# Abstract

The aim of this project is to track the changes of the community over time in order to detect the community evolution in the complex network. This project uses the co-authorship network in digital library as the experiment network.

During the study of the complex network, detecting and anaylsing the community (consisted by groups of vertices that are more densely connected than others) has becoming the study focus. The community is the miniature of the whole complex network, which is the basis of the understanding and analysing the complex network structure and related properties. However, in the last few years, people always pay attention to the structure of the community in the static complex network. Whereas most of the complex networks are the dynamic networks, therefore, just understanding the static network is not nearly enough. From the ever-changeable structure of network, people can efficaciously detect communities` structure and the evolution process, therefore, this project focus on extracting the communities and detecting the evolution of them.

The main tasks of this project include the following aspects:

- Firstly, author discuss several mainstream algorithms of community extracting, and introduce the CPM which is used in this project as the network extracting algorithm detailed.
- Secondly, according to the previous work in evolving community, author chooses one algorithm which is base on the core node detection algorithm to track the community evolution, this chosen algorithm is used to find the status of community in the next time interval. It has the following steps: (1) Finding the core nodes of the community. (2) Using the core nodes to establish the evolution relationships of communities in the adjacent time. This algorithm is best characterized by non-parametric calculation, and it can find the merging point and split point on the path of community evolution.
- Finally, using the above algorithms in the co-authorship network, the data is obtained from the journal (Research of Finance) in China digital library CNKI to find the community structure and evolution, analyse nodes in the community to find the changes. The experiment results show that the algorithm chosen above is suitable for the chosen co-authorship network, and easily to find the community structure and evolution process.

Also this project has been done successfully, there are still some shortages and improvement space. Shortages of this project and the suggestions of the future work will be also raised lastly.

# Table of contents

**Chapter 1: Introduction**.....1
1.1 Background.....1
1.2 Aims and objectives.....2
1.3 Appliance of complex network in the literature system.....3
1.4 Signification of the project.....5
1.5 Dissertation structure.....5
**Chapter 2 Complex network and Communities**.....7
2.1 Complex Network.....7
2.1.1 Basic concept.....7
2.1.2 Related property.....8
2.2 Communities.....11
2.2.1 The community in sociology.....11
2.2.2 The community in network.....12
**Chapter 3: Extracting community from network**.....13
3.1 Community finding algorithms.....13
3.2 Clique percolation method (CPM).....14
3.2.1 K-clique community.....15
3.2.2 Extraction of the k-clique-communities.....15
3.3 Project communities extraction frame.....17
**Chapter 4: Evolution of communities**.....20
4.1 Previous work in evolving communities.....20
4.2 Community evolution.....21
4.2.1 Evolution path.....21
4.2.2 The types of evolving communities.....22
4.2.3 Measurement of community evolution.....23
4.3 Core node detection.....25
4.3.1 Symbol definition.....25
4.3.2 Core node detection Algorithm.....26
4.4 Community evolution algorithm based on core node detection algorithm.....28
**Chapter 5: Experiment in co-authorship network**.....32
5.1 Purpose of experiment.....32
5.2 special works.....32
5.2.1 Experiment tools.....32
5.2.2 Experiments steps.....33
5.2.3 Data set.....34
5.3 Network construct.....34
5.3.1 Time division.....35
5.3.2 Obtained Networks.....35
5.4 Community extraction.....36
5.5 Community evolution.....38

5.5.1 Big communities..... 38
5.2.2 Small community..... 40
**Chapter 6: Summary and future development**.....41
**References**..... 43
**Appendix: Source code**

# Chapter 1: Introduction

## 1.1 Background

Large quantity of systems in the real world in which people live can be expressed as complex network, more over, most of these large quantity of complex systems are dynamic network, such as society network, co-actor network, Email network, neural network, co-authorship network, biological network, food chain network and etc.

In the complex network, every node represent an entity in the network, and the edge represents the relationship between two entities. In the co-authorship network, nodes represent authors, while the edges represent co-authorships that mean the two authors that linked by the edge co-author one paper in that network. More and more researches show that, although there are so many different types of complex network that may be refer to different subject areas, they have lots of the common statistical property: small-world effect, clustering features, scale-free and so on.

With the deep study of complex network in the physical and mathematical aspects, people find that many networks have the community property, that is to say, a network is consisted by lots of communities, nodes in the community are more dense, while the connection between communities is rarer relatively [25]. Usually, nodes in the same community have the same properties. In the co-authorship network, scientists in one community always have the similar study interesting, and in the WWW network, web page in the same community usually refer to the related topic. Finding the community structure in the network is very important for people to fully understand the network structure and analyse the network properties.

The community detection technology has already developed in recent years, GN algorithm, W-H algorithm, hierarchical clustering algorithm and etc. are the inchoate algorithms, however, with the passage of time, algorithms have been developed, and becoming more operable, that includes modified GN algorithm, Clique Percolation Method (CPM), Discovering Web Community algorithm and etc. These methods are

used in various kinds of network analysis [7].

Most of real networks are the dynamic network, new nodes will occur in some time while old nodes will disappeared in one community even in the network, take WWW network for example, webpage and links in the WWW network are updated every day. Such changes cause to the community change. In the network, new communities will emerge at any time, and similarly, old communities will disappear in the network, and the communities may shrink and also may grow. Detecting the evolution of the community in the dynamic network will help people to understand the development of the whole network and predict the changes of network, and also can realize the network structure optimization, resource search, resource recommendation and etc. Because the nodes in the complex network are very intelligent, the structure of network which is consisted by these intelligent nodes and community evolution are full of complexity, so how to find community structure and evolution process effectively is one of the difficulties in the complex network research area.

## 1.2 Aims and objectives

Finding the community in the complex network is usually very difficult, because of the complexity of network itself. The size of most real world network is very large and the structure of community in network is not very clear some time. The number of community in the network is unknown; the number of nodes in the community is unknown either. Moreover, the communities always change as the time goes on; thus there are some difficulties of community detection and evolution tracking.

The main difficulties of community detection and evolution tracking are:

- People always use the fixed parameters standard to track the evolving communities when analyse the communities in the different time interval in order to compare the changes of communities in the same condition. However, in the community evolution, the communities` structure always changes over time; hence it is hardly to get the community evolution path with the fixed parameter efficiently.
- It is hardly to measure the evolution types of the community, because that the

community changes are always multiple, just using birth, shrink, disappearance, etc. cannot express the evolution accurately.

Therefore, considering the above difficulties, author chooses the proper algorithm (CPM) to extract communities, and uses an algorithm based on the core node detection algorithm to track the community evolution.

The primary aim of this research is to use the property algorithms to investigate how communities in the complex network evolve over time, such as shrinking, growth, disappearance, birth, etc.

According to the aim, objectives are broken down as following:

- Study algorithms that is used to community extracting, and find the proper community extracting algorithm for the experiment network (co-authorship network)
- According to the previous work in evolving community, choose one efficient algorithm as the community evolution finding method for this project.
- Data collection and build the co-authorship network as the experiment network, using the chosen algorithm (CPM) to finding the community.
- Using the algorithm, which is base on the core node detection algorithm to track the community evolution, and then inspect the evolving communities.

## 1.3 Appliance of complex network in the literature system

With the increasing of capacity of the storage equipment, people entered upon an age of Mass memory. Thus people are confronted with large quantity of scientific and technical literature, but how to store, inspect and exhibit this literature efficiently has becoming a challenging and important task. Because of the lack of capability of handle such large quantity of data, most existing literature systems can only handle the data simply; more over, the relationships among the data are always ignored. To illustrate, when a person searches an author's paper, most of existing system can only show the papers that related to that author, instead of showing the relationship between that author and others. Thus system cannot help users to get full understanding of the information. In order to solve the above problem, the study of

co-authorship network is badly needed.

The more perfect system should include the following functions: when a user search an author, the system could show the research community that include that author, more over, that system can also tracks the dynamic changes of that community, provide the research topic in some research area, and show the changes of the topic over time, in order to help users to learn the research topics of that area in a short time; and when a user search a paper, the system can not only show the paper that user needs, but also recommend some more important literature in that area to the user.

There have already some data mining algorithms to cope with the above function requiring, however, most of these algorithms are not very efficient, and have the long runtime. According to these shortages, people turn to use the complex network theory to analyse the literature data. The analysis method that bases on the complex network is best characterized by using graph theory that is different from the traditional data mining method to find the community structure, and it is faster and more accurate.

Mane and Börner use Kleinberg's burst detection algorithm, co-word occurrence analysis, and graph layout techniques to find and show the outburst topics (research topics in a certain time period) that is included in the PNAS literature [14], the system can give an efficient guide to users by discovering the research topic, therefore, users can know the interesting of scientists in a certain time period easily.

White et al. developed a web-based literature-mapping system that is used to mapping the literatures that published in PNAS during 1971 to 2002 [26]. The biggest feature of this system is that: when users put in a key word (can be the author's name, paper title, etc.), this system can not only show the related information about this key word, but also can show the other 24 key words which closely connect with the input key word; therefore Howard et al. provide the interrelationship of key words to users efficiently through this system.

## 1.4 Signification of the project

Community is the miniature of the network, and it is the basis of understanding network structure and analysing network properties. Studying the evolving

communities in the network is significant to analysing the network properties and the dynamic change tendency of the whole network. The project has the theory and appliance significations.

- Theory signification

The study of statistical properties of network is an interested task for lots of scientists all the time. Macroscopic properties of network is based on the whole network while the microcosmic properties is based on the nodes in network. Thus the community structure is one of the network properties between macroscopic and microcosmic properties, the community evolution express the development of dynamic network. Therefore, the study of evolving communities in the network is crucial for understanding the functions of nodes and the relationship of them in the network.

- Appliance signification

The complex network is the expression of a set of interrelated entities and their relationships. The form of expression is used in lots of real world system, the project has widespread appliance, and it can provide the method for realizing network structure optimization, resource search, resource recommendation and etc.

## 1.5 Dissertation structure

There are six chapters in this dissertation.

- The first chapter is the introduction, which introduces the background of this project, complex network in literature system, research aims, objectives and the significations of this project.
- In the second chapter, author introduces the concept of complex network and communities.
- Third chapter introduces the existing community extraction method and give a detailed introduction of CPM, which is used as the community extraction method in the project.
- The fourth chapter gives the community evolution tracking algorithm which is

based on the core node detection algorithm, the algorithm is best characterized by non-parametric calculation, and it can find the merging point and split point on the path of community evolution.

- In the fifth chapter, author constructs a co-authorship network; data is obtained from the China digital library CNKI. Using the above algorithms to extract the communities and find the community evolution. Analysing the evolving communities to learn the properties of network.
- The last chapter is the conclusion and the future work.

