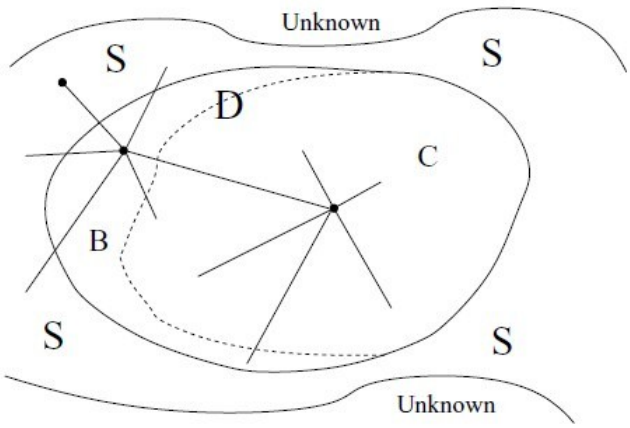


Figure 3.2 Local Community Definition

Taken from Ref [26]

In this figure, C can be denoted as the center community and B is the boundary. S represents the neighbors of vertex in community. At initial phase, it is assumed that start vertex belongs with the strong community thus R =1 when $C_{border} = 0$ [24]. R means the local modularity which described in chapter 2. Normally, the value of R lies on the interval $0<R<1$, the more big R occurred, the stronger of community structure can be presented. Furthermore, from the theory of local community, candidate vertex can be agglomerated often depends on its current R value and compare with previous one. The relevant equation is presented as below [26].

$$\Delta R = R^{'} - R$$
$$\Rightarrow \frac{B_{in} + x - z}{B_{total} + y - z} - \frac{B_{in}}{B_{total}}$$
$$\Rightarrow \frac{B_{in} + x - z(1-R) - R \times B_{total} - R \times y}{B_{total} + y - z} \quad (3.1)$$
$$\Rightarrow \frac{x - R \times y - z(1-R)}{B_{total} + y - z}$$

X is the number of edges from community and terminated by the patricidal vertex in the boundary region. Y is the number of increasing edges from boundary to neighbors. Z is the number of decreasing edges when the new vertex has been merged, because this agglomeration may affect the status of the boundary. For example, considering Fig 3.1, the author denotes vertex in the center community as A, so vertex B and C touch it. When vertex D has been merged in the community and touch B or C vertex, some outward edges (B or C belongs) which start the boundary to neighbors have been considered as the remove edges, thus center community will enlarge to involve the previous nodes B or C depends on which vertex has been touched. And then, this vertex will go into the central community. It means the end point of those vertices located in the community area.

Based on similar idea, if the new candidate vertex D touches starting vertex, its outward edges will be calculated as the increasing edges and no decreasing edges can be detected. After many times of agglomeration, the form of boundary will become more sharply and many of the vertices belong with the center community. It is clear that the number of edges inside probably more than outside because each step only vertex which has the smallest number of outward edges can be merged. Let us back into the equation 3.1, ΔR depends on the current value of R, increasing edges (Y) and decreasing edges (Z) depends on candidate vertex [3]. This figure shows those parameters below.

Figure 3.3 Boundary Status

3.2.3 Reconstruct the real network

While the new candidate vertex has been agglomerated in community, it should be present in the graph interface simultaneously. The author chooses Java Universal Network/Graph (JUNG) to visual data. It is a framework for the modeling, analysis, and visualization of graphs in Java [29]. Some properties of this framework present below [29].

- Support most type of graphs, for example, directed and undirected network
- Support many layout algorithms, for example, forced-layout, spinger-layout, and tree-layout.
- Support network event handing and user interaction, for example, transforming, picking, and adding or deleting nodes.

Springer-Layout algorithm has been chosen to visual data which collected by the program. When the new vertex joining, update the graph and repaint the interface for dynamic exhibition how the network construct. Besides, define the edge type for target the different information (user name) to the relevant vertex. In addition, some methods like graph controlling and user interaction also have been implemented in the graph user interface. For example, enlarge or reduce the graph, picking the nodes or transforming a graph.

3.3 Different Stopping Criteria

When vertices have been agglomerated in the community, local modularity algorithm needs an accurate approach to stop adding nodes because the real network is too large and too complex to ever be explored fully. Therefore, if the terminal condition cannot stop the program, the agglomeration of vertex will never stop.

In this part, two possible methods and its properties will be discussed. It is clear that an appropriate approach of terminal condition may directly affect the performance of community detection.

- *Scalability of the community*

Based on Clauset's theory, he defines K as the stopping criteria. It means the merging nodes constantly continue until number of vertices K has been satisfied [3]. However, this quantity is roughness due to it cannot reflect the essential property of community. In terms of the concept of community, the community structure means there exits an area which has more edges connect neighbors inside than outside. Therefore, if explored region of network lack of some community structures, the program may also has been terminated due to the scale of community has been satisfied.

Moreover, if the local algorithm fails to stop in time, some nodes which belong to other community are still identified as one community [26]. Therefore, the accuracy of the found communities is overstated. There have two figures presents this phenomenon below.

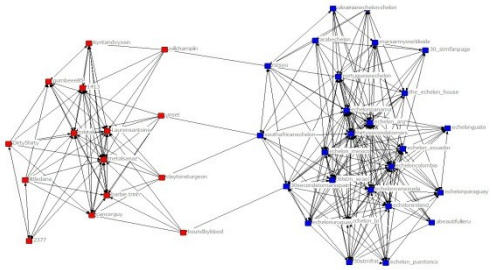


Figure 3.4 Program Stops in Time

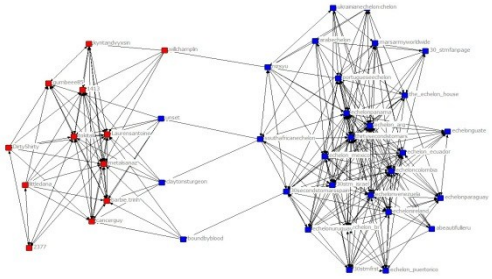


Figure 3.5 Program Fails to Stop in Time

- *P-Strong Community*

As described above, some drawbacks have occurred in the previous scheme. Thus the author introduces a new method named P-Strong community to replace it. First of all, strong community has been defined below. A subgraph $C \in G$ is a strong community when every vertex in C has more neighbors inside C than outside [26].

$$K_i^{in}(C) > K_i^{out}(C), \forall i \in C \qquad (3.2)$$

This equation can be applied as local stopping criteria. For example, program merges nodes until C becomes strong. However, this method is too strict that may not be able to directly use to detect community structure in the real network. Such as myspace network, because stopping the program depends on all nodes property, number of inside neighbors more than outside. Therefore, some nodes must fail with the equation (3.2). That is the reason why P-Strong community has been defined as below [26].

$$\sum_{i \in C} [k_i^{in}(C) > k_i^{out}(C)] \geq p\,|C| \qquad (3.3)$$

This equation means some nodes can achieve (3.2) but not all, just only a fraction p of nodes [26]. Depends on this theory, two equations are regarded as equivalence when p has been set 1, it is clear that this condition is more flexible than previous one because small p will lead the program stop early and vice versa [26].

Therefore, the author modifies program that each agglomeration doing the following way: define a counter and initially set it equals zero. When candidate vertex assigns the biggest value of ΔR , check its K^{in} and K^{out} values. If the condition has been satisfied, plus one to the counter, and these process will continue until value of counter achieves condition. It is clear that the higher value of P will lead stopping program become hardly. Besides, multiple value of P can be used simultaneously to test the classification of nodes since a community that is P_1 strong is also P_2 strong ($P_1 > P_2$) [26]. The following equation can be used to test P.

$$p_{eff} = \frac{1}{|C|} \sum_{i \in C} [k_i^{in}(C) > k_i^{out}(C)] \qquad (3.4)$$

Depends on this equation, P strong for all $P \leq P_{eff}$, and not P strong for all $P \geq P_{eff}$ [26].
In the real case, set different parameters {P} = {0.75, 0.76,……1} and perform the program with that stopping criteria [26]. Otherwise, the theory of local community can be used to

decide the best C_i because the smallest B_out and largest R can lead the graph separated easily.

3.4 Program Implementation

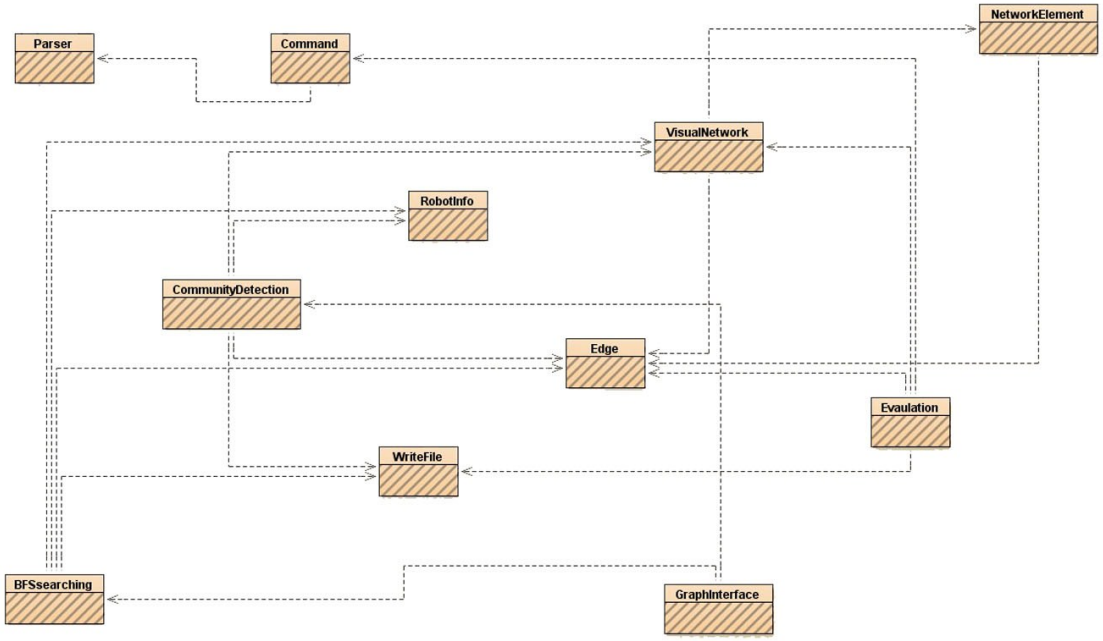


Figure 3.6 the Class View of the Program

Crawler and local modularity algorithm have been cooperated for detection local community, which written in CommunityDetection class. BFSsearching class has been used to compare the result with local method. Evaluation class has been used to test the program by explored the synthetic network, where it will describe in Chapter 5.

