Opinion dynamics on typical complex networks and applications

Wenting Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Mathematics

University College London

April 2, 2017

I, Wenting Wang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis investigates the dynamics in models of how opinions within a network of people, or of entities, change over time before arriving at a consensus. Considering the system as a complex network, continuous models are derived based on differential equations with each node in the network representing a person or entity. The interactions between the entities are explored and the influence of the topology of the network is established. It is shown that the structure and evolving mechanisms are crucial factors to determine whether there will be a stable consensus and to establish the network efficiency at which the system approaches a consensus. Both linear and nonlinear dynamics are considered. A new algorithm of a network partition is developed based on the fact that some nodes achieve local consensus earlier than the global stable solution. The experimental results show that the algorithm outperforms existing methods. Special consideration is given to networks which undergo an explosive phase transition, when a small number of new connections cause a rapid change in network dynamics with consensus occurring after the transition point. Results indicate that the consideration of spatial variations incorporating a social outcast strongly influence the dynamics

approaching consensus. The methods are applied to illustrate the two party election competition, which demonstrates characteristic behaviour prior to majority.

Papers

- W. Wang, and R. Bowles, Opinion Dynamic in Evolving Networks with Suppression Effect of Explosive Transition. Academic Conference of Complexity Science in the Real World Network, EPSRC, (2012).
- Y. Zhang, M. A. Aziz-Alaoui, C. Bertelle, S. Zhou, and W. Wang, Fence-sitters protect cooperation in complex networks. Phys. Rev. E. 88, 032127, (2013).
- W. Wang and R. Bowles, Opinion dynamics on complex networks with a social outcast, Netsci, (2015).

Acknowledgements

It is March 19th,2017 today. Finally I have gained enough courage to put an end to the correction of my thesis. Never can I say this is an accomplished work nor this area is fully-developed with the tools from mathematics, physics, etc. It may be a life-long job and the pursuit of mine to continue with the research in both methodology and applications of complex networks. However, there is a deadline for my correction of the thesis. Since the oral defense on May 19th, 2015, I have made full use of the time to improve every Chapter of the thesis by the suggestions of my supervisors and examiners.

It is greatly appreciated that I have the best supervisor in the world. Thank you, Robert(Dr Robert Bowles). When I first came to the Department of Mathematics, I was just a graduate student from other area with plain English skill. Robert showed great patience and provided me with any mathematical skills I needed. During all the past 6 years, I have suffered from several unhappy personal issues. Robert helped me to balance the attention between work and life. In those hard times, it was always my dear supervisor to support my research in a very positive way.

Great appreciation to Prof Frank Smiths too, who is always the idol of our PhD students. Never you can imagine a mathematician with reputation that high is so nice to the young researchers. He is always there, suggesting and tutoring in a humorous and smart way. He leads the trends of the research and trains the youths with love.

There was always some 'loud' discussions about the nonlinear dynamical systems between my second supervisor Prof Steven Bishop and me. It will be a precious memory that Steven, as the authority of the dynamical systems, gave me the chance to speak out loudly my dreams of science and helped me in the way I understood.

It was definitely a fortune that I have been tutored for a while by Prof Anthony Finkelstein from Department of Computer Science. It was the very moment when I have difficulty to finish my upgrade report. Then Anthony helped me with both methodology and data, which became the Chapter 6 of my thesis later.

I have met some true friends of life in KLB, thank you my guys, to make companions in the past 6 years. Thank you, Hannah, Peter, Toby, Stephen, Nana, Ben, and everyone. You guys always know that I can never remember your family names because we never use them in our office.

I appreciate greatly my two examiners, Dr Sergei Timoshin and Dr Mark Blyth, not only because you have gave me very important suggestions and comments to help me to improve my research, but also, you are the only two people who really know what I have done in this area besides my supervisors.