# Chapter 2 Complex network and Communities

## 2.1 Complex Network

### 2.1.1 Basic concept

In the real world, systems can be expressed as complex network, if they refer to the relationships among entities. In the network, every node represents an entity, and the relationship between entities is represented as the edge in the network, such as WWW network, co-actor network, Email network, neural network, co-authorship network [15], biological network, food chain network. Complex network can be described as $G = (V, E)$, where V is the set of nodes and E is the set of edges, every edge in E has two nodes in V.

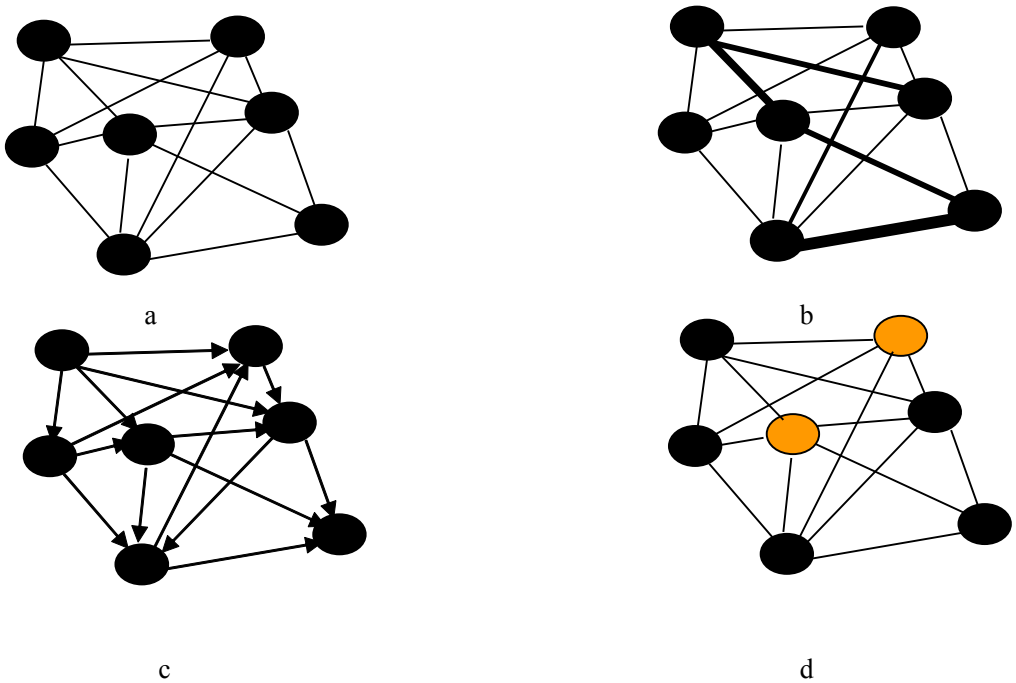



Figure 2-1 different types of network

a: undirected network. b: directed network. c: weighted network. d: heterogeneous network

According to the property of E and V, there are many types of network. If every set of V, (i,j) and (j,i) has the same edge, then the network can be named as undirected network, otherwise, the network is the directed network. For example, co-authorship network, phone network co-actor network are the undirected networks, and food chain network, Email network neural network are the directed networks. Every edge in E can be represented by the common property, such as relationship type and weighing. If every edge has been weighted, then the network is the weighted network; otherwise, the network is the unweighted network. Take neural network for example, weighting represents the connection intensity between two neurons. In some networks, there are many types of nodes and edges sets, for example, web data network contains large different types of resources and relationships, every node in the network represents one resource, the edge between resources represents RDF link, therefore, this network called heterogeneous network.

## 2.1.2 Related property

Plenty of studies found that, some networks such as co-authorship network is greatly difference of random network.

### 1 small-world effect

Even though the co-authorship network is very huge, the average path length of network, which is defined as the average distance between any two nodes in network, is real small.

$$L = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i \geq j} d_{ij}$$

$d_{ij}$ represents the distance between the nodes i and j

The small-word is firstly proposed by Hungarian--F.Karinthy in 1929, he thought any two persons could be linked by average six lines that were composed by six persons [5]. After that, in 1967, Milgram who are the social psychologist of Harvard

University using Email experiment proposed the six degrees of separation that confirm the property of small-world [16]. He chose 300 volunteers randomly, and let them to send an Email to a certain stockbroker; the rule is that volunteers should send the Email using the shortest delivery path that they thought the path is. Surprisingly, more than sixty Emails sent to the appointed stockbroker, moreover, the delivery path of these sixty Emails are only five in average. The experiment shows that, every two strangers in the world can be connected by six people in average.

Hereafter, people use other method to test the six degrees of separation hypothesis. Tjaden and Wasson designed a game named Kevin Bacon [21]. The game shows the small-world effect in the co-actor network. A Bacon number of an actor is the shortest path from this actor to Kevin Bacon, so if one actor has cooperated with Kevin Bacon, then the Bacon number of the actor is one. If an actor has not cooperated with Kevin Bacon but has cooperated with another actor who has cooperated with Kevin Bacon, then the actor's Bacon number is two, and so on. The experiment shows that in this co-actor network which contains six hundred thousand actors, the biggest Bacon number is only eight, and the average Bacon number is 2.944. Similarly, professor Grossman and Ion in university of auckland proposed the Erdos number project tested the small-word in collaboration network of mathematician [10], and Elmacioglu and Lee do some research on savant collaboration network based on the construction of DBLP data set [9], and test the six degrees of separation phenomenon.

All these studies stated clearly that, in the huge network, the distance among individuals is real close.

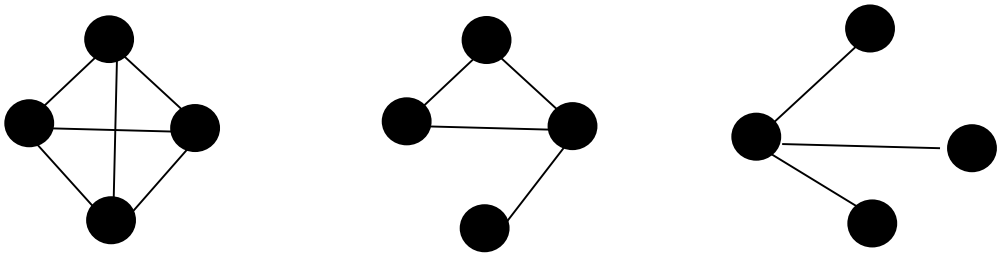
## 2 Clustering features

In the co-authorship network, if a node i is directly linked by edges with other $k_i$ nodes, we call the node i is adjacent to these $k_i$ nodes, however, $k_i$ nodes may have the relationship of adjacency with each other. This phenomenon in the network called clustering property. Consider that in the friendship network, two friends of a person may be the friends each other [12].

One of the most important metric of network is clustering coefficient (cc). In co-authorship network (unweighted network), the clustering coefficient of node i is defined as:

$$C_i = \frac{2\mathrm{E}_i}{k_i(k_i - 1)}$$

$E_i$ is the number of edges which connect i with $k_i$ . Clustering coefficient of the network is the average value of all nodes` clustering coefficient.



C=1 C=1/3 C=0

Figure 2-2 cc in the unweighted network

Apparently, if every node in the network is isolated, the cc of this network is zero; and if any two nodes are adjacent nodes, the network is the complete graph; therefore the cc of this network is 1. In general, the cc of most networks is between 0 and 1. For the co-authorship network, the cc is close to a nonzero constant coefficient as the size of network increasing.

## 3 Scale-free

The degree of node i in network is defined as the number of nodes which is adjacent to i. In the large scale random network, the degree of node is shown as normal distribution, while in the regular network, the degree of node is constant, the distribution of nodes in both above networks has the rule, that is to say, nodes in the above networks are distributed in a certain range and there are not one node of which the degree is much lager than the average degree, so the network is called scale network. As plenty of researches show that the degree of nodes that in most real network such as WWW [2], e-mail network [8], co-authorship network coincides the power law distribution $P(k) \propto k^{-\gamma}$ (in other words, scale-free distribution), so these network are scale-free networks.

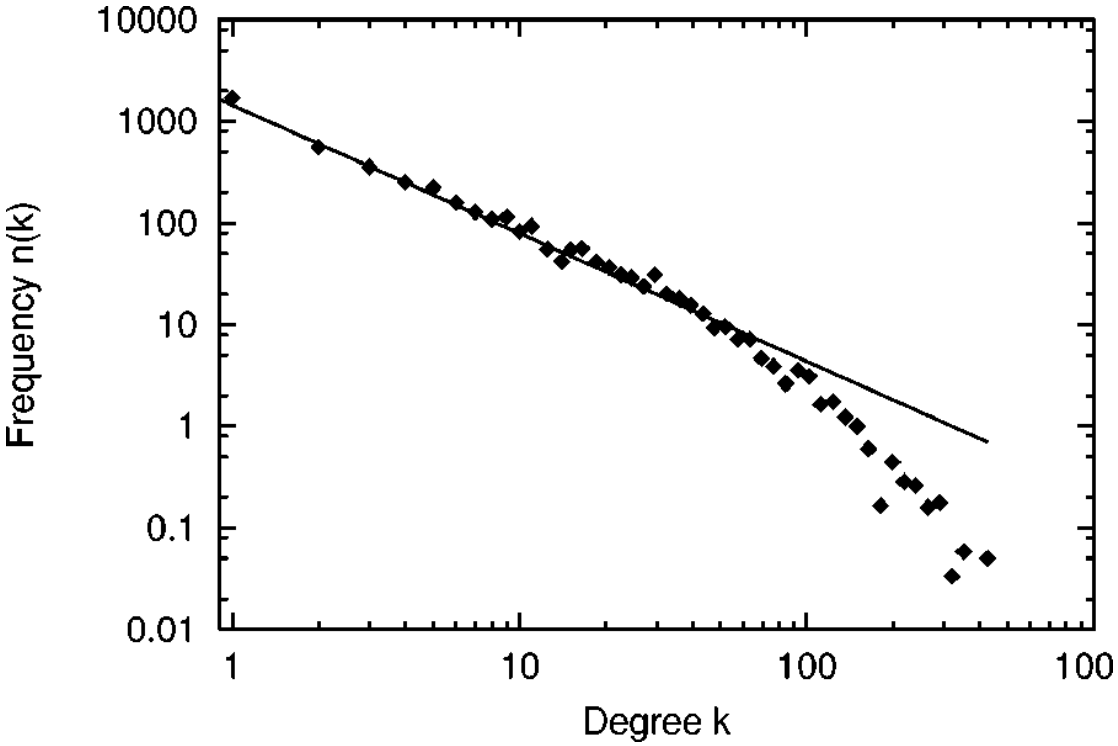



Figure 2-3 degree distribution of e-mail network(from [8].)

Figure 2-3 describes the degree distribution of e-mail network. In this network, the node represents the e-mail address, and every edge represents the relationship between two e-mail addresses, and the degree of nodes in this network coincides the power law distribution, $P(k) \propto k^{-1.8}$. We can easily find that most of nodes have small degree value, and there is a little node that has great degree value.