User Graph Interface

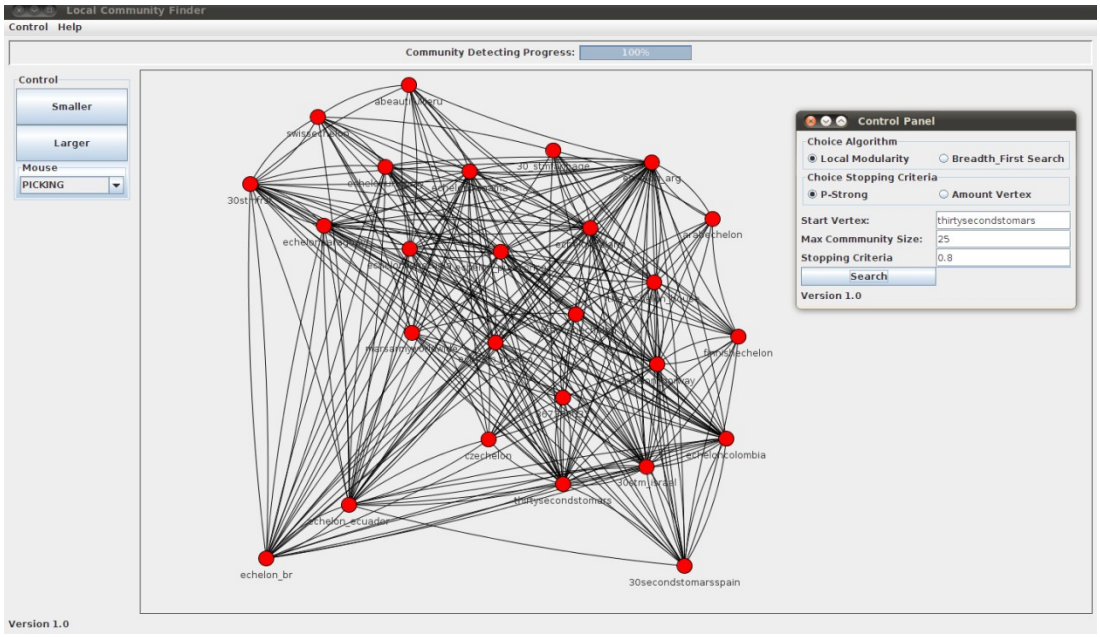


Figure 3.7 Graph User Interface of the Program

Summary

Within this chapter, some details information about the implementation has been introduced. For example, some key parameters, increasing and decreasing edges, initial status of the community and different stopping criteria. Those of them have a significant effect on the performance of the program. The next chapter will describe the result network given by the program.

Chapter IV Result Analysis with Local Community Structure in myspace

In this chapter, the result collected by myspace network will be discussed.

4.1 Community Structure in myspace

The author applies the program to explore myspace network and found the community structure shows below, starting vertex thirtysecondstomars. The stopping criteria have been set C = 25, P = 0.8. That is also to say, at least 20 vertices which have more neighbors inside than outside. Finally, 26 vertices have been fetched in the network. That means at least 6 vertices fails the definition of P-Strong community.

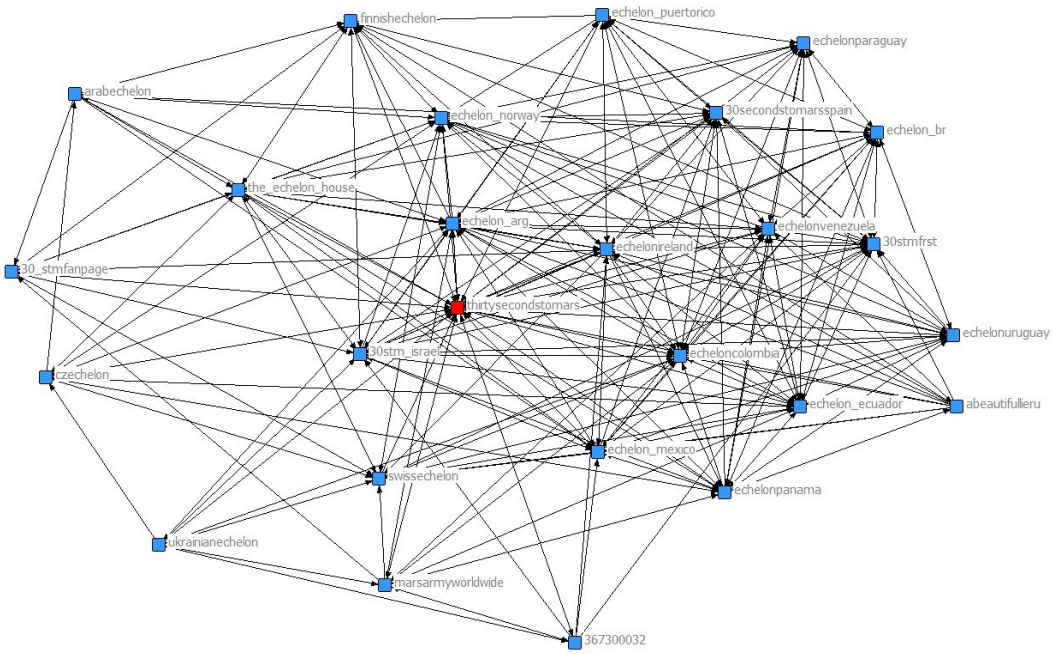


Figure 1.1 Community Structure in Myspace

Considering the definition of local modularity, the high value represents the good community structure. The figure 4.2 illustrates the local modularity as a function of the number of each agglomeration step.

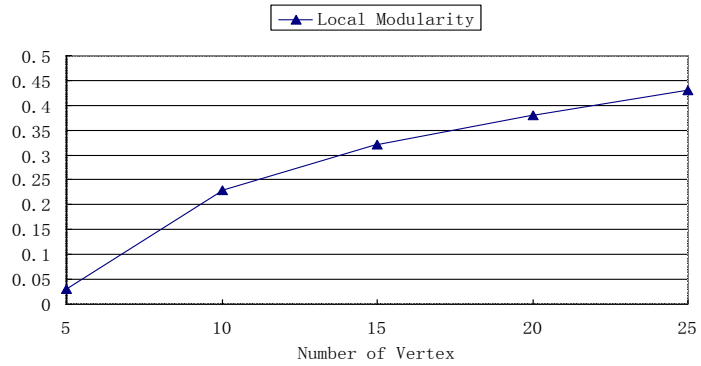


Figure 4.2 Local Modularity R for personal friends' information in myspace network

Besides, the stopping criteria have been tested as above. The accurate terminal condition implies stopping program when the first enclosing community structure has been found. Thus given the sub-graph S, S∈G, the best partition of the sub-graph will lead the high value of local modularity and the corresponding community structure also has high quality. In addition, the start vertex may directly affect the form of the result network. The reason is that the community which belongs with the first node has been considered as the strong quality and its boundary equals zero. Thus if the starting node is a hub or leaf, the result may totally different. Where the hub node means the high degree of that vertex and leaf node is opposite. The researcher chooses the vertex thirtysecondstomars, has the number of friends close to one million. Thus the degree of this node is significant high. That may implies this vertex located in the center of the sub-graph.

4.2 Result analysis using Social network method

The appendix A presents the property of all vertices in the community and each vertex denotes the personal account in myspace. It only focuses the relationship of friend's information.

The starting vertex thirtysecondstomars, has the biggest degree equals 25. The mean degree of the graph equals 14.538. Depends on experimental data, the degree distribution of most network in the real world is fairly right-skewed [3]. That is also to say, the large majority of nodes in the network have a low degree but small nodes are opposite. The figure 4.1 shows the degree distribution of each vertex in the graph below.

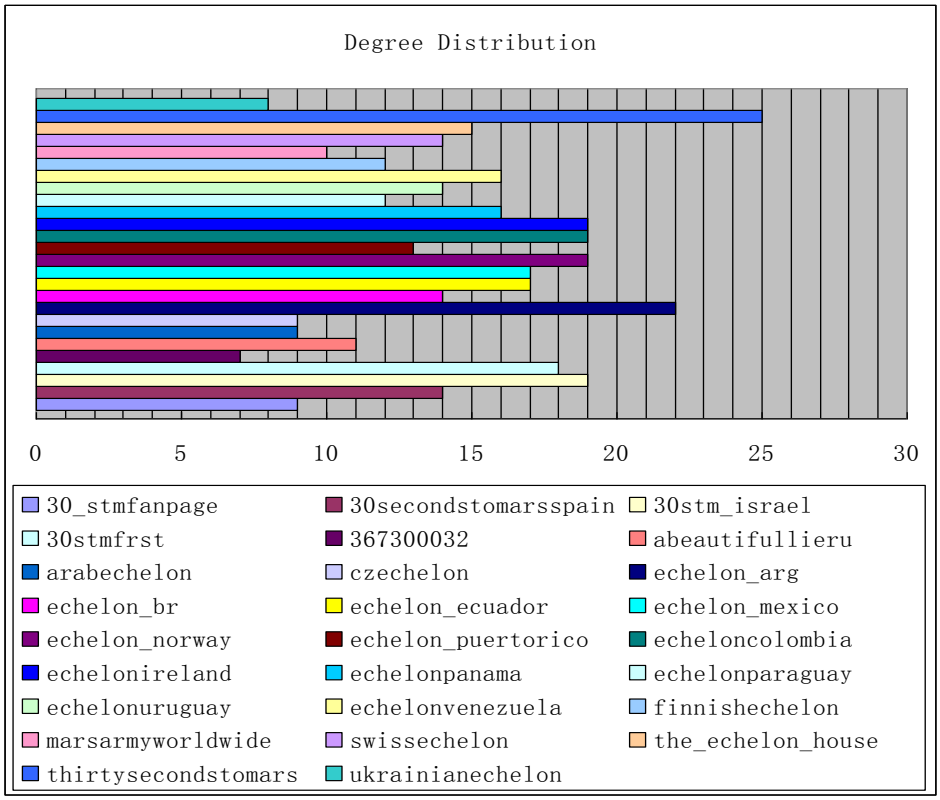


Figure 4.3 Degree Distribution

Based on this bar chart, the degree distribution of each vertex in the graph is similar with the power- low distribution. The degree of large majority of nodes falls in the range from about 10 to 19. Higher or lower values are rare.