Finally, I would like to dedicate this thesis to my mother in heaven. Daddy and I live with the pain of losing you. My mother was diagnosed with gastric cancer in 2013 and

passed away in April, 2015. Maybe it was the darkest time of my life to take care of my mother and write up my thesis at the same time. My mother was always a fighter. She was a surgeon and had a very successful career life. When I was a kid, she always told me that, to be a good surgeon, you should have the courage to cut and sew up yourself. I take this sentence as the faith of my scientific life. I will face the mistakes and insufficient in my research and make it better. In the rest days of my life, I will make the best of myself because this is the life you gave. Mom, I love you.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The complex science always develops itself by learning from classical physics and mathematics when cooperating with them in applications. The theory of complex network is widely used in the study of social networks. We define the synchronization process of opinions on a social network as 'opinion dynamic'. As one of these applications, the opinion dynamics bring complex networks, matrix theory, control theory together. This thesis studies the methodology to describe, analyze, predict and control the networked dynamical systems towards synchronization.

1.2 Problem Definition

Opinion dynamics, as one of the social dynamics studied extensively in recent years, is the dynamics of systems incorporating the evolution of two or more competing states [Saber et al., 2007] through various mathematical and statistical physics theories. A

lot of previous studies use the words 'opinion dynamics' to describe different systems where people exchange their opinions [Sznajd-Weron and Sznajd, 2000, Stauffer, 2001, Lambiotte et al., 2009, Lorenz, 2005]. We will review some of them in Chapter 2. This study is based on the continuous opinion model of John P. Curtis and Frank T. Smith [Curtis and Smith, 2008]. Assume two persons hold initial opinions X_1 and X_2 respectively and they persuade each other with the powers μ_{12} and μ_{21}. The opinions will evolve in accordance with the equations

$$\dot{X}_1 = \mu_{21}(X_2 - X_1), \tag{1.1}$$

$$\dot{X}_2 = \mu_{12}(X_1 - X_2), \tag{1.2}$$

where the $dotX_1$ and $dotX_2$ indicate the differentiation with respect to time. If we define the difference of the opinions as

$$u = X_2 - X_1, \tag{1.3}$$

then, subtracting Equation (1.1) from Equation (1.2) and using Equation (1.3) yield

$$\dot{u} = -(\mu_{12} + \mu_{21})u, \tag{1.4}$$

which leads to the solution

$$u = Ae^{-(\mu_{12}+\mu_{21})t}. \tag{1.5}$$

We redefine the parameters $X_1(0)$ and $X_2(0)$ as follows, which will be used in Chapter 3

$$X_1(0) = X_{10}, \tag{1.6}$$

and

$$X_2(0) = X_{20}. \tag{1.7}$$

In this case,

$$u = X_{20} - X_{10} \tag{1.8}$$

The solution of Equation (1.1) becomes

$$\dot{X_1} = \mu_{21}(X_{20} - X_{10})e^{-(\mu_{12}+\mu_{21})t}, \tag{1.9}$$

The integration of X_1 over time t up to the current time yields

$$X_1 = -\frac{\mu_{21}(X_{20} - X_{10})}{\mu_{12} + \mu_{21}} e^{-(\mu_{12}+\mu_{21})t} + C_1, \tag{1.10}$$

where C_1 is a constant. By some technical manipulation of Equation (1.10), the solution for X_1 becomes

$$X_1 = X_{10} + \frac{\mu_{21}(X_{20} - X_{10})}{\mu_{12} + \mu_{21}}(1 - e^{-(\mu_{12}+\mu_{21})t}). \tag{1.11}$$

Similarly, we find the solution of X_2 as

$$X_2 = X_{20} + \frac{\mu_{12}(X_{10} - X_{20})}{\mu_{12} + \mu_{21}}(1 - e^{-(\mu_{12}+\mu_{21})t}). \tag{1.12}$$