## 2.2 Communities

### 2.2.1 The community in sociology

Sociologist F.Tonnies was the first man who proposed the concept of “community”, and gives the detailed explanation in the book- Gemeinschaftund Gesellchaft that published in 1887. The community means any organism that has a collaborative relationship. In Tonnies opinion, the wide range of "Geneinschaft" meaning is not only contains the geographically-based communities, but also blood relationship,

spiritual relationship and so on, and the soul of the community is the collective sense of cultural awareness. With the development society, the study of community raised the sociologists' interest.

There are two general types of definitions of community in sociology, one is emphasize the functionality, they think the community is composed of persons who have the same goal and interested party; another is defined as geographically-based communities, they divide the society into many pieces that include people who work, study, live (etc.) together.

### 2.2.2 The community in network

Community is the common property in the network. It is the basic unit of the network. However, there is not an explicit definition for community yet. An accepted opinion is that: a community is the network's subgraph in which nodes tend to linked with other nodes in the same subgraph. Hence, the nodes in the community link stronger with each other. Take the co-authorship network for example, authors in the same community has the similar research interests. Therefore, it is very important for learning and analysing the network to study the community.

In fact, the community cannot be found easily in most real network; thus we need to use some community detection method to dig the hidden community in network.

The community detection means the process of assigning the nodes in the network to several subsets by types.

# Chapter 3: Extracting community from network

Community structure is one of the important properties in network. In term of topology, nodes in the network tend to link with other nodes that exist in the same community; that is to say, nodes in the same community are denser; thus the density in the community is higher than that among communities. In terms of property, nodes in the same community also have the same or similar properties; therefore, study on community construct is very helpful for people to understand network structure and analysing the network properties. Nowadays, there are several types of community finding algorithms, such as Kernighan-Lin algorithm, spectral bisection method, GN algorithm, CPM.

## 3.1 Community finding algorithms

Finding the community is very important for studying the relationship between structure and function of network, therefore, the analysis of community is one of the research topics in complex network all the time. Community analysing technology has the close relationship with graph partition in computer science and hierarchical clustering in sociology.

Graph partition intends to divide nodes in the network into several node set, and it requires that there should be the least edges among communities. However, it is a NP -hard problem, in order to try to solve it, people raise several algorithms among which there are two famous types: Kernighan-Lin algorithm and spectral bisection method. Kernigham-Lin algorithm is based on the greedy algorithm [29], it intended to divided the network into 2 given size communities, therefore, Kernighan-Lin algorithm requires to know the size of communities in advance, however people can not know the size before knowing the community, this shortage pose a bad influence to the valve of algorithm usage. Complexity of spectral bisection method which is based on Laplace matrix eigenvalue [23] and Wu-Huberman algorithm are lower than Kernighan-Lin algorithm [27], but they require to know the number of communities

in advance, however, in the real world network, the number of communities is not given in advance. In order to overcome these shortages, Capocci et al. modified the traditional method and proposed one new spectral bisection method which is based on standard matrix [6], this new method can be used in the network in which the structure of communities is not obvious. All the above algorithms have a common shortage, algorithms divide network just into two parts, if people want to divide network into several parts, they need to run the algorithm many times.

Hierarchical clustering is a traditional method in social network, it bases on the similarity and intensity of relationship between nodes to divides network. There are two types of clustering algorithm: splitting algorithm and condensation algorithm. GN algorithm is the representative at splitting algorithm, its mean idea is to delete the edge that have the biggest betweenness. The shortage of the GN algorithm is the repeated calculation. Modularity has been proposed by Newman et al. as the quality standard of network division. Tylar et al. point that using a certain set of nodes instead of all nodes in network to calculate edge betweenness, so the calculating speed has been improved greatly by this method. In 2004, Newman et al. proposed one fast algorithm which is based on greedy condensation algorithm, newman fast algorithm is suitable for large complex network, however, in 2005, Clauset et al. modified this algorithm and used it to analyse the web page link network of Amazon on-line book store successfully.

Both splitting algorithm and condensation algorithm divide network into separate community [25], but the communities in network are correlative and overlapping in most time, according to the above requirement, Palla et al. raised clique percolation method (CPM) which can detect the overlapping communities [17][18], CPM is widely used in the co-authorship network, phone network, protein network and so on.

## 3.2 Clique percolation method (CPM)

Clique percolation method is one of newer communities detection methods, it differ from the traditional method which used to divide network into separate communities, CPM is used to analyse the structure of communities that overlapping with each others. For example, the co-authorship network is composed by many overlap

communities, and authors is divided by the study interest in co-authorship network, now the author i is interested in several study areas, therefore i belongs to several communities. In that case, it is real hard to extract the community, so we need to use CPM in co-authorship network to extract communities.

### 3.2.1 K-clique community

The Clique percolation method is proposed by Palla and his co-workers in 2005 [17], they think the network can be thought as the set of small complete networks in which each node is connected to every other node, and the small complete network is called clique, if there are k nodes in it then it called k-clique. Two k-cliques are adjacent only if they share k-1 nodes. If one k-cliques can reach to another k-clique through several adjacent k-cliques (part of k-clique chain), then these two k-cliques are called k-clique connectedness. Provided that some nodes are the nodes that exist in several k-clique which is not adjacent(having no k-1 public nodes), then this nodes are called the overlap part of these k-cliques; thus we can use CMP toadjacent (overlapping communities in the network [22]. In figure 3-1, there are two 3-clique-communities, and they are colored by black and grey separately.

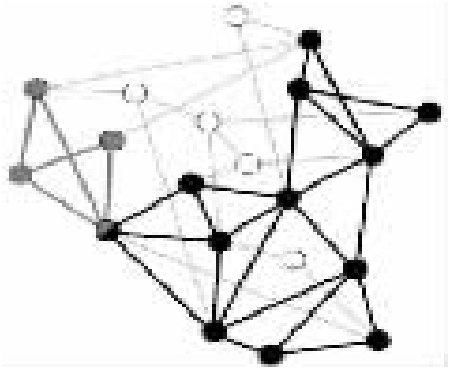


Figure 3-1 3-clique-community(from [7])

### 3.2.2 Extraction of the k-clique-communities

Searching k-clique in the network is the necessary condition for finding the k-clique community. In the CPM, we search the k-clique from the big nodes (largest degree) to

small ones, firstly, looking for the possible biggest k-clique s from checking the degree of every node in the network. Then, choose node i and find the k-clique of size s which contains the node i. Next, remove the node i and the edges that link with i in k-clique, and choose another node i` to do the same thing as node i. Repeat these three steps above until finding all s-clique, then let s to be s-1, repeat all of the above steps, until finding all k-clique in every size.

The method of finding s-clique can be illustrated as following:

First, define two set name A and B for node v, and set A contains nodes that connect with each other, and B contains nodes that connect with nodes in set B.

Initialization, A= {v}, B={neighbor of v}

- Selecting a node randomly in B, and move it to A. Then delete nodes in B, the nodes deleted are those who are not connected with every node in A.
- If |A|<s and |B|=0, or, A or B is the subset of some existing clique, then stop and return to the first step of the loop. Otherwise, if |A|=s, then recode A, and return to find new cliques.

In this way, all the s-cliques that begin with v can be found.

After finding all the cliques, people can get the clique-clique overlap matrix, every row (column) represents a k-clique in matrix, and the members in the matrix are number of the common nodes of the corresponding clique. K-clique-community is found using this overlap matrix, if the number in the matrix is lesser than k-1 then set it to be zero, while set others to be one. After that, people obtain the k-clique matrix, and every connectedness part is the k-clique community.

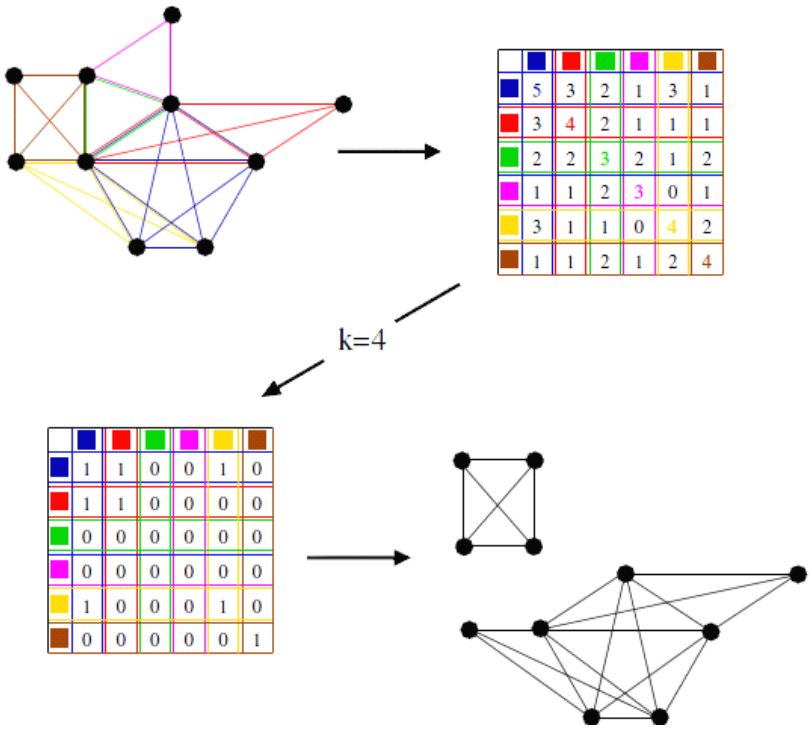



Figure 3-2 Extraction of 4-clique-communities (from [17])

In the top left figure, there are six cliques colored with six colors, the right of cliques are obtained clique-clique overlap matrix, in order to get 4-clique-communities, let k equals to 4, then modify the overlap matrix, check the diagonal elements and set ones bigger than 4 to be 1 others to be 0, set the off-diagonal elements that smaller than 3 to be 0 other to be 1. After that, the matrix is shown like the picture in the bottom left corner of the picture. From this modified clique-clique overlap matrix, two 4-clique-communities are obtained and shown in the bottom right.