Furthermore, Betweenness measures the degree an individual lies between other individuals in the network [21]. In the real case, it describes the number of people who a person is connecting indirectly through their direct links. Therefore, if the value of betweenness becomes small, it may point out that vertex as the bridge to connect different area due to the information flowing. In the current situation, vertex has a small value of betweenness may imply its boundary location. For proving this assumption, researcher applies the program to explore myspace network again, but stopping criteria has been set intentionally overestimation for output more than one community structure. The following picture presents the result.

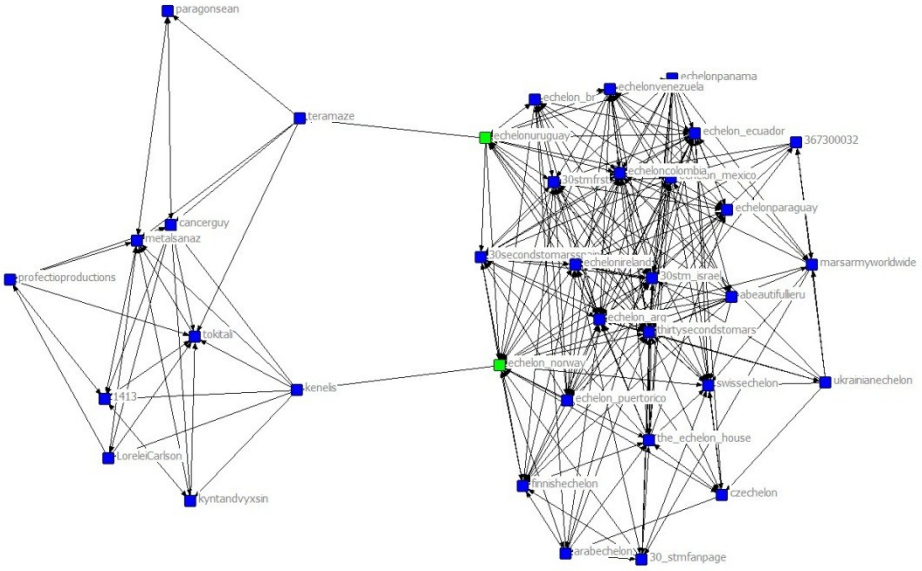


Figure 4.4 Boundary Vertices in the Community

It is clear to observe that the vertices {echelonuruguay, echelon_norway} located in the boundary and its betweenness for {1.248, 2.832}.

Moreover, the closeness measures how close the each vertex in the graph [21]. In this case, the small value denotes a very intimate relationship of people in myspace. Depending on information from appendix A, the start vertex has the lowest closeness that may imply it is the center leader of this group. In contrast, the boundary vertex, echelonuruguay, the closeness value is 62.

In addition, there is a phenomenon in the community that some vertices' name is similar. For example, echelon_mexico, echelonpanama, echelonuruguay, echelonireland, and all of them also have the similar value of closeness; it may reflect that they are in the same social group or joining the similar social actively. Considering the start vertex, thirty seconds to mars is the famous American rock band from Los Angeles formed in 1998 [30]. Thus those nodes could be deemed as the fans of this band, and the last alphabet may imply the area which fans' organization located, for example, Mexico, Panama, Uruguay, Ireland and so on.

Finally, the author tested each vertex's clustering coefficient and presented in appendix A. In generally, clustering coefficient is a measure of degree to which node in the graph tends to the same cluster [21]. It can be divided into global or local state. The global clustering coefficient gives overall indication of the network and local gives an indication of single node [21]. In this case, the overall graph clustering coefficient equals 2.704. The highest value of single node is echelonuruguay, 2.747. Furthermore, the high value of clustering coefficient may be induced by the disassortative mixing [3]. This phenomenon occurred in most complex network, the degree of adjacent vertices appears to be negatively correlated in network [31]. That is also to say, that node tends to connect other different characteristic node in the real network. Normally, in the complex network, there have two common types, assortative

mixing and disassortative mixing. Assortative is a bias in favor of connections between network nodes and similar characteristics [31]. That is the ordinary phenomenon. However, in the rare case, the disassortative mixing is a bias in favor of connections between network nodes and dissimilar characteristic [31]. In this case, the researcher built the similarity matrix and found the similarity value of each node often fall into the range between -0.434 to 0.637, which is typical.

Summary

Community structure in real network involves a lot of information. There are many ways to utilize the property of local community structure to understand the characteristic of real network. The author mainly analyzes some important parameters, degree, betweenness, closeness, and clustering coefficient. We need recognize those of parameters are also local; it means within global graph, the role of relevant node may has been changed.

Chapter V Algorithm Evaluation

Testing the algorithm means analyzing the full-know division of the network and recovering its community [24]. This chapter aims to describe how well the program performs by using standard test approach.

5.1 Synthetic network

The Benchmark generator has been used to create the synthetic network. It is free software written in C++. The traditional method of synthetic network generation named computer-generated networks (ad hoc network) given by Newman and Girvan [12]. They generate a large number of graphs with n=128 vertices, divided into four equal sized group of 32 vertices each. The vertex degree is divided into the P_{in} and P_{out}. P_{in} represents the edges have been randomly connected to the other nodes in the same community and P_{out} means to connect vertex in the different communities. Besides, the value of P_{in} and P_{out} were chosen to make the expected degree of each vertex constant equals 16 [12]. The following figure presents this type of the synthetic network.

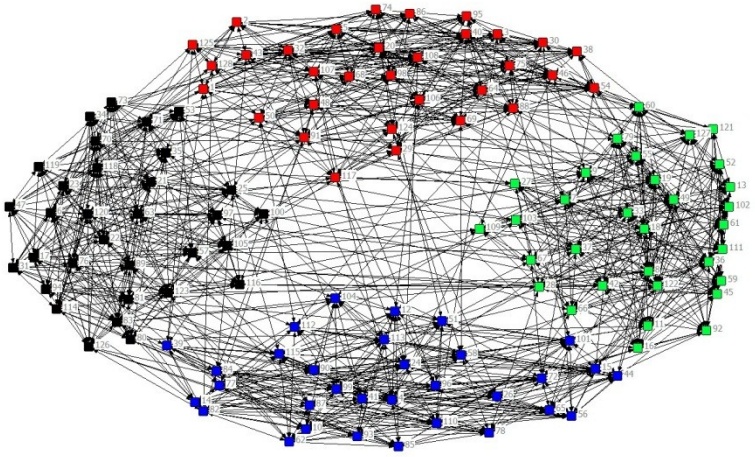


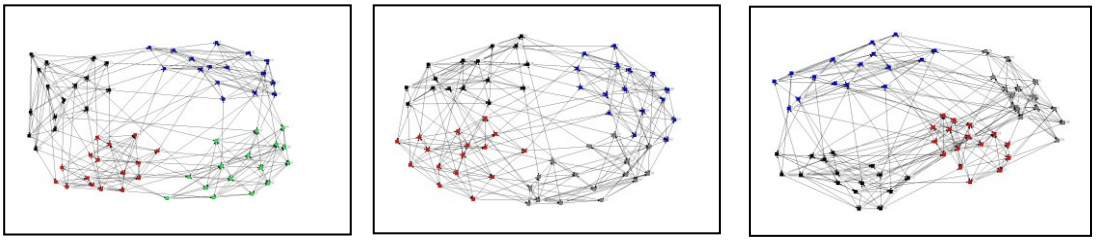
Figure 2.1 ad hoc Network

Based on the current situation, the author generates the synthetic network for total 64 vertices, and each vertex has the expected degree 8 which divided in the P_{in} and P_{out}. The network has consisted by the four equal sized communities, the relevant parameters presents in the following table.

Table 5.1 Parameters in Benchmark Generator

number of nodes	average degree	maximum degree	exponent for the degree distribution	exponent for the community size distribution	mixing parameter	minimum for the community sizes	maximum for the community sizes
64	8	8	1	1	0.2	16	16

In the real case, it is possible to vary the mixing parameter. This value means the average ratio of external degree/total degree. That is also to say, it can decide how many edges occurs inter-community. Thus, varying this value is similar with to vary Z_out. Finally, we create three synthetic networks with mixing parameter 0.2, 0.25 and 0.3 respectively.



mixing parameter 0.2 mixing parameter 0.25 mixing parameter 0.3

Figure 5.2 Three Synthetic Networks for Testing Algorithm

To evaluate the algorithm, the web-pages interconnections have been created as the Figure 5.2 dose, because the local algorithm has been designed to cooperate with crawler, thus cannot work solely. Then use the program to explore the network and evaluate its performance.

5.2 Evaluation Approach

It is clear that local algorithm often create a binary partition of the target graph into community itself, C, and the remaining non-community nodes, C = V – $\tilde{C}$ [24]. In the real network, the division is unknown but as for synthetic network, these partitions are fully known. Thus the equation $P_R = \{C_R, \tilde{C}_R\}$ is already known, but the found partition $P_F = \{C_F, \tilde{C}_F\}$ may differ [24]. That is the reason the author can use the program to explore these synthetic networks for testing how good the structure found is.

In contrast, the traditional approach is quantified by the number of correctly identified nodes but this method has some problems. For example, as the problem of stopping criteria described above, if the program fails to stop in time, some non-community nodes will still be identified as the one community. If the algorithm incorrectly finds one community as N nodes, but there exits K communities of N/K nodes each, one could set a+1/N for each correct node and -1/N for each incorrectly node, thus can give a composite score of 2/K-1 [24]. That implies the different synthetic networks which have different scales cannot be directly compared [24]. Thus, we can use normalized mutual information to measure how well P_R and P_F correspond to each other [24].

$$I(P_R, P_F) = \frac{-2\sum_i \sum_j X_{ij} \log(X_{ij} N / (X_{i.} X_{j.}))}{\sum_i X_{i.} \log(X_{i.} / N) + \sum_j X_{.j} \log(X_{.j} / N)} \quad (5.1)$$

X is the 2×2 confusion matrix, where the rows represent the real community and the columns denote the found communities. Thus X_{ij} means the number of nodes from real group i that have been found in group j [24]. Besides, the sum of over row i of matrix X_{ij} is denoted $X_{i,}$ $X_{i.}=X_{i1}+X_{i2}$. The sum of over columns j of matrix X_{ij} is denoted $X_{.j}$, $X_{.j}=X_{1j}+ X_{2j}$ [24]. Generally speaking, I ($P_{R,}$ P_F) means how much is known about partition P_R by knowing partition P_F [24]. Thus if the found communities have been identified as the real communities, the value I ($P_{R,}$ P_F) =1.Otherwise, if the found communities are totally independent with the real communities, for instance, the whole network has been identified as the one community, I ($P_{R,}$ P_F) = 0 [24].