## 3.3 Project communities extraction frame

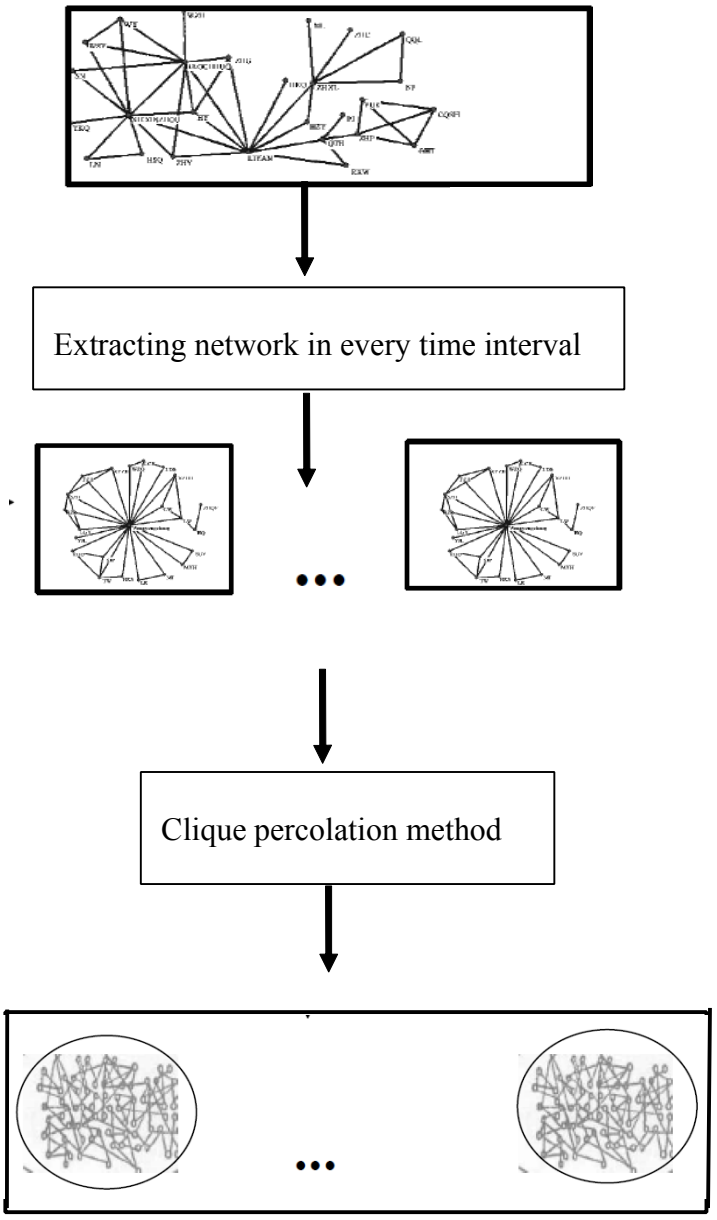



Figure 3-3 frame of communities extraction

The figure above shows the overall frame of community extraction, and the general flow is described as following:

(1) Getting the whole network data set in the research time interval.

(2) Exacting the network segment in every time interval.

(3) Analyse the topology of the whole network using the network segments.

(4) Using the clique percolation method to extract the communities.

# Chapter 4: Evolution of communities

In general, most real life networks including co-authorship network are very huge and dynamic.however the numbers of community and members in community could not be known in advance. Moreover, the network changes every often, the change of community structure have large influence upon the network structure, as new communities will emerge, old communities will disappear and some communities will merge with other communities.

## 4.1 Previous work in evolving communities

Nowadays, some researches notice that there is the relationship between changes in dynamic network and community evolution detection, and try to analyse the evolving community to analyse, explain and forecast the structure and properties of dynamic network.

Toyoda and Kitsuregawa do the research about evolving community of web[24], an web community is composed by several web pages, which are linked by hyperlink, moreover, the topical subject of that web pages in the same community is always the same, therefore, the changes of community show the evolution of topic. The researchers extracted the community from the network that is built form the web archives of web pages in the year of 1999, 2000,2001 and 2002, after that, they built the community chart and analysed the process of web community evolution and the phenomenon of community emerging. They defined several attributes to describe the evolution of the dynamic network when analysing the community evolution, the attributes are: growth rate, stability, novelty, disappearance rate merge rate and so on. From the experiment, they found that, the size of community and any types of changes of community coincided the power law distribution.

Kumar et al. analyse the evolving community of blog network, which has 250 thousand blogs and 750 thousand hyperlink[11]. They defined the temporal graph which is the extension of traditional network graph, and use the pruning and

expansion algorithm to extracting community, in order to analyse the phenomenon of bust in the network. From the research, they found that the number and size of the community in blog network increased obviously.

Palla, Barabasi and Vicsek analysing two type of network to study community evolution[19]. One of them is the co-authorship network which contains 30 thousand relationship of collaboration, and spanning 142 months; another is the communication network, which contains four million call records of phone users, spanning 52 months. Firstly, they extracted the community in every time interval using CPM algorithm. Then, they built an joint network from two adjacent time interval, and find the relationship of communities in the adjacent time interval using the joint network, thereby getting the process of community evolution. They defined the correlation and nationality to describe the community evolution. From the experiment, they found that, the change rate of members in the lager communities is higher than others in the social networks.

Lin et al. raised the Facetnet frame[13], which is used to analyse the community and community evolution in the dynamic network, and analysed the data set of Nec Blog from Aug. 2005 to Sep.2006, and DBLP data set. They raised the conception of community network, evolving network and so on, and analysed the community evolution.

Baumes et al. [3] raised the ViSAGE system, in order to simulate the community evolution in the social networks, and analyse the property and phenomenon of community emerging, thereby forecasting the possible change of network. ViSAGE contains four types of model, they decided that there is no new community added and old community disappeared in the simulated network.

## 4.2 Community evolution

### 4.2.1 Evolution path

The evolution path of community shows the change process of communities in the network. The evolution path can be described as: Evol=(TS,{C},ER), where TS is the

ascending time series S1, S2, ... ,St. $\forall t_s \in S_x, \forall t_{s+1} \in S_{x+1}, t_x < t_{x+1}$

{C} is the set of communities in network segment set in the corresponding time interval.

ER is the evolution set, ER ⊂{C} ×{C} .（Ci ,Cj ）∈ER means Ci and Cj are the two status of the same community in the adjacent time interval, if the T(Cj)=T(Ci)+1, then we call Ci is the predecessor of Cj, and Cj is the successor of Ci. In the community evolution path, there may be no predecessor(successor) of a community, or more than one predecessor(successor) of one community.

### 4.2.2 The types of evolving communities

According to the existence of predecessor or successor, the process of community evolution can be divided into several types: emerging(birth), disappearance(death), growth, contraction, merging,splitting.

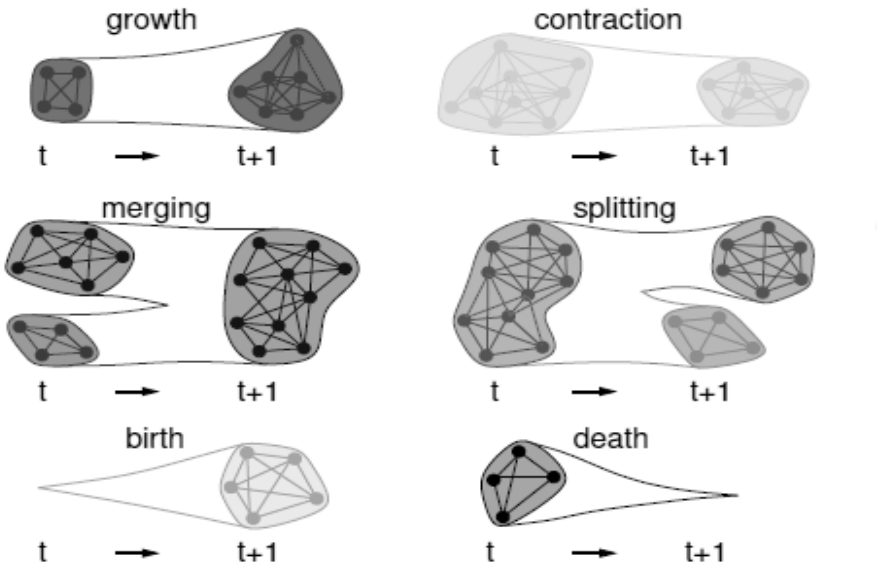



Figure 4-1 The change of community (from [20] )

- **Birth**

Community c emerges in the time T(c)+1, only if c does not exist in the network in

the time T(c), in other word, there is no predecessor of c in the community evolution graph $EG = (TS, \{C\}, ER)$, $\forall c` \in C, (c`, c) \notin ER$.

- **Death**

Community C dies in the time T(C)+1, only if C does not exist in the network in the time T(C)+1, in other word, there is no successor of c in the community evolution graph $EG = (TS, \{C\}, ER)$, $\forall C` \in \{C\}, (C, C`) \notin ER$.

- **Other changes**

Community C has successor in time T(C)+1, $EG = (TS, \{C\}, ER)$, $\forall C` \in \{C\}, (C, C`) \in ER$. Moreover, the successor can be divided into stable and changeable status according to the difference between C and it`s successor. If C keeps stable, then in the graph $EG = (TS, \{C\}, ER)$, $\forall C` \in \{C\}, (C, C`) \in ER$ and C`=C, that is to say, no member leaves or enters the community. If the community changes in the next time, then we should analyse the changes in order to know the community tendency. The changes include: growth, contraction, merging and splitting. However, the change of community is the combination of several basic types in the most cases.

### 4.2.3 Measurement of community evolution

We defined some attributes to measure community evolution, they include:

Stability, growth rate, novelty, disappearance rate, split rate, merge rate and so on [24].

$C_i$ and $C_j$ is the adjacent community status of the same community and $T(C_i)=T(C_i)+1$, i.e., EG=(TS,{C},ER), $(C_i, C_j) \in ER$.

- **Stability**

The stability of community $C_i$ can by defined as：

$$R_{stability}(C_i, C_j) = \frac{|C_j \cap C_i|}{|C_i|} \tag{4.1}$$

The higher stability of community, the more number of nodes that stay in the community, if the $R_{stability}$=1, then we call the whole network is the static network in that time.

- **Disappearance rate**

The disappearance rate of community means the members in the community disappear proportionately. In the range of $T(C_i)$~$T(C_j)$,

$$R_{disappear}(C_i, C_j) = \frac{|C_i - C_j|}{|C_i|} \tag{4.2}$$

$R_{disappear}$ is proportional to the number of memberswho disappear from community, while $R_{disappear}$ is inversely proportional to the size of community itself.

- **Growth rate**

The growth rate of community is the proportion of new members . In $T(C_i)$~$T(C_j)$

$$R_{grow}(C_i, C_j) = \frac{|C_j - C_i|}{|C_i|} \tag{4.3}$$

$R_{grow}$ is proportional to the number of new members of community, while, $R_{grow}$ is inversely proportional to the size of community.

- **Rate of community change**

The changes of member in the community shows the changes of community in the process of evolution, In $T(C_i)$~$T(C_j)$

$$R_{change}(C_i, C_j) = \frac{|C_j \oplus C_i|}{|C_i|}, C_j \oplus C_j = (C_j - C_i) \bigcup (C_i - C_j) \tag{4.4}$$

$R_{change}$ is proportional to the number of new and disappeared members in community,

while, $R_{change}$ is inversely proportional to the size of community.