To calculate the mutual information, the author chose Andrea Lancichinetti's program to process data. The program requires two files which divided as inputting and outputting for mutual value. That is also to say, those files can be deemed as the real cluster and found cluster. The following figure illustrates the result below.

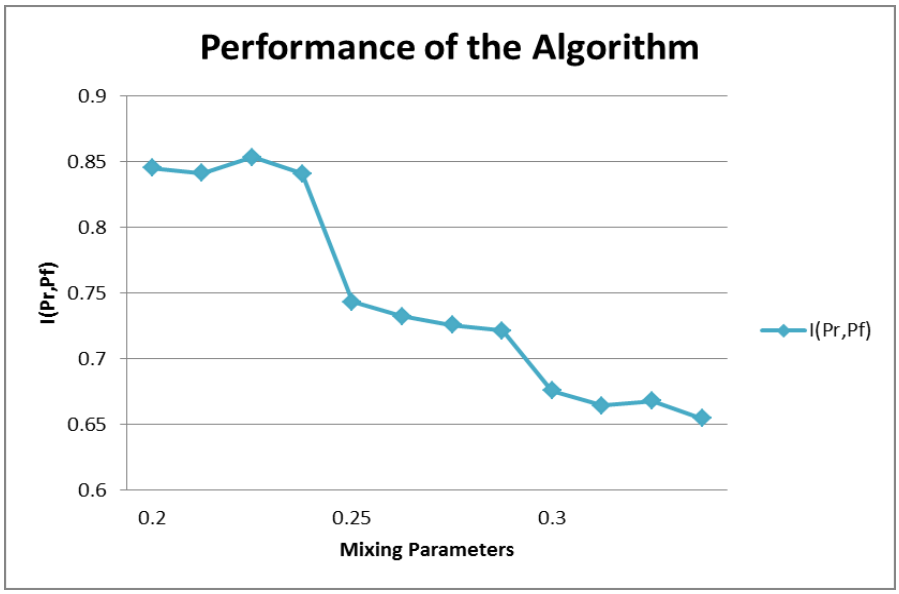


Figure 5.3 the Result of Testing the Algorithm by Exploring three Synthetic Networks

In this approach, the communities within three different synthetic networks with mixing parameter 0.2, 0.25 and 0.3 have been used to support the real partition respectively, then using the program to explore them and output relevant found partitions. In addition, the information has been normalized mutual for getting I ($P_{R,}$ P_F). It is clear that the small number of inter-community edges can lead the partition easily and accurately. That is the reason as the amount of mixing parameter increasing; the accurate of the algorithm becomes decreasing. Furthermore, it also should notice that the mixing parameter cannot constant the outward links in the network. It means within each synthetic network, the vertex's Z_out is not constantly same. That may affect the result due to randomly choose the starting vertex. In other words, the different P_out of starting vertex may get the different accuracy of the algorithm. The reason is we assume the starting vertex belongs with the strong community where its

boundary equals zero and local modularity equals one. Therefore, the author considered this assumption is too idealization since it may directly affect the subsequent node agglomeration, especially within real network, the state of starting vertex is unknown until the significant large of graph has been explored. Accordingly, this part could remain for further research.

Finally, the average local modularity R during each agglomeration step has been presented. Based on this information, the local peak value of the local modularity occurs when the enclosed community has been detected. Then the value will be decreased until some new vertices have been merged. The next local peak will occur in the same condition, but the fluctuation tends to be smooth when the adequate numbers of vertices have been agglomerated. Figure 5.4 presents this phenomenon.

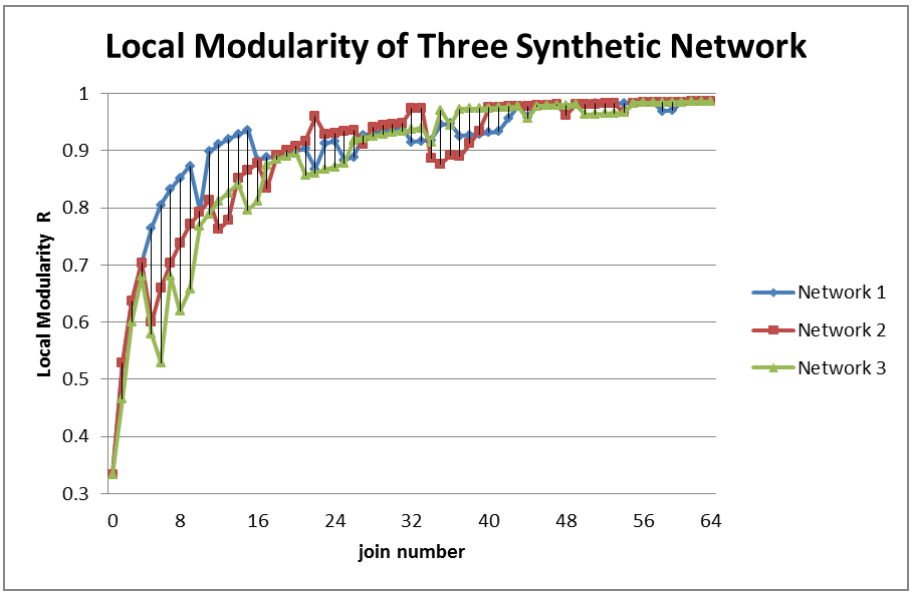


Figure 5.4 the Varying Value of Local Modularity within Synthetic Network

Summary

Evaluation the performance of community detection algorithm is a research topic in recent year. The author uses the normalized mutual information to test the accuracy of the algorithm. Finally, this part also presents the local modularity under the different type of synthetic networks for understanding the properties of the agglomeration node. Next chapter will describe the breadth first searching and compare it with the local algorithm.

Chapter VI Comparing BFS searching

In general, the property of breadth first searching will be described in this chapter. Then, using this method to detect the community structure and compare with local algorithm.

6.1 Breadth First Searching

Breadth first searching is an algorithm that can find the target node in the graph. From the root nodes, programs explores all fixed distance neighbors at a time, then iterate those nodes and go into the next layer of neighbors. The relevant pseudocode presents below [32].

1. From the start node and push it in the list
2. Detect the all neighbors that the node touches
3. Push those nodes in the list.
4. Iterate all nodes in the list
   If the node has been explored
   Skip that node and continues next
   Else add that node in the list
5. Check the terminal condition.
6. Repeat step 2

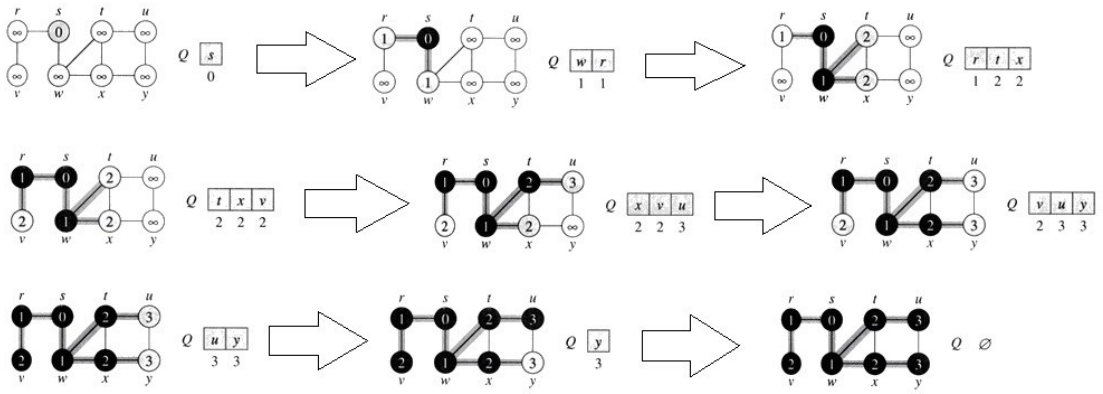


Figure 3.1 each Process of Breadth First Searching

Taken from Ref [21]

6.1.1 Space Complexity

Considering all of the nodes in the same level must be explored until its child nodes has been discovered, the space complexity can be deem as the number of nodes at the deepest level [32]. Thus, we can set the branching factor as b and graph depth as d, the complexity equals $b^1 + b^2 + b^3 + \ldots + b^d = O(n^d)$ [32]. Moreover, if we have the global knowledge of this graph, the space complexity can also be expressed as O(|E|+|V|), where E denotes the number of edges and V denotes the number of vertices [32]. Furthermore, in the worst-case, the graph

has the depth one and all nodes must be explored and push them in the list. In the real application, breadth-first searching often apply in bounded space [32].

6.1.2 Problem solving

Normally, Breadth-first searching can be used to solve these problems below [32].

- Testing whether graph is connected
- Computing spanning forest of graph
- Detection the minimum number of edges in the graph that start from starting vertex to the current vertex or reporting there is no directly links between those two vertices.
- Detection the cycle in the graph or reporting there is no cycle existing.

6.2 Weak Community Definition

As the equation 3.2, the strong community can be defined as $K_i^{in}(C) > K_i^{out}(C), \forall i \in C$ [24]. It means each node in the community must have more neighbors inside than outside. Based on the Luo, Wang and Promislow (LWP) algorithm, the new local modularity has been defined as following.

A is the sub-graph of the graph G, A∈G, B is the boundary set, N is the neighbor set, thus $A_{i,j}$ is the adjacent matrix that means the vertex in the community region have relation to the neighbor region. Community region involves central community C and boundary set B, thus, community region = C ∪ B.

$$A_{i,j} = \begin{cases} 1 & \text{If vertex i and j are connected, and either i or j belong with community region.} \\ 0 & \text{Otherwise} \end{cases}$$

$$M_{in} = \sum_{i,j} A_{i,j}[i \in C,B][j \in C,B] \qquad \delta(i,j) = [i \in C,B][j \in C,B] \qquad (6.1)$$

Where $\delta(i,j)$ is 1 if vertex i and j belong with community region and 0 otherwise

$$M_{out} = \sum_{i,j} A_{i,j}[i \in B][j \in N] \qquad \lambda(i,j) = [i \in B][j \in N] \qquad (6.2)$$

Where $\lambda(i,j)$ is 1 if only one vertex i and j belong with community region and 0 otherwise [27].

In specific, M_{in} means how many endpoints within community and M_{out} means how many endpoints within boundary region. Comparing with Clauset's method, this approach directly computes the ratio of internal and external edges. Thus, local modularity M as below [27],

M = (amount of internal edges) / (amount of external edges) (6.3)

Depends on the LWP algorithm, if M≥1, a sub-graph is a community. This condition is similar with Clauset's method, with R≥0.5 [24].

In addition, this local modularity is closely related to the concept of weak community. It means the community is weak if $M_{in} > (1/2)\ M_{out}$ [24]. Thus, the weak community is defined if this equation has been satisfied.

Weak Community = $1/2 < M_{in} / M_{out} < 1$ (6.4)

6.3 Comparing Local algorithm and BFS Searching algorithm

6.3.1 Detection Local Community Structure using BFS Searching

Firstly, the author chose starting vertex and explores its adjacent matrix, then use BFS algorithm to iterate all vertices and merge those nodes into community. Thus this is the extremely greed method to find the community structure because it not consider each node's property. Furthermore, the stopping criteria of P-strong community is not appropriate method in this case due to a significant number of nodes may fails this condition. Therefore, the terminal condition has been replaced by the scale of community. This picture presents the network detected by myspace using BFS searching. The number of vertex in community is 60.

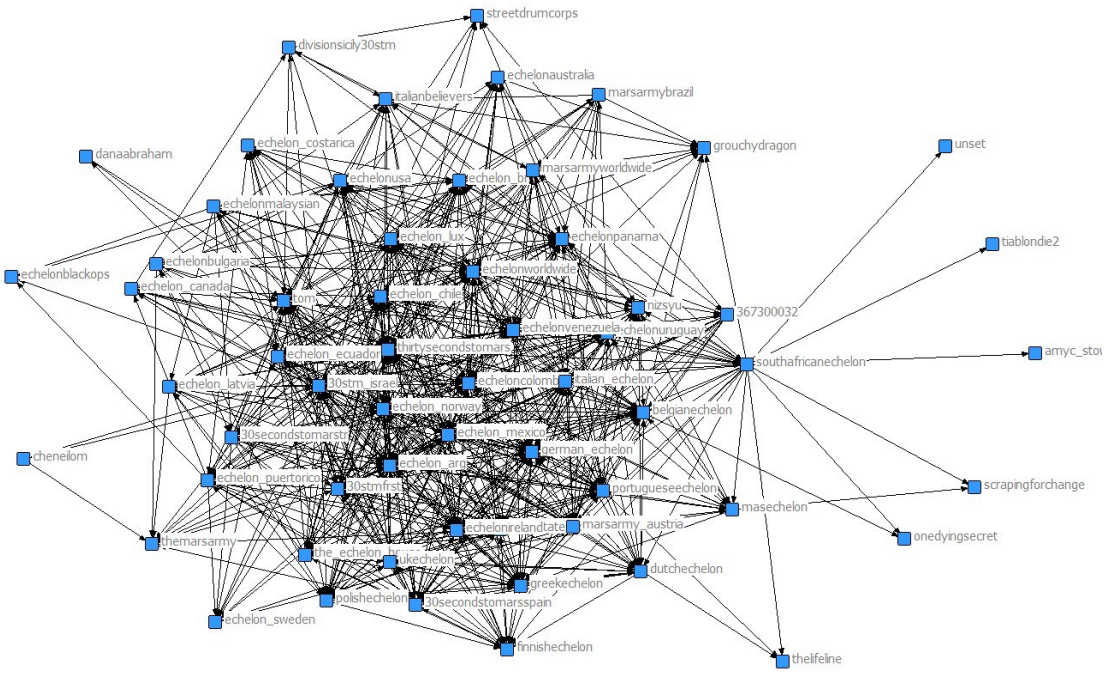


Figure 6.2 using BFS Searching to Find Community in Myspace

6.3.2 Comparing algorithm using Strong and Weak community

In the chapter 4, the definition of strong and P-strong community have been described. In this chapter, the definition of weak community will also be introduced. Therefore, if it is possible to identify outputting community belong with either one (weak or strong), the performance of algorithm is simply to exam. In other words, if the local algorithm outputs the P-strong community and the BFS searching outputs weak community, thus performance of local algorithm is better than BFS and vice versa. In another case, if both of them output the community belong with P-strong, it is able to vary P to decide how many nodes achieve the condition since the community is P_1 strong is also P_2 strong ($P_1 > P_2$).

Besides, due to fairly compare the result community, the researcher set the same stopping criteria and the same starting vertex. Therefore, amount of vertex in the community has been used to stop the program. The following figure illustrates the result of the two algorithms output.

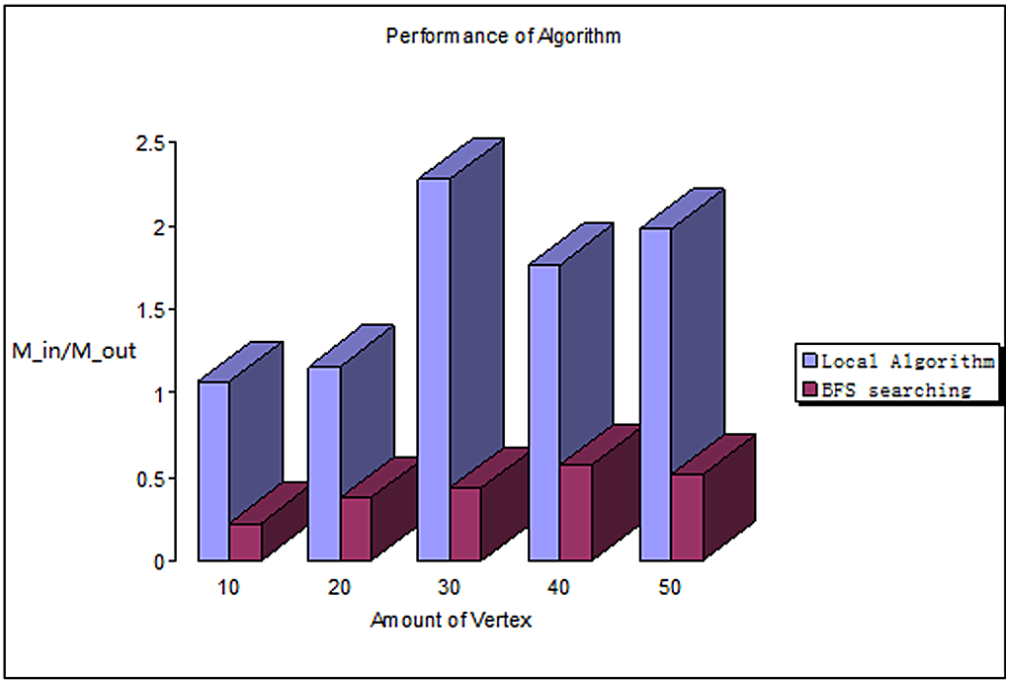


Figure 6.3 Performance of two Algorithms, Data Collected by Myspace Network

Depends on the definition of strong and weak community, the community given by the local algorithm at least achieves P-strong due to the ratio of M_in and M_out. On the other hand, comparing BFS searching, it belongs with the weak community because the ratio value only around in 1/2. Therefore, the performance of local algorithm is better than BFS searching.

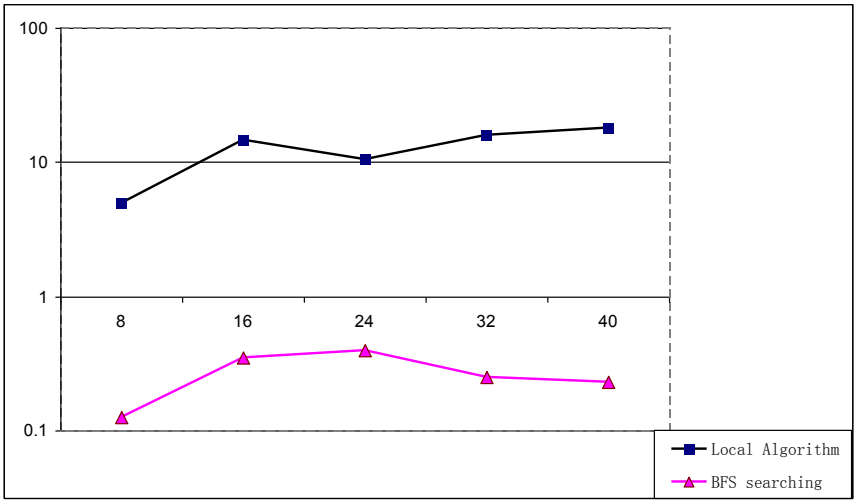


Figure 6.4 Performance of two algorithms, data collected by synthetic network with mixing Parameter 0.2

The result given by the synthetic network is also similar. Local algorithm performs very well due to the significant high ratio with M_in and M_out. Comparing BFS searching, the community structure is too weak due to the equation 6.4.

Furthermore, the agglomeration step of two methods is also quite different. It is known that the local algorithm merges node that depends on its current R, and the current boundary status. Therefore, during each step, only small degree of neighbor node can be agglomerated. The clustering coefficient can be used to express this condition. It defines how close the vertex and its neighbors are to being a complete graph [21]. Thus, the node only small outward links and large inward links will be merged. Comparing BFS searching, this is the extremely greed approach due to the node's property has been ignored. It means all candidate nodes which touched the start node will totally be added in the community. Therefore, the quality of community is worse than the local method.

6.3.3 Modularity Analysis

As become the standard quantifying the strength of the community structure, modularity can be used to exam how good the structure found is [12]. Thus author used this value to compare the performance of two algorithms. In generally, the high value of Q will reflect the high strength of community structure.

Moreover, the synthetic networks with mixing parameter 0.2 are selected to evaluate the performance of the two algorithms. These figures show the result network, total for 33 nodes.