- **Split rate**

The split rate of the community means the proportion of members who leave to other community. In $T(C_i)$~$T(C_j)$:

$$R_{\text{split}}(C_i, C_j) = \frac{|split(C_i, C_j)|}{|C_i|} \quad split(C_i, C_j) = \bigcup_{v \in c_i - c_j} (\exists C \neq C_j, st \ T(C) = T(C_j) \text{ and } v \in C)$$

(4.5)

- **Merging rate**

The merging rate of the community shows the merging degree of this community nwhich may be merge with other nodes in other communities. In $T(c_i)$~$T(c_j)$, author defines the merging rate of one community as following:

$$R_{\text{merge}}(C_i, C_j) = \frac{|merge(C_i, C_j)|}{|C_j|}, \quad \text{merge}(C_i, C_j) = \bigcup_{v \in c_j - c_i} (\exists C \neq C_i, st \ T(C) = T(C_i) \text{ and } v \in C)$$

(4.6)

- **Community age**

The age of community is the lifespan of it. In the community evolution path Evol=(TS,{C},ER），the longest one $C_{k1}, C_{k2}, \cdots, C_{kL}$ represents the community lifespan, and in this example, the community lifespan is L

## 4.3 Core node detection

### 4.3.1 Symbol definition

For the co-authorship is the dynamic network, author adds the time property in the network description; therefore, the definition of co-authorship network with time property will be described as:

$$G= (T,W,V,E) \quad (4.7)$$

T is the time set which presents the time interval.

W is the weight set(edges in unweighted network have the same weighing value 1).

V presents the nodes in the network, $V=\{v_1, v_2, ..., v_N\}$ , where N presents the number of the nodes, i.e. the size of G,|G|=N.

E means the edge set, $E=\{e_1, e_2, ..., e_M\}$, where M presents the number of the edges in the network.

Every edge $e \in E$ can be describe as: ( $v_i$ ,$v_j$, t, w), where $v_i, v_j \in V, t \in T, w \in W,$ that means nodes i and j have the relationship in the time t, and the degree of the relationship is w.

| Symbol | Definition |
|---|---|
| C | Community |
| $C_i^t$ | Community i in the time interval t |
| E( $C_i^t$ ) | The set of edges that in community $C_i^t$ |
| V( $C_i^t$ ) | The set of nodes that in community $C_i^t$ |
| \|V( $C_i^t$ )\| | Number of nodes in community $C_i^t$ |
| Cen(v) | Core degree of node v |
| Core( $C_i^t$ ) | The set of core nodes in community $C_i^t$ |

### 4.3.2 Core node detection Algorithm

The purpose of the core node detection algorithm is to find the core node in the given community. The following algorithm is the general calculation frame, in the algorithm, in this project D(v) is the degree of a node.

Input parameter: the given community

Output result: core node set of community.

```
1:   procedure Core Detection
2:   if D(v1)=D(v2)=...= D(vN) then
3:       return C
4:   end if
5:   Cen(vi)=0 i∈[1,N]
6:   for every edge e ∈E(C) do
7:       vi and vj are nodes connected with e
8:       if D(vi)<D( vj ) then
9:           Cen(vi )=Cen(vi )-|D(vi )-D(vj )|
10:          Cen(vj )=Cen(vj )+|W(vj )-W(vi)|
11:      end if
12:      if D(vi )≥ D( vj ) then
13:          Cen(vi )=Cen(vi )+|D(vi )-D(vj )|
14:          Cen(vj )=Cen(vj )-|D(vj )-D(vi)|
15:      end if
16:  end for
17:  corset = {}
18:  for every node vi ∈V(C) do
19:      if Cen(vi)≥ 0 then
20           input vi  into coreset;
21:      end if
22:  end for
23:  return coreset
24:   end procedure
```

Through the algorithm, the returned result coreset is the Core( $C_i'$ ) of given community. For all the nodes $v_i$ in community, the core degree of nodes is calculated by comparing the node degree with their neighbours: if $D(v_i)$ is lager than $D(v_j)$ $(i \neq j)$ ,

then the node $v_i$ is more important than $v_j$ in the given community, otherwise, $v_j$ is more important than $v_i$ . The algorithm use $|D(v_i)-D(v_j)|$ to show the quantitative difference between two nodes, and $v_i$ compares degree with all adjacent nodes, if $D(v_i)$ is lager than 0, then the node $v_i$ is one of the core node of the given community, otherwise, $v_i$ is a general node.

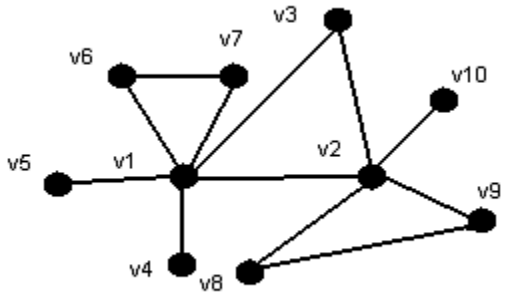



Figure 4-2 Core node detection

According to the algorithm the above community has the core node set$\{v_1, v_2\}$.
$Cen(v_1)=23$, $Cen(v_2)=12$, $Cen(v_4)=Cen(v_5)=-5$, $Cen(v_6)=Cen(v_7)=-4$, $Cen(v_8)=Cen(v_9)=-3$, $Cen(v_3)=-7$.

## 4.4 Community evolution algorithm based on core node detection algorithm

Berger-wolf and Saia [4] proposed a method that using the overlapping degree between two communities to establish the evolution relationships. However the method needs to choose a threshold s, if the $\text{overlap\_degree}(C^{(t)}, C^{(t+1)}) \geq s$ , then these two communities $C^{(t)}$ and $C^{(t+1)}$ has the evolution relationship, and $C^{(t+1)}$ is the successor of $C^{(t)}$ . However, it is very hard to choose a proper threshold, if setting s to a lager value, the algorithm may not find the successor of a community $C^{(t)}$ if there is a great change in $C^{(t+1)}$ , while if setting s to a smaller value, then community $C^{(t)}$

may obtain so many successors which have weak correlation with $C^{(t)}$.

Palla and Barabasi intend to track the community evolution using edge overlap [19]. However, the method cannot find the splitting evolution. If the edge_overlap_degree( $C_i^{(t+!)}$, $C^{(t)}$ ) is lager than the edge_overlap_degree( $C_j^{(t+!)}$, $C^{(t)}$), then $C_i^{(t+!)}$ is the successor of $C^{(t)}$, however, if $C^{(t)}$ splits into two communities $C_i^{(t+!)}$ and $C_j^{(t+!)}$, then the algorithm will lose one successor.

According to the shortage of the above algorithm, author chooses an algorithm base on core node detection. The algorithm do not require the threshold and easily to find the successors and predecessors.

Cause the stability of core in communities, there must be at least one core node in the successors, and for every core node, no one can burst in, core nodes must transform from general ones. Therefore $C_j^{(t+1)}$ and $C_i^{(t)}$ have the evolution relationship if and only if the following two conditions hold at the same time[30].

(1) there must be at least one core node of $C_i^{(t)}$ in $C_j^{(t+1)}$ : $Core(C_i^{(t)}) \cap Node(C_j^{(t+1)}) \neq \Phi$

(2) there must be at least one core node of $C_j^{(t+1)}$ in one of ancestry communities of $C_i^{(t)}$: $Node(C_x^{(t-m)}) \cap Core(C_j^{(t+1)}) \neq \Phi$.

---

Input parameter: the given $C_i^{(t)}$ and all the communities in the time interval t+1

Output result: Evol( $C_i^{(t)}$ )

1: Evol( $C_i^{(t)}$ )={ $C_i^{(t)}$ }

2: Core( $C_i^{(t)}$ )= CoreDetection( $C_i^{(t)}$ )

3: **for** every community $C_j^{(t+1)}$ in snapshot t+1 **do**

4: Core( $C_j^{(t+1)}$ )=CoreDetection( $C_j^{(t+1)}$ )

5: **if** $Core(C_i^{(t)}) \cap Node(C_j^{(t+1)}) \neq \Phi$ **and**

$Node(C_x^{(t-m)}) \cap Core(C_j^{(t+1)}) \neq \Phi$ **and**

$Node(C_x^{(t-m)})$ is one of ancestry communities of $C_i^{(t)}$ **then**

6: establish the relationship $C_i^{(t)} \longrightarrow C_j^{(t+1)}$

7: Evol( $C_j^{(t+1)}$ )= Community Evolution ( $C_j^{(t+1)}$ )

8: Evol( $C_i^{(t)}$ )= Evol( $C_j^{(t+1)}$ ) $\cup$ Evol( $C_i^{(t)}$ )

9: **end if**

10: **end for**

11: **return** Evol( $C_i^{(t)}$ )

---

If community $C_i^{(t)}$ has more than one successors, then $C_i^{(t)}$ is considered as a split point in its evolution path; and similarly, $C_i^{(t)}$ is considered as a merging point if it owns more than one predecessor. If there is no successor of $C_i^{(t)}$, then the evolution path of $C_i^{(t)}$ stops at the time t, and if no predecessor of $C_i^{(t)}$ in network, then $C_i^{(t)}$

is a new $C_i^{(t)}$, and is born in the time t.

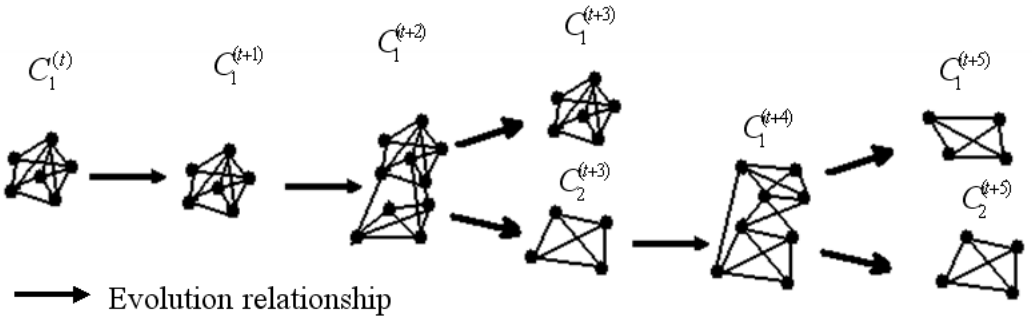



Figure 4-3 An example of community evolution path

Figure 4-3 shows the Evol($C_i^{(t)}$) of community $C_i^{(t)}$, in the evolution path, $C_1^{(t+2)}$ and $C_1^{(t+4)}$ are the two split points, and the length of Evol($C_i^{(t)}$) is 6.