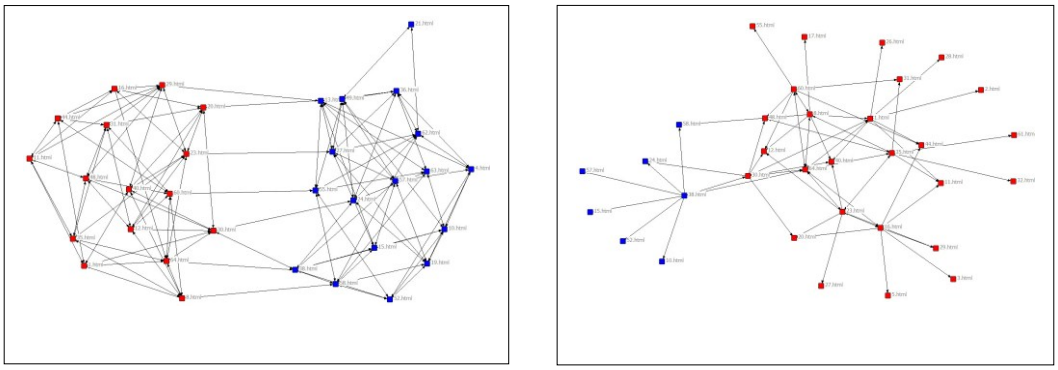


Figure 6.5 Result Taken From Synthetic Network with Mixing Parameters 0.2

To simply evaluate the modularity, the author only chose amount of the two community's nodes, (approximately 32 nodes to terminal condition), the left side network produced by local algorithm and right side given by the BFS searching.

Based on these result, hierarchical analysis has been used to find the best partition of each network. Furthermore, dendrogram is utilized to analyze the data and present relevant modularity respectively, which is shown by chars below:

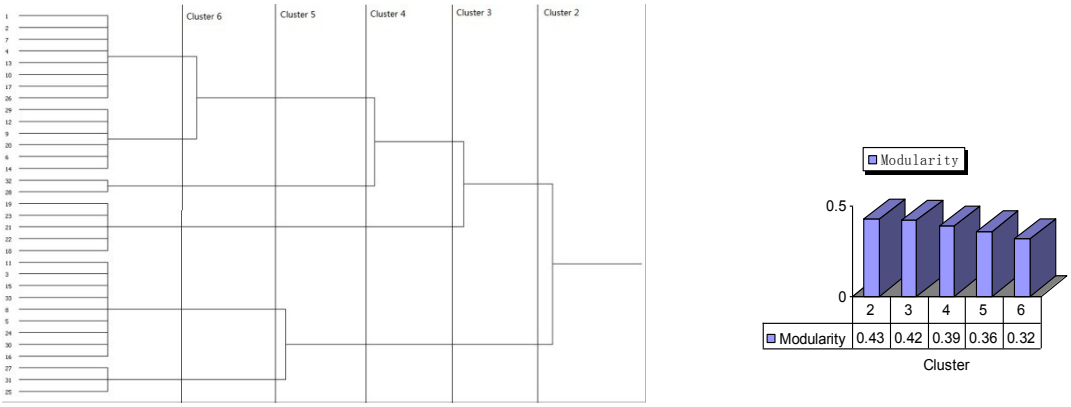


Figure 6.6 plot of the modularity and dendrogram for synthetic network, data collected by Local Algorithm

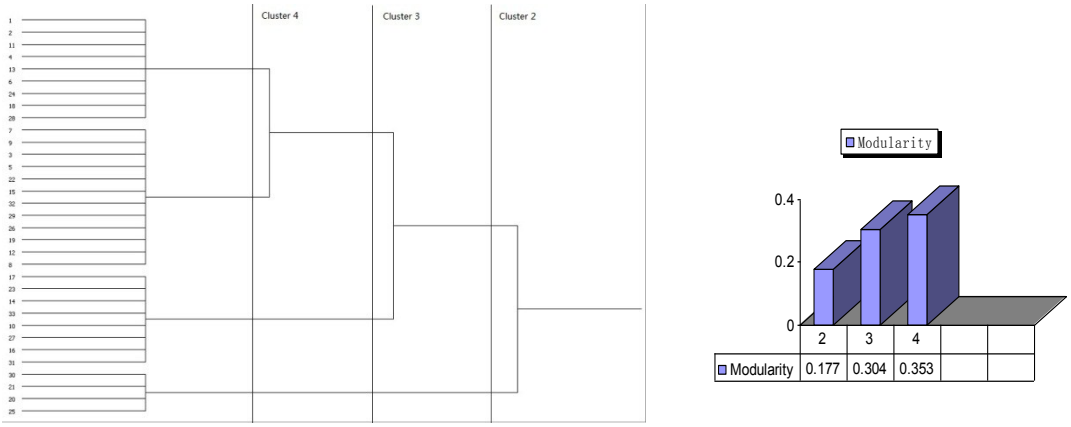


Figure 6.7 plot of the modularity and dendrogram for synthetic network, data collected by BFS searching

Within dendrogram, the number in the left side denotes the vertex in the synthetic network, the vertical line in the right side present the partition of the graph. The modularity of each partition in the histogram is also displayed in this figure.

From the given result, local algorithm performs very well within two clusters. The author set the stopping criteria with 32 vertices and two communities have been correctly classified, thus the peak modularity value occurred in partition two, 0.43, which is typical. Comparing with BFS searching, the situation is opposite. The peak modularity value occurred in partition four, with 0.353. The reason is BFS searching will try to find the whole graph, not focus on local community structure due to each node has been unconditionally added in the community. Therefore, the vertex which placed in the different four communities has been found. That is the reason when cluster equals four, the high value of modularity occurs. Moreover, based on the tendency of varying modularity, within local algorithm, the peak modularity value implies the best separation of current graph. Then the value decreases because the number of communities has been overestimated. About the BFS searching, due to nodes have been unconditionally merged, it is hardly to find the community structure with two clusters. However, as number of partition increasing, the value of modularity also increase because the node detected by BFS is randomly distributed. That implies the vertices may locate in the four different communities. Therefore, we can recognize BFS searching find the whole graph, not the local community structure.

6.3.4 Comparing Time Complexity

In general, the running time of local modularity algorithm is O (k^2 d), where K is the number of vertices in the community and d is the mean degree of graph, or just O(k^2) for a sparse graph since the vertex's degree is significant low [3]. Furthermore, the running time of BFS searching is O(n^d) [32], or expressed as O(|E|+|V|) [32]. Where e is the number of edge and v is the number of vertex. Therefore, even though the quality of output communities has not enormous difference, the local algorithm is more efficient than BFS searching.

Summary

Normally, crawler is used to fetch information from web. In this program, breadth first searching has been used to embed in the crawler for finding adjacent matrix. Based on the result of comparing with two algorithms, it could be recognized that the breadth-first searching may be not appropriate approach of community finding due to it try to find the whole network. However, it is also a useful method to cooperate with other algorithm in community finding task.

Chapter VII Conclusion

This chapter states the summary information with some key points, for instance, implementation local community detection algorithm, evaluation and comparison. Finally, some potential research direction with local community detection will be recommended.

7.1 Summary

- **Summary 1:**

Implementation local community detection algorithm to explore World Wide Web should notice that the edge within network is directed but the algorithm generally assumes that edges are undirected. This becomes especially important when the program decides which adjacent vertex to agglomerate, since it might have large in-degree and small out-degree, but the form of the network is unknown until the adequate subgraph has been explored. That is also to say, choosing an appropriate starting vertex is also vital due to the hob node or leaf node will directly affect the performance of community detection. Furthermore, an appropriate stopping criterion is also important for the accurate result. Clauset proposes stopping agglomeration until an arbitrary number of vertices is reached within the community. However, this approach is not suitable within some real network, especially WWW, because the status of target region is unknown before explore it. That implies if this region lack of some community structure, the program also be terminated due to arbitrary number achievement. To solve this problem, the other researcher's method has been implemented, that is p-strong community. In generally, this approach will exam the property of each candidate vertex until the terminal condition has been satisfied. Comparing the previous approach, this method is more strict due to the program will constantly explore network unless the community structure has been found.

- **Summary 2:**

To establish synthetic network is the standard method to evaluate the algorithm of community detection. This type of network has some properties that cannot be ignored. For example, the equal sized community, which can be used to normalized mutual information. In addition, based on the experimental data, increasing number of edges inter-community will lead the accuracy of community detection decreasing. That implies the smallest outward links will lead to communities that are easier to find.

- **Summary 3:**

Breadth first searching has widely applied to exam the graph. In the program, it also has been used to cooperate with local algorithm. Thus, if BFS can directly detect the community structure, it is very efficient than any existing approach. Unfortunately, the result network given by BFS is still only identified as weak community. In addition, based on modularity analysis, BFS method is identified to find the whole network, not focus on the local community structure.

7.2 Further Research Recommendation

First of all, Clauset developed the local modularity algorithm to find the community structure. The running time is O (k^2d). However, this is the one vertex at a time method. Which means only one candidate vertex can be merged in the community during each process. Although this is a good approach to find one community in network, but within large network, especially with WWW, it may be inefficient due to the possible solution of iterative local expansion to process nodes [26]. That is also to say, when the local community C_0 starting has been found, then using crawler to explore C_0's adjacent matrix and insert them into neighbor set. In addition, randomly choose one node in that matrix and using local algorithm to find the community $C_1, C_2,$ until the terminal condition (C_n) has been satisfied. This iterative approach given by Cheng, Zaiane [26], but its accuracy should further discussion. Besides, an appropriate stopping criteria within this method is also worth study. One possible solution is combining Bagrow's method [24], this part could remain in the future work.

Furthermore, as we described above, stopping criteria is a vital parameter within local community detection algorithm. In this paper, the two possible methods have been implemented in the program: arbitrary number within community and p-strong community. However, there may a room for improvement. Bagrow propose a new method named trailing least-squares and identify its performance within Amazon co-purchasing network [24]. Therefore, the further work could be considered as identify its accuracy within different types of network.

Finally, the properties of local community structure are worth studying to understand the real world network. In this paper, some characterizations with myspace network have been discussed, further information are beyond the scope of this dissertation. Nonetheless, the property of surrounding area is also worth studying.