# Chapter 5: Experiment in co-authorship network

## 5.1 Purpose of experiment

With the deep study of each research area and the interpenetration among some study areas, more and more new technologies are raised by researchers, moreover, the research direction and interesting topic of researchers are changed all the time. Although researchers can get lots of academic information, such as author, paper, conference, magazine, citing relationship, from literature database (ACM digital library, IEEExplore, ScienceDirect, etc.), getting the research trends, development history, and the least topic is not easy for researchers, especially for beginner. According to these difficulties, people study co-authorship network, and try to solve them. Co-authorship network can show the relationship between authors; therefore, it is easy to show the research interesting of authors. Through the relationship of authors, users can get the research topic of a certain area to a certain extent. Some co-authorship networks have been researched[19][1][3], and this project researches on the co-authorship network in China digital library CNKI, project uses the CPM to find the communities in co-authorship network, and algorithm based on core node detection algorithm to track the evolving communities.

## 5.2 special works

### 5.2.1 Experiment tools

CFinder, JDK 1.6, Netbeans and Excel.

CFinder is the analysis software for analysing the static communities based on the clique percolation method. Furthermore, it provides three methods of community finding:CPM, CPMd and CPMw, where CPMd and CPMw is the extension of CPM. CFinder can visualize the results of community detection, and provides the function

of community analysis, such as, size distribution, degree distribution, membership distribution and overlap distribution.

### 5.2.2 Experiments steps

(1) According to the collaboration of authors, create the network in every time interval, the time interval is one year in this project, and analyse the topological properties of obtained networks.

(2) Using CPM to extract communities in every network

(3) Finding predecessor (successor) of communities in the networks, and analysing the community evolution properties while obtaining community evolution path.

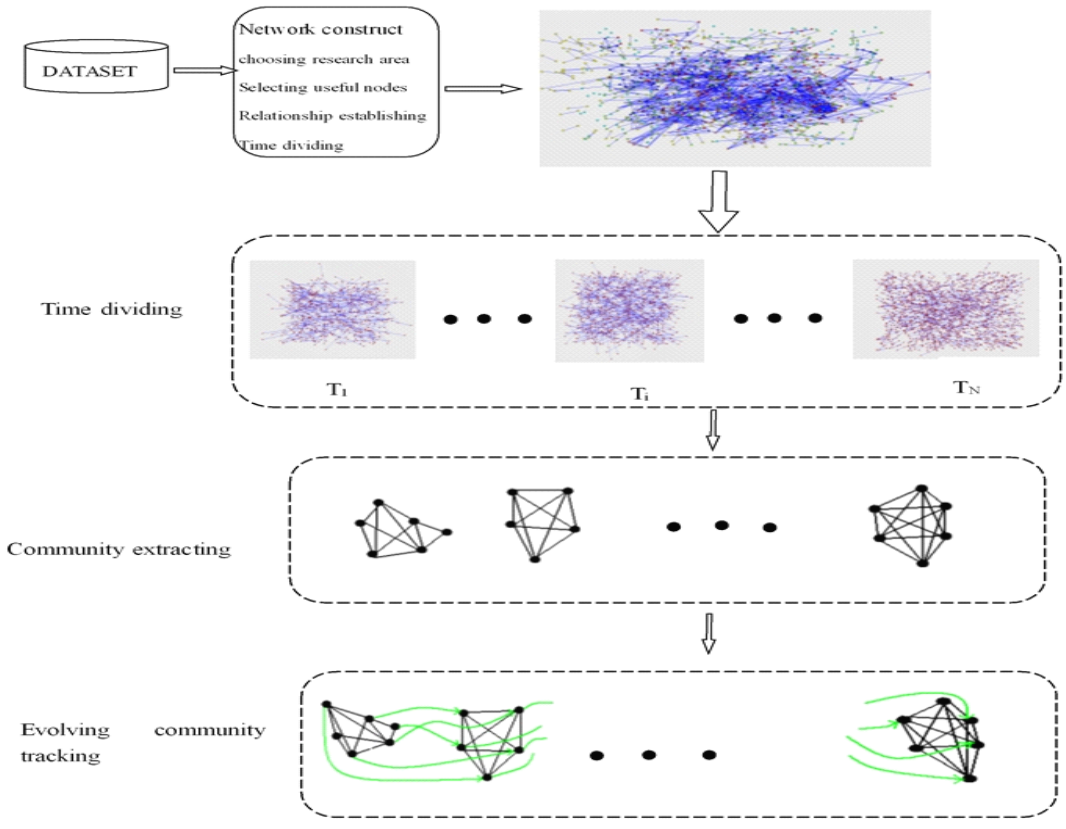



Figure 5-1 Experiment frame

### 5.2.3 Data set

The data in this project is collected from China digital library CNKI(Chinese National Knowledge Infrastructure ), which contains more than 7000 types of journals and magazines since 1979, it is the most popular and widely used digital library. CNKI classifies papers by research areas and one can search paper by just inputting author name, title, date and so on. CNKI has an excellent reputation in the academia in China, cause it provides great convenience to researchers, especially for Chinese researches. In the project, author just chooses the data related to the financial area.

The following figure display the numbers of documents that collected by CNKI from 2005 to 2009, it is easy to find that the number rises in the research time(2005-2010).

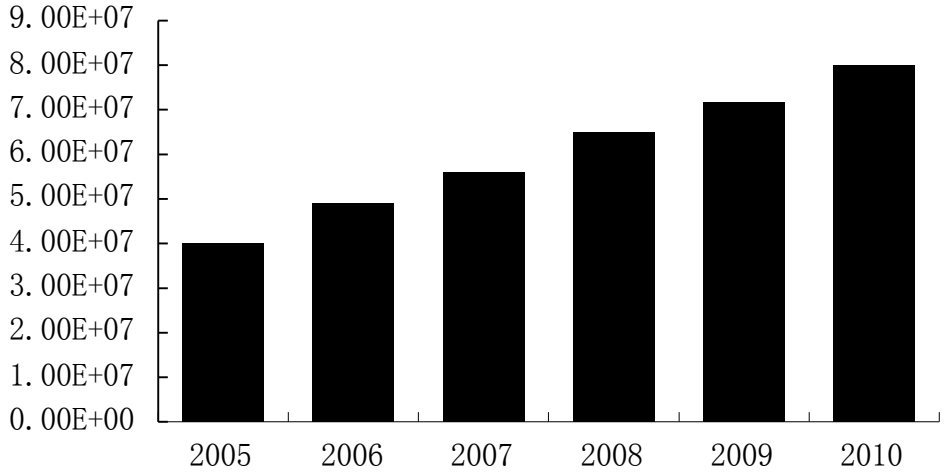


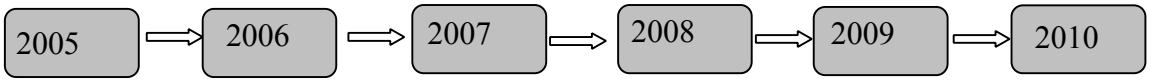

Figure 5-2 Number of documents

## 5.3 Network construct

Co-authorship network could display the research hot spot of a research area in a certain extent. This project collects the author data from the CNKI, and construct the 2005,2006,...,2010 co-authorship networks separately, the obtained networks are unweighted undirected network. Every author in the network has a corresponding

node, and the edges in the network represent the collaboration relationships;

### 5.3.1 Time division




This project chooses one year as time interval to construct the co-authorship network in CNKI, one of the reason is that if the time interval is too small, the community extraction will be interrupted, and Zhao et al. point that it is enough for the literature network to choose one year as the time interval[28]. Another reason is that, lots of conferences are held yearly, thus it is reasonable for the literature network to choose one year as time interval.

### 5.3.2 Obtained Networks

The following figure is one of the co-authorship networks that was constructed by the collected data

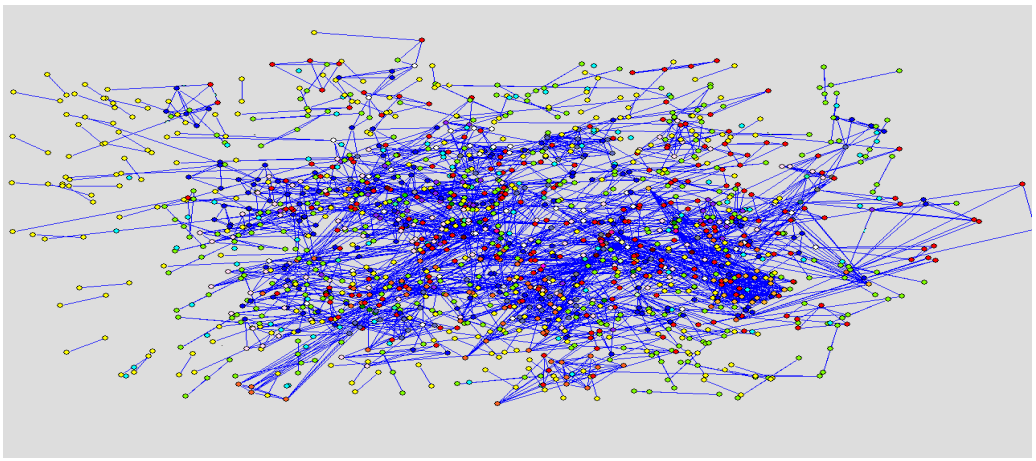
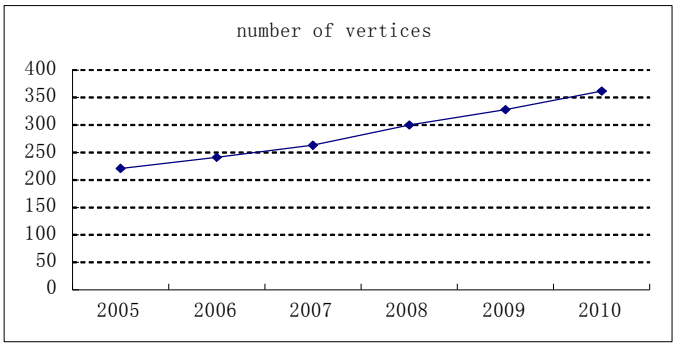
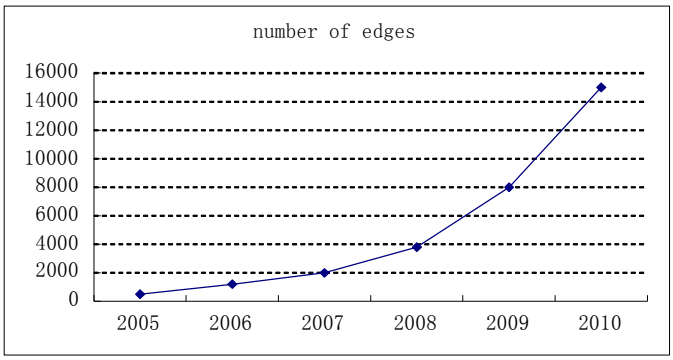


Figure 5-3 co-authorship network in 2010

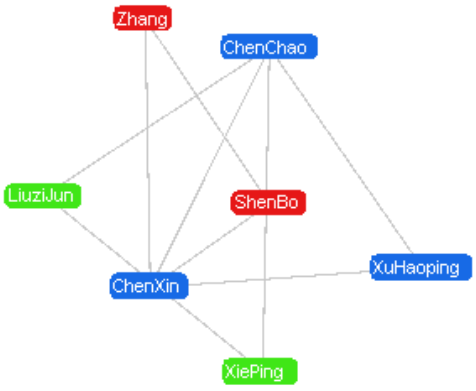
The project analyses the obtained networks, and the results show that size of network grows every year, and the phenomenon can be easily found from the following pictures.



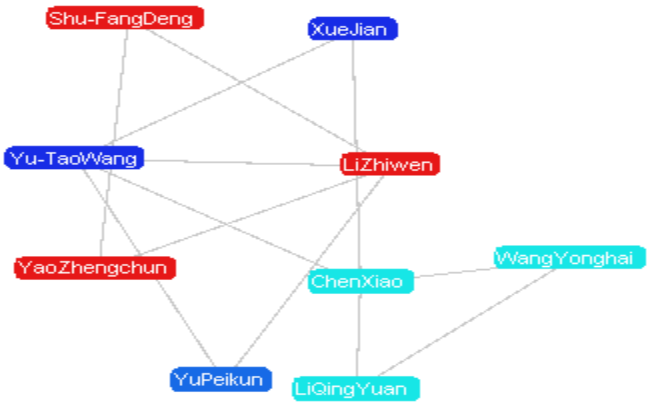




## 5.4 Community extraction

In this project, author uses CPM to extract communities; the following figures show parts of extraction results.




(a)




(b)


(c)

Figure 5-4 communities

## 5.5 Community evolution

According to the evolution tracking experiment, author find there are some properties of community in co-authorship network.

- If a community has longer lifespan, then the rate of this community will be higher.
- Bigger communities have more nodes.
- In the community evolution path, if the community has higher correlation rate with its successor(predecessor), then there is a lower probability for that community to be a merging point or split point, otherwise, if it has a lower correlation rate, it will be an merging or split point to a great extent. The project defines the correlation rate as following: $CR(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$ where $C_i$ and Cj are the adjacent communities.
- If an community contains a large number of nodes, the community has a higher probability to be a merging or split point, otherwise, if there are a few nodes in the community, then it will more stable than the bigger communities.

However, most of the above properties have mentioned by Palla and Barabasi in the [19] , that means the algorithms the project chooses is suitable for the constructed co-authorship network.

### 5.5.1 Big communities

(1) Nodes

According to the composition of community, nodes in the big community are the authors who are more authoritative in the financial area in China (cause the project data is collected from CNKI in the financial area), and most of nodes in it have a bigger degree.

(2) Evolution

As time goes by, the nodes of the community are changing, new nodes are added and old nodes are disappeared at any time interval, nevertheless, some nodes in it are stable, especially core nodes. In the evolution process, the correlation rate of big community fluctuates in the range between 0.6 and 0.9. The phenomenon not only shows the changes of the community, but also display that most nodes in it are stable. In addition, for a big community, it is easier to find its successors or predecessors in most of time. In other words, big communities cannot appear or disappear suddenly in general.

Although there are not emerging phenomenon for a big community, we can also learn the changes of whole network by analysing the big community evolution process.

| Disappeared nodes | |
|---|---|
| 2006 | Shenmingzhi, jinxin, wangweiqiang, wugang |
| 2007 | Huangqingming, hangjie, yingjianyao, weijiang, wuxiaoru |
| 2008 | Sunmingbo, sunmingbo, gengxuesongli |
| 2009 | Yuananfu, yibo, yaozhijian shaoyihua, yingyong |
| 2010 | Wulinha, xiehuongyuan, shenminyao, wanganquan, shonghongbing |
| Appeared nodes | |
| 2006 | Shenxiaoxia, xieyoudong,lizhipeng |
| 2007 | Fanbaoqun, gengxuesongli, c.smith, fuyuxiu |
| 2008 | Shaoyiha, dengfeng, chenlinfen, chengjun, yanghuachu |
| 2009 | Liuxiangwu,caixiaoshen,chenyue,zhangjianhong |
| 2010 | Wangshengxi, moweiwui, liyan, sunyuanyuan, gaomin, dengjiechao |
| Community core | |
| 2005 | {xuqingrui, caining, xuxiaodong, huangqingming, guobin, chenjin ,weijiang} |
| 2006 | {xuqingrui, chenjin,gengxuesongli} |
| 2007 | {xuqingrui, xuxiaojun, maoyihua, chenjin, guobin} |
| 2008 | {xuqingrui, guobin, chenjin, wangyi} |
| 2009 | {xuqingrui,chenjin,maqingguo,liujingjiang} |
| 2010 | {xuqingrui, xuxiaojun,guobin,liuzeyuan,chenjin,maqingguo } |

Figure 5-5 An example of the changes of nodes in a big community

### 5.2.2 Small community

(1) Nodes

The size of small community is far smaller than big community, according to the composition, most of small communities have more clear meaning. A small community shows the specific subject of authors in the community.

(2) Evolution

Different from big communities, most of small communities are changed severely, that means that they could be fully stable, and also could emerge and disappear suddenly, therefore, small communities has shorter lifespan than bigger ones.

However, there are some small communities that does not changed severely, the following example is a small community evolution process, and we can see from the figure, the size and core set of it grow steadily.

| year | nodes | core set |
|---|---|---|
| 2005 | lili,qiyingfeng linyifu | {lili,qiyingfeng ,linyifu} |
| 2006 | lili,qiyingfeng,linyifu,zhangtao | {lili,qiyingfeng ,linyifu,zhangtao} |
| 2007 | lili,qiyingfeng,linyifu,zhangtao | {lili,qiyingfeng ,linyifu,zhangtao} |
| 2008 | lili,qiyingfeng,linyifu,zhangtao | {lili,qiyingfeng ,linyifu,zhangtao} |
| 2009 | lili,qiyingfeng,linyifu,zhangtao , guobin | {lili,qiyingfeng ,linyifu,zhangtao,guobin} |
| 2010 | lili,qiyingfeng,linyifu,zhangtao , guobin | {lili,qiyingfeng ,linyifu,zhangtao,guobin} |

Figure 5- a small community evolution process

# Chapter 6: Summary and future development

The project carries out n research on evolving community. The aim of it is to find the property algorithms to investigate how communities in the complex network evolve over time, such as shrinking, growth, disappearance, birth, etc., which is achieved by the end of this project.

Firstly, the author focus on the study of community detection algorithms, because lots of communities in the co-authorship network are overlapping, author chooses CPM to extract the community, and the results of the experiment in this project shows that CPM is a suitable community detection method for the co-authorship network.

Secondly, most of algorithms use the fixed parameters standard to track the evolving communities, the shortage is that algorithms cannot fully focus on the dynamic change of networks, and choose a suitable parameter is very hard for the dynamic network. Therefore, this project use a core-base algorithm to track the communities evolution, and final finds that the core-base algorithm can track the evolving community very well.

Lastly, the project experiment with the above chosen algorithms to constructed co-authorship network. Extracting the communities from network, and analysing the evolution of big and small communities at the same time.

Even though the project has achieved the above objectives, there are also some future developments.

According to the experience and time limited, the size of chosen co-authorship network is not big enough, and the data just span 5 years, in the future research the data set should be spanning more than 5 years, cause the more data collected, the higher accuracy of the result, furthermore, whether the algorithm is suitable for bigger network needed to be researched.

In this project,author defines the measurement standards of community evolution, however, in the experiment part, this project just use the standards in the social

network, the accuracy of these standards in the unsocial network should be researched.

when analyse the communities in the different time interval，the project uses the same condition to extract communities from the network. However, in the community evolution, the communities` properties are also changes over time; hence it is hardly to get the community evolution path by analysing the communities that are extracted using the fixed parameter efficiently.

There are also some functions should be completed in this project, such as how to constructed the network accurately and how to track the evolving communities in bipartite network.

# References

[1] Asur, S., Parthasarathy, S. and Ucar, D. (2007) An Event-based Framework for Characterizing the Evolutionary Behavior of Interaction Graphs, *TKDD,* 3(4).

[2] Barabasi, A-L., Albert, R. and Jeong, H. (2000) Scale-free characteristics of random networks: the topology of the world-wide web.*Physica A:Statistical Mechanices and its Applications,* 281(1-4), pp.69-77.

[3] Baumes,J. Chen H.,Francisco M.,Goldberg，M., Magdon-Ismail M. and Wallace W. (2008) ViSAGE: A virtual laboratory for simulation and analysis of social group evolution,*TAAS,* 3(3).

[4] Berger-Wolf, T.Y. and Saia, J. (2006) A framework for analysis of dynamic social networks, In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining ,Philadelphia,2006.* New York: ACM, pp. 523-528.

[5] Braun,T. (2004) Hungarian priority in network theory, *Science*,304(576), pp.1745.

[6] Capocci, A., Servedio, V.D.P., Caldarelli, G. and Colaiori, F. (2004) Detecting communities in large networks, *Physica A: Statistical Mechanics and its Applications,* 352(2-4), pp. 669-676.

[7] Ding, L.H., Shi, P. (2008) *Network Community Detection.* 1st ed. Chemical Industry Press.

[8] Ebel, H. Mielsch, L-I. and Bornhold, S. (2002) Scale-free topology of e-mail network. *Physical review E,* 66(3), pp. 035103-1-035103-4.

[9] Elmacioglu, E.and Lee, D.(2005) On Six Degrees of Separation in DLP-DB and More,*SIGMOD Record,* 34(2),pp. 33-40.

[10] Grossman, J. and Ion,P. (2006) *The Erdos Number project* [WWW] Oakland University. Available from: http://www.oakland.edu/enp [Accessed 09/05/2010].

[11] Kumar, R., Novak,J., Raghavan,P. and Tomkins, A.(2003) On the Bursty

Evolution of Blogspace,*World Wide Web*,8(2), pp.159-178.

[12] Li,X. and Chen,G.(2003). A local-world evolving network mode, *Physica A:Statistical Mechanics and its Application,* 328(1-2), pp. 274-268.

[13] Lin,Y., Chi, Y. Zhu, S. Sundaram,H. and Belle L. Tseng. (2008) FacetNet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks. *WWW 2008*, pp.685-694.

[14] Mane, K.K. and Börner, K., (2004) Mapping topics and topic bursts in PNAS, *PNAS*, 101,pp. 5287-5290.

[15] Maru,J.T., Borner,K., Goldstone,R.L. (2004) The simultaneous evolution ofauthor and paper networks, *PNAS ,* 101(Suppl 1), pp. 5266-5273.

[16] Milgram, S. (1967) The small world problem, *Psychology Today 2,* pp. 60-70.