Reference

[1] Bagrow, P. J and Bolt, M. Eric. (2005) *Local Method for Detecting Community* [J]. Phys. Rev. E, 72(4):046108

[2] Clauster, A and Mark, E. J. Newman. (2004) *Finding Community Structure in very Large Networks* [J]. physical Review E, 72:02613

[3] Clauset, A. (2005) *Finding Local Community Structure in Network* [J]. Physical Review E, 72(2):026132

[4] Ester, M, Kriegel, H. P and Xu, X.W. (1996) *A Density Based Algorithm for Discovering Clustering in Large Spatial Databases with Noise* [C]. In proceeding of Second International Conference in Knowledge Discovery and Data Mining pages 226-231.

[5] Girvan, M and Mark, E. J. Newman. (2002) *Community Structure in Social and Bio-Logical Networks* [J].PNAS, 99(12):7821-7826, 2002.

[6] Karypis, G, and Kumar, V. (1999). *A Hierarchical Clustering Algorithm, Using Dynamic Modeling* [J]. Computer, 32(8):68-75

[7] Johnson, S.C. (1999) *Hierarchical Clustering Schemes* [J]. Psychometric, 32(3):241-254

[8] Kernighan, B. D and Lin, S. (1970) *An Efficient Heuristic Procedure for Partition Graphs* [J]. The bell system Technical Journal, 49(2):291-307,

[9] Lintion C.Freeman. (1997) *A Set of Measure of Centrality Based Upon Betweenness* [J]. Stoichiometry, 40(1):35-41

[10] Lintion, C. Freeman. (2004) *The Development of Social Network Analysis*, Vancouver, Empirical press

[11] Mark, E. J. Newman. (2003) *The Structure and Function of Complex Network* [J]. SIAM Review, 45:167-256

[12] Mark, E. J. Newman. (2004) *Finding and Evaluating Community Structure in Networks* [J]. physical Review E,60.026113,2004

[13] Mark, E. J. Newman. (2004) *A Fast Algorithm for Detecting Community Structure in Networks* [J]. Physical Review E, 69:066133

[14] Mark, E. J. Newman. (2004) *Detection Community Structure in Networks* [J] Eur.phys.J.B 38, 321-330

[15] Mark, E. J. Newman, [online] *The mathematic networks* Avaliable at: http://www.commetrix.de/IRIS/IRIS_2007-version-resubmitted-08jun07.pdf [Accessed 5/5/2010]

[16] Mehmed, K. (2002) Data Mining: Concepts, Models, Methods, and Algorithms, West Sussex, Wiley-IEEE Press

[17] Philip,V. Fellman [online] *Modeling terrorist Networks-Complex System at the Mid-Range* Available at: http://www.psych.lse.ac.uk/complexity/Conference/FellmanWright.pdf [Accessed 3/5/2010]

[18] Radicchi, F, Castellano, C, Cecconi, C, and Loreto, V. (2004) *Defining and Identify Communities in Networks* [C]. In proceeding of Natl Acad. Sci., volume 101, page 2658-2663

[19] Romm, C, Pliskin N and Clarke, R. (1997) *Virtual communities and society：toward an integrative three phase model* International journal of information mandgement,17(4), 261-270.

[20] Stogaz, H, S. (2001) *Exploring Complex Networks* [J]. Nature, 410:268-276.

[21] Scott, j. (2000) *Social Network Analysis: A Handbook, 2nd ed*, London, Sage press

[22] Wan, Y, Chen, D. B. (2009) *P2P Botnet Control Strategy Based on Social Network Analysis* [J] Computer Science, volume 36, No.6

[23] Dunbar, R (1998) *Grooming, Gossip, and the Evolution of Language* [M].Harvard University.

[24] Bagrow, P.J. (2008) *Evaluation Local Community Methods in Network*. [J]. J.Stat.Mech, P05001

[25] Danon, L, and Díaz-Guilera, A. (2005) *Comparison Community structure identification.* [J]. J.Stat.Mech, P09008

[26] Cheng, J, Zaiane, O, R and Goebel, R. (2009) [online] *Detecting Community in Large Networks by Iterative Local Expansion* Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.6784&rep=rep1&type=pdf [Accessed 10/7/2010]

[27] Luo, F, Wang, J.Z and Promislow, E, (2006) *Exploring Local Community Structure in Large Networks* [J]. Web Intelligence (Piscataway, NJ: IEEE Computer Society) pp 233–9.

[28] Marcus, K. (2008). *Mean clustering coefficients: the role of isolated nodes and leafs on clustering measures for small-world networks* New Journal of Physics **10** (8): 083042.

[29] Jung. (2010) *Java Universal network/graph Framework* Available at: http://jung.sourceforge.net/ [Accessed 15/7/2010]

[30] 30 seconds to mars (2010) [online] *30.seconds to mars by rich and shelley* Available at: http://www.30secondstomars.ca/30stm/index.htm [Accessed 20/8/2010]

[31] Mark, E. J. Newman. (2003) *Mixing Patterns in Networks* Physical Review E **67**: 026126

[32] Knuth, Donald, E. (1997) *the Art of Computer Programming Vol 1. 3rd ed.*, Boston: Addison-Wesley.

[33] Gregory, S (2007) an Algorithm to Find Overlapping Community Structure in Networks [J], PKDD, pp.91-102

Appendix A. nodes property in myspace network

Vertex	Degree	Betweeness	Closeneess	Clustering Coefficient
30_stmfanpage	9	1.904	67	2.056
30secondstomarsspain	14	1.722	62	2.407
30stm_israel	19	10.134	57	1.842
30stmfrst	18	6.539	58	2.065
367300032	7	1.004	69	2.095
abeautifullieru	11	1.207	65	2.364
arabechelon	9	1.688	67	2.056
czechelon	9	1.959	67	1.778
echelon_arg	22	14.544	54	1.745
echelon_br	14	2.496	62	2.407
echelon_ecuador	17	5.413	59	2.169
echelon_mexico	17	6.74	59	1.926
echelon_norway	19	2.832	57	1.784
echelon_puertorico	13	1.248	63	2.821
echeloncolombia	19	7.452	57	2.012
echelonireland	19	8.957	57	1.801
echelonpanama	16	5.031	60	2.083
echelonparaguay	12	0.975	64	2.636
echelonuruguay	14	1.248	62	2.747
echelonvenezuela	16	3.818	60	2.167
finnishechelon	12	2.509	64	2.182
marsarmyworldwide	10	3.197	66	1.778
swissechelon	14	6.045	62	1.473
the_echelon_house	15	5.444	61	1.924
thirtysecondstomars	25	23.229	51	1.613
ukrainianechelon	8	1.684	68	2