[17] Palla, G., Derenyi, I., Farkas, I. and Vicsek, T. (2005a) Uncovering the overlapping community structure of complex networks in nature and society, *Nature,* 435, pp. 814-818

[18] Palla, G. and Derenyi, I. (2005b) Clique percolation in random networks, *Physical Review Letters,* 94, pp. 160-202.

[19] Palla, G., Barabasi, A-L. and Vicsek, T. (2007a) Quantifying social group evolution, *Nsyutre,* 446, pp. 664-667.

[20] Palla, G., Barabasi, A-L. and Vicsek, T. (2007b) S*upplementary Information (in Quantifying social group evolution)*[PDF]. Available from: http://www.nature.com/nature/journal/v446/n7136/suppinfo/nature05670.html .[Accessed on 22th Feb., 2010]

[21] Reynolds, P.. (1999) *The Oracle of Bacon* [WWW]. Available from: http://oracleofbacon.org/ [Accessed 23/04/2010].

[22] Peng, M.J.(2009) *Complex network study,* Yang Zhou University.

[23] Pothen, A., Simon, H. and Liou, K.P. (1990) Partitioning Sparse Matrices with Eigenvectors of Graphs, *SIAM. J. Matrix Anal. & Appl.,* 11(3), pp. 430-452.

[24] Toyoda, M. and Kitsuregawa,M. (2003) Extracting evolution of web communities from a series of web archives,In: *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, Nottingham,August 2003*. New York: ACM, pp.26-30.

[25] Wang, X.F., Li, X. and Chen, G.R. (2006) *Theory and appliance of complex network*. Beijing: Tsinghua University Press.

[26] White, H.D., Lin, X., Buzydlowski, J.W., Chen, C. (2004) User-controlled mapping of significant literatures, *PNAS*, 101, pp. 5297-5302.

[27] Wu, F. and Huberman, B.A. (2004) Finding communities in linear time a physics approach, *Eur. Phys. J. B,* 38(2), pp. 331-338.

[28] Zhao, Q., Bhownick, S.S. and Zheng, X. (2008) Characterizing and predicting community members from evolutionary and heterogeneous networks, In:*CIKM08*, *Califonia, October 2008*. New York: ACM, pp. 309-318.

[29] Kernighan, B.N. and Lin, S. (1970) An efficient heuristic procedure for partitioning graphs*, The Bell system technical journal*, 49(1), pp. 291-307.

[30] Wang, Y., Wu, B. And Yang, S.Q. (2008) CommTracker: A Core-Based Algorithm of Tracking Community Evolution,*Lecture Notes in Computer Science*, 5139, pp.229-240.

# Appendix: Source code

### Core_node_algorithm

```
/*
 * File:   core_node_algorithm.c
 * Author: Minghui Jiang
 * Created on August 1, 2010, 4:52 AM
 */

#include <stdio.h>
#include <stdlib.h>

/* The code is use to calculate the
 *degree of nodes that in the given
 *communities
 */
int main(int argc, char** argv) {
    FILE *fp;
    char v[50][20]={'\0'},ch='\0';
    int i,j,h,e,r;
    int cen[50]={0},tmp=0;
    int flg=0,d[50]={0};
    if(argc<2){
        printf("Please enter the name of the community.\n");
        return;
    }
    if((fp=fopen(argv[1],"r+"))==NULL){/*read the community file*/
        printf("Cannot open the file\n");
        return;
    }
    for (i=0;i<50;i++){/*community nodes searching*/
        if(feof(fp)!=0)break;
        else{
            for(j=0;j<20;j++)
            {
                ch=fgetc(fp);

    if(ch!='\t'&&ch!='\n'&&feof(fp)==0)
                    v[i][j]=ch;
                /*put each appeared node of community in the array v*/
```

```
                else break;

            }

            for(h=0;h<i;h++)
                if(strcmp(v[h],v[i])==0)
{/*node have already appeared in the array v*/
                    d[h]++;break;
                    /*puls one to the degree of given node*/
                }
            if(h==i)d[i]++;/*node is the first time to appear in the community node searching*/
            else i--;

            flg=i;/*the value of flg represent the number of nodes in the given community*/
        }
    }

    rewind(fp);/*reread the community file*/
    for(i=0;i<50;i++)
        for(j=0;j<20;j++)
            v[i][j]='\0';

    for (i=0;i<50;i++){/*calculate the core degree of each node*/
        if(feof(fp)!=0)break;
        else{
            for(j=0;j<20;j++)
            {
                ch=fgetc(fp);
    if(ch!='\t'&&ch!='\n'&&feof(fp)==0
)
                    v[i][j]=ch;
    if(ch=='\t'){e=0;break;}
    if(ch=='\n'||feof(fp)!=0){e=1;break;}
            }

            for(h=0;h<i;h++)
                if(strcmp(v[h],v[i])==0)
break;
            if(h<i) i--;
            if(e==0)tmp=h;
```

```
            else{
                if(d[h]<d[tmp])
                {

cen[h]=cen[h]-(d[tmp]-d[h]);

cen[tmp]=cen[tmp]+(d[tmp]-d[h]);
                }
                else
                {

cen[h]=cen[h]+(d[h]-d[tmp]);

cen[tmp]=cen[tmp]-(d[h]-d[tmp]);
                }
            }
        }
    }
    fclose(fp);
     FILE *in;

if((in=fopen(argv[1],"a"))==NULL){/*
write the core node set to the community
file*/
        printf("Cannot open the file\n");
            return;
        }
        fputc('\n',in);
        fputc('{',in);
        for(i=0;i<flg+1;i++){
            printf("nodes:%s has core
degree %d\n",v[i],cen[i]);
            /*printf core degree for every
node on the screen*/
        }
        printf("core node is :");
        for(i=0;i<flg+1;i++){
            if(cen[i]>=0){
                for(j=0;j<20;j++)

if( v[i][j]!='\0' )fputc(v[i][j],in);
                    else break;
                fputc('\t',in);
                printf("%s",v[i]);/*print
core nodes on the screen*/
            }

        }
        printf("\n")
```

```
    fclose(in);
    in=fopen(argv[1],"rb+");
    fseek(in,-1L,SEEK_END);
        fputc('}',in);
                                        fclose(fp);
                                        return (EXIT_SUCCESS);
                                    }
```

**Core_based_community_tracking_algorithm**

```
/*
 * File:      track_core_node.c
 * Author: J
 *
 * Created on august 3, 2010, 5:51 AM
 */

#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#include <math.h>
/*
* This algorithm is based on the above
*algorithm and use to track the
* community evolution path.
 */
int main(int argc, char** argv) {
    FILE *fp_given,*fp_com;
    FILE *give_community_evolution;
    int
i=0,j=0,h=0,flg_nodes=0,flg_core=0,n=
0,m=0,p=0,q=0;
    int          flg_nodes_com=0,
flg_core_com=0;
    int
compare_para1=0,compare_para2=0;
    char ch='\0';
    char
core_given[500][20]={"\0"},v_given[50
0][20]={"\0"};
    char
core_com[500][20]={"\0"},v_com[500][
```

```
20]={"\0"};


    if(argc<4)
    {
        printf("Please give the name of community in time t and "
               "the numbers of community in time t+1 "
               "(please no more than 10 once)and "
               "the names of communities in the time t\n");
        return;
    }

if((fp_given=fopen(argv[1],"rb+"))==NULL){/*read the given_community file*/
        printf("Cannot open the file\n");
        return;
    }

give_community_evolution=fopen("evolution_path","ab+");
    for (i=0;i<500;i++){/*community nodes searching*/
        if(feof(fp_given)!=0)break;
        else{
            for(j=0;j<20;j++)
            {
                ch=fgetc(fp_given);

    if(ch!='\t'&&ch!='\n'&&!feof(fp_given)&&ch!='{')
                    v_given[i][j]=ch;
                /*put each appeared node of community in the array v_given*/
                else break;

            }
            if(ch!='{'){
                for(h=0;h<i;h++)

    if(strcmp(v_given[h],v_given[i])==0) /*node have already appeared in the array v_given*/
                    {break;i--;}
            }
            else {
```

```
                ch=fgetc(fp_given);
                for(n=0;n<500;n++){


    if(ch!='}'&&!feof(fp_given)){

    for(m=0;m<20;m++){
                                if
(ch!='}'&&ch!='\t'&&!feof(fp_given))

    {core_given[n][m]=ch;ch=fgetc(fp_
given);}
                                else
break;

                            }

    ch=fgetc(fp_given);


                        }
                        else break;
                }
                flg_core=n;/*the value
of flg_core represent the number of core
in the given community*/
                ch=fgetc(fp_given);
                }
                flg_nodes=i;/*the value of
flg_nodes represent the number of nodes
in the given community*/
            }
        }

for(p=0;p<((int)argv[2][0]-48);p++){

    if((fp_com=fopen(argv[p+3],"rb+"))
==NULL){/*read the given_community
file*/
            printf("Cannot open the
file\n");
            return;
        };
        for
(i=0;i<500;i++){/*community nodes
searching*/
            if(feof(fp_com)!=0)break;
            else{
                for(j=0;j<20;j++)
                {
                    ch=fgetc(fp_com);

    if(ch!='\t'&&ch!='\n'&&!feof(fp_co
```

```
m)&&ch!='{')
    v_com[i][j]=ch;
                /*put each appeared node of community in the array v_com*/
                else break;
            }
            if(ch!='{'){
                for(h=0;h<i;h++)
    if(strcmp(v_com[h],v_com[i])==0)
/*node have already appeared in the array v_com*/
                    {break;i--;}
            }
            else {
                ch=fgetc(fp_com);
    for(n=0;n<500;n++){
    if(ch!='}'&&!feof(fp_com)){
        for(m=0;m<20;m++){
                    if (ch!='}'&&ch!='\t'&&!feof(fp_com))
    {core_com[n][m]=ch;ch=fgetc(fp_com);}
                    else break;
                }
    ch=fgetc(fp_com);
            }
            else break;
        }
    flg_core_com=n;/*the value of flg_core represent the number of core in the given community*/
                ch=fgetc(fp_com);
;
            }
            flg_nodes_com=i;/*the value of flg_nodes represent the number
```

```
of nodes in the given community*/
                }
            }
            for(i=0;i<flg_core;i++)
                        /*there must be at least one core node of C(t) in C(t+1)  : */
                if (strcmp(core_given[i],v_com[i])==0){
                    compare_para1=1;
                    break;

                }
            for(i=0;i<flg_core_com;i++)
                        /*there must be at least one core node of C(t+!) in C(t) */

      if(strcmp(core_com[i],v_given[i])==0){
                        compare_para2=1;
                        break;

                }

      if(compare_para1&&compare_para2)

      fprintf(give_community_evolution,"%s\t",argv[p+3]);
                        /*print community evolution path in a binary file*/

          fclose(fp_com);
      }

      return (EXIT_SUCCESS);
}
```