Appendix B Essential Codes

The following page presents some vital methods within two main classes, one for community detection, the other for dynamically visual data in user graph interface. The entire program has been uploaded within relevant web page.

```
/**
 * This class has been applied to detect community structure in myspace.
 *@author miyun fan
 * @version 1.0
*/
 // import file omitted
 // method omitted
/**
        * This method is used to connect to the internet and explore the particular
        * web page depends on the user input, then read a line with its hyperText.
        * If the corresponding page cannot find, throw the IO exception.
        * @param strUrl
        */
       private void ReadHtml(String strUrl) {
              try {

                     URL url = new URL(strUrl+"/friends");
                     URLConnection conn = url.openConnection();
                     conn.setDoOutput(true);
                     InputStream is = null;
                     is = url.openStream();
                     BufferedReader bReader = new BufferedReader(new
                     InputStreamReader(is));
                     StringBuffer sb = new StringBuffer();
                     String rLine = null;
                     String tmp_rLine = null;

while ((rLine = bReader.readLine()) != null) {
                            tmp_rLine = rLine;
                            int str_len = tmp_rLine.length();
                            if (str_len > 0) {
                                   sb.append("\n" + tmp_rLine);
                                   processHtml(tmp_rLine, strUrl);
                            }
                            tmp_rLine = null;
                     }
                     is.close();
              } catch (IOException e) {
                     System.out.println("cannot read the HTML");
              }
       }
/**
 * This method is a filter that only remains the valid information defined
 * by the regular expression.
 *
 * @param inputUrl
 * @param strUrl
 */
public void processHtml(String inputUrl, String strUrl) {
       String info = "";
       String regUrl = "http://www\\.myspace\\.com/?[^\\s/]{2,20}(?=\")";
       Pattern p = Pattern.compile(regUrl, Pattern.CASE_INSENSITIVE);
       Matcher m = p.matcher(inputUrl);
       boolean blnp = m.find();
while (blnp == true) {
       // ignore the no meaning vertex
       if (robot.checkRobotInfo(m.group())) {
              break;
       }
       // using globa map to store all of the information
       putMapList(strUrl, m.group());
       if (!deepUrls.containsKey(m.group())) {
              deepUrls.put(m.group(), (deepUrls.get(strUrl) + 1));
              if (deepUrls.get(strUrl) == 1) {
                     neighbors.add(m.group());// add the element within
neighbors
              } else {
                     unKnowArea.add(m.group());
              }
              info = "Webdepth:" + (deepUrls.get(strUrl)) + " " + "Vertex:"
                            + m.group(0) + "\n";
              System.out.println(info);
       }
       inputUrl = inputUrl.substring(m.end(), inputUrl.length());
       m = p.matcher(inputUrl);
       blnp = m.find();
}
}
```

```
/**
         * This method is calculate the dR of each vertex within neighbor.
         */
        public double localModularuty(int temB_in,int B_in,int B_out,int
increasingEdges,int decreasingEdges,double R){
                int den = 0;
                double mol = 0;
                double dR = 0;

                den = B_in + B_out + increasingEdges - decreasingEdges;
                mol = ((temB_in) - (R*increasingEdges)) - (decreasingEdges*(1-R));
                if(den == 0)
                {
                        dR = 0;
                        return dR;
                }else{
                        dR = mol/den;
                        return dR;
                }
        }
```

```
/**
         * This method is try to find the maximum value of dR, then get the relevant
         * vertex. Besides, add it into community and delete it in neighbors.
         * Finally, the while loop within webcrawler method has been used to explore
         * the new vertex until the terminal condition has been activated.
         */
        public void findVertexintoCommunity() {
                double maximum = -100;
                String vertex = "";
                String info = "";
                int tem = 0;
                int temB_in = 1;
                // try to find the maximum value of dR and get the relevant vertex.
                for (Map.Entry<String, Double> state : nodeDR.entrySet()) {
                        if (state.getValue() > maximum)
                                maximum = state.getValue();
                }
                for (Map.Entry<String, Double> it : nodeDR.entrySet()) {
                        if ((it.getValue() == maximum) && (!community.contains(vertex)))
                                vertex = it.getKey();
                }
                // add it in the community, delete it in neighbors.
                community.add(vertex);
                visual.ViewProgress();
                boundary.add(vertex);
                neighbors.remove(vertex);
                showNetwork(vertex);
                // delete it in hash-map which has been stored the dR and its vertex
                if (nodeDR.containsKey(vertex))
                        nodeDR.remove(vertex);
                tem = neighbors.size();
                // update the neighbors
                Set<Entry<String, Set<String>>> set = nodeRelation.entrySet();
                for (Entry<String, Set<String>> entry : set) {
                        if (entry.getKey().equals(vertex)) { // compare key
                                Set<String> lst = entry.getValue();
                                for (String value : lst) {
                                        if(!community.contains(value))
                                        neighbors.add(value);
                                }
                        }
                }
                int preB_out = 0;
                preB_out = updateBoundary(vertex);
                // Update local modularity (R)
                this.B_out = neighbors.size() - tem + B_out - preB_out;
                temB_in = findTemB_in(vertex);
                this.B_in = (B_in + temB_in);
                R = B_out + B_in;
                this.R = B_in / R;
                write.Write(info);
                TeststoppingCriteria(vertex);
        }
```

```
/**
        This method is used to calculate dR value.
*/
public void dRcalculate() {
                int B_out_edges = 0;
                int increasingEdges = 0;
                int decreasingEdges = 0;
                double dR = 0;
                int index = 0;
                for (String it : neighbors) {
                        temList.add(it);
                }
                // this code is check the particular page whether has been explored,
                // if the vertex has not been explored, call the method of ReadHtml to
                // explored relevant vertex. Otherwise,
                // using the old B_out value.
                while (index < temList.size()) {
                        String nodeInfo = temList.get(index);
                        if(!exploredList.contains(nodeInfo)){
                        exploredList.add(nodeInfo);
                        ReadHtml(nodeInfo);
                        // this code is try to find each neighbor's size (B_out_edges).
                        B_out_edges = findTemB_out();
                        vertex_temB_out.put(nodeInfo,B_out_edges);
                        }
                        B_out_edges = vertex_temB_out.get(nodeInfo);
                        int temB_in = 0;
                        if (B_out_edges == 0) {
                                temList.remove(nodeInfo);
                                neighbors.remove(nodeInfo);
                                continue;
                        }
                        if (this.times == 1) {
                                temB_in = 1;
                                increasingEdges = B_out_edges;
                                System.out.println("increasingEdges:" + increasingEdges);
                        }
                        if (this.times != 1) {
                                temB_in = findTemB_in(nodeInfo);
                                // find the increasing or decreasing edges
                                if ((B_out_edges > this.B_out) || (B_out_edges == this.B_out)) {
                                        if (B_out_edges == B_out) {
                                                increasingEdges = 0;
                                                decreasingEdges = 0;
                                        }
                                } else {
                                        decreasingEdges = B_out - B_out_edges;
                                }
                        }
                        dR = localModularuty(temB_in, B_in, B_out, increasingEdges,
                                        decreasingEdges, R);
                        nodeDR.put(nodeInfo, dR);
                        index++;
                }
/** This method is used to dynamically visual the network in user graph interface
public void showNetwork(String vertex) {
        for (String in : community) {
        Set<Entry<String, Set<String>>> set = nodeRelation.entrySet();
                for (Entry<String, Set<String>> entry : set) {
                        if (entry.getKey().equals(vertex)) { // compare key
                                Set<String> lst = entry.getValue();
                                for (String value : lst) {
                                if (value.equals(in)) { // compare value
                                visual.addVertex(vertex.substring(vertex.lastIndexOf('/') + 1));
                                visual.addEdge(new Edge("Edge"),vertex.substring(vertex
                                                .lastIndexOf('/') + 1),
                                        value.substring(value.lastIndexOf('/') + 1));
                                                visual.View();
                                        }
                                }
                        }
                        if (community.contains(entry.getKey())) { // compare key
                                Set<String> lst = entry.getValue();
                                for (String value : lst) {
                                if ((value.equals(in)) && (value.equals(vertex))) {
                                visual.addVertex(vertex.substring(vertex.lastIndexOf('/') + 1));
                                visual.addEdge(new Edge("Edge"), entry.getKey.substring(
                                entry.getKey().lastIndexOf('/') + 1),
                        vertex.substring(vertex.lastIndexOf('/') + 1));
                                visual.View();
                                        }
                                }
                        }
                }
```

```
/**
     * This method is used to add the key(person) and value(his friends) in hash
     * map
     * @param key
     * @param value
     */
    public void putMapList(String key, String value) {
        if (!nodeRelation.containsKey(key)) {
            Set<String> lst = new LinkedHashSet<String>();
            lst.add(value);
            nodeRelation.put(key, lst);
        } else {
            Set<String> lst = nodeRelation.get(key);
            lst.add(value);

        }
        if (key.equals(value)) {
            nodeRelation.remove(key);
        }
    }
```

```
/**
     * This method is used to write network information given by crawler
     *
     * @param key
     * @param lst
     */
    public void CrawlerNetwork() {
        Set<Entry<String, Set<String>>> set = nodeRelation.entrySet();
        for (Entry<String, Set<String>> entry : set) {
            String key = entry.getKey();
            Set<String> lst = entry.getValue();
            writeCrawlerNetwork(key, lst);
        }
    }
    public void writeCrawlerNetwork(String key, Set<String> lst) {
        StringBuffer buffer = new StringBuffer();
        for (String value : lst) {
            buffer.append(key.substring(key.lastIndexOf('/') + 1));
            buffer.append(" ");
            buffer.append(value.substring(value.lastIndexOf('/') + 1));
            buffer.append("\n");
        }
        write.CrawlerNetwork(buffer.toString());
    }
```

```
/* This method is used to calculate how many edges from participial boundary vertex to the
community*/
public int findTemB_in(String vertex) {
        int temB_in = 0;
            Set<Entry<String, Set<String>>> set = nodeRelation.entrySet();
            for (Entry<String, Set<String>> entry : set) {
                if (entry.getKey().equals(vertex)) { // compare key
                    Set<String> lst = entry.getValue();
                    for (String value : lst) {
                    if (community.contains(value)) { // compare value
                            temB_in += 1;
                        }
                    }
                }
                if (community.contains(entry.getKey())) { // compare key
                    Set<String> lst = entry.getValue();
                    for (String value : lst) {
                    if ((community.contains(value))&&(value.equals(vertex))) { //
                            temB_in += 1;
                        }
                    }
                }
            }
        return temB_in;
    }
    public int findTemB_out() {
        B_out_edges = unKnowArea.size();
        unKnowArea.clear();
        return B_out_edges;
    }
```

```
/**
 * This class has been used to visual data in the user graph interface.
 * @author miyun fan
 * @version 1.0
 */
// import file omitted
// methods omitted
/** This method is used to show the network
public void ShowNetwork(){
        mainFrame = new JFrame("Local Community Finder");
        mainFrame.setSize(900, 900);
        net = new NetworkElement();
        layout = new SpringLayout<String,Edge>(net.graph);
        layout.setSize(new Dimension(650, 650));
        vv = new VisualizationViewer<String, Edge>(layout);
        vv.setPreferredSize(new Dimension(630, 630));
        vv.getRenderContext().setVertexLabelTransformer(new ToStringLabeller());
        vv.getRenderer().getVertexLabelRenderer().setPosition(Position.S);
        vv.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY));
        Container content = mainFrame.getContentPane();
        final GraphZoomScrollPane panel = new GraphZoomScrollPane(vv);
        content.add(panel,BorderLayout.CENTER);
    }
    /** repaint the interface when new vertex has been added*/
    public void View() {
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.getContentPane().add(vv);
        mainFrame.repaint();
        mainFrame.pack();
        mainFrame.setVisible(true);}
    }
```

```
/** This method is used to add vertex in the graph
    public void addVertex(String vertex) {
        net.addVertex(vertex);
        endVertex = new Transformer<String, Paint>() {
            public Paint transform(String info) {
                return Color.RED;
            }
        };
        vv.getRenderContext().setVertexFillPaintTransformer(endVertex);
        getCurrentProgress();
    }
   /**Create the Swing frame and its content.*/
    private void makeFrame()
    {
        JPanel contentPane = (JPanel)mainFrame.getContentPane();
        contentPane.setBorder(new EmptyBorder(6, 6, 6, 6));
        makeMenuBar(mainFrame);
        // Specify the layout manager with nice spacing
        contentPane.setLayout(new BorderLayout(6, 6));
        //create the search and relevant text field
        JPanel searchPanel = new JPanel();
        searchPanel.setBorder(BorderFactory.createLoweredBevelBorder());
        JLabel progressLabel = new JLabel("Community Detecting Progress:");
        searchPanel.add(progressLabel);
        progressBar = new JProgressBar(0,120);
        progressBar.setMinimum(0);
        progressBar.setStringPainted(true);
        searchPanel.add(progressBar);
        JSeparator separator = new JSeparator();
```

```
searchPanel.add(separator);
contentPane.add(searchPanel, BorderLayout.NORTH);
statusLabel = new JLabel(VERSION);
contentPane.add(statusLabel, BorderLayout.SOUTH);
// Create the toolbar with the buttons
JPanel toolbar = new JPanel();
toolbar.setLayout(new GridLayout(0, 1));
toolbar.setBorder(BorderFactory.createTitledBorder("Control"));
smallerButton = new JButton("Smaller");
smallerButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { makeSmaller(); }
                });
toolbar.add(smallerButton);
largerButton = new JButton("Larger");
largerButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) { makeLarger(); }
                });
toolbar.add(largerButton); mouse = new JButton();
mouse.setLayout(new GridLayout(0, 1));
mouse.setBorder(BorderFactory.createTitledBorder("Mouse"));
vv.setGraphMouse(gm);
JComboBox modeBox = gm.getModeComboBox();
modeBox.addItemListener(gm.getModeListener());
gm.setMode(ModalGraphMouse.Mode.TRANSFORMING);
    mouse.add(modeBox); toolbar.add(mouse);
JPanel flow = new JPanel();
flow.add(toolbar);
contentPane.add(flow, BorderLayout.WEST);
mainFrame.pack();
}
```