In this model, the opinions of two persons converge to the limit of $\frac{\mu_{12}X_{20}+\mu_{21}X_{10}}{\mu_{12}+\mu_{21}}$. A three persons model was also examined in some detail in [Curtis and Smith, 2008], which inspired the N-persons models of this study. To describe and model a large-scale social phenomena mathematically, we consider an additional factor which indicates whether the pairwise persuasion between persons i and j happens or not. We will introduce a connection to each pair of persons i and j in the study, and consider the system as a complex network. The corresponding adjacency matrix of the network is defined as A. The entry of $A = a_{ij}$ is 1 if i talks to j, and 0 otherwise. We use b_i instead of μ_{ij} to represent person i's ability to persuade j for $1 \leq j \preceq N$ and $j \neq i$. Then, the N-persons model is described as the following equations

$$\begin{cases} \dot{X_1} &= b_2a_{12}(X_2 - X_1) + b_3a_{13}(X_3 - X_1) + \cdots + b_Na_{1N}(X_N - X_1) \\ \dot{X_2} &= b_1a_{21}(X_1 - X_2) + b_3a_{23}(X_3 - X_2) + \cdots + b_Na_{2N}(X_N - X_2) \\ \vdots &= \vdots \\ \dot{X_N} &= b_1a_{N1}(X_1 - X_N) + b_2a_{N2}(X_2 - X_N) + \cdots + b_{N-1}a_{N-1N}(X_{N-1} - X_N) \end{cases} \tag{1.13}$$

In this model, we maintain the dynamics that people don't impact themselves from [Curtis and Smith, 2008]. Obviously, if all $b_i > 0, i = 1, 2, \cdots N$, we can always obtain a solution like $X_1 = X_2 = \cdots = X_N = s(t)$ when $t \to \infty$. The explanation is given in

Equation (2.18)-(2.20). The consensus $s(t)$ is a constant when all $a_{ij} = 1$, and b_i and initial opinions X_i are fixed. When setting some $a_{ij} = 0$ and putting the network topology back into sight, the model becomes closer to social reality. It is always difficult to define the term of 'topology' of a network. Hence,there are lacks of tools when analyzing the relations between the network topology and the dynamics on networks. The control of networks have wide uses in many areas. In this thesis, we intend to make improvements in the analytical tools in complex networks. The numerical researches are followed to test the effectiveness of the methodologies.

1.3 Research Objectives

A lot of research works on complex networks aim at applying mathematical and physical methods to solve problems of different disciplines, such as economics, politics, biology and ecology, etc. The synchronization process is ubiquitous in nature and play a very important role in many different contexts. The previous studies have been focused on how to achieve a synchronization in systems and how to predict and control the chaotic response if the network system is chaotic and consequently sensitive to even a small perturbation [Olfati-Saber, 2005, Xiao and Boyd, 2004, Watts and Strogatz, 1998]. However, previous studies have achieved solutions of opinion models on undirected networks. The directed network models are rarely studied. This is partly due to the fact that analysis tools such as graph theory are not well developed in directed networks, especially directed-weighted networks. For example, there are no standard definitions for the algebraic connectivity, which is an important measurement for the

network synchronization and its speed, while its counterpart for undirected graphs has been extensively used in studying the synchronization problems.

In this research, all the models are defined as Equation (1.13). Some of the models include a "social outcast" which has strong and negative influence on others connecting to it. We study the directed-weighted network and develop new methods as well.

The specific objectives of this research are as follows:

1. The speed of synchronization impacted by network topology, which is investigated in Chapters 2 and 3;

2. The stability of different networks against perturbation, which is investigated in Chapter 4;

3. The process of synchronization in different networks, which is investigated in Chapters 5 and 6;

4. The emergence of synchronization in systems enhanced by the nonlinearity during the evolution towards synchronization, which is investigated in Chapter 7.

1.4 Research Methodologies

This study will use multiple methodologies inseparably:

- The complex networks are used to describe the opinion system, especially the topology;

- The graph theory and matrix theory are used to analyze the eigenvalue spectrum of the Laplacian matrices in charge of system evolution;

- The master stability method is used to determine whether there will be a synchronization in the system and the speed of synchronization;
- Analytical methods, for instance, the solutions of ordinary differential equations and the asymptotic methods;
- Some methods in control theory to analyze the nonlinear models;
- Programming platforms such as Matlab, Mathematica, Origin and Pajek to simulate and visualize the system evolution.

It is worth mentioning that during the study, some of the methodologies are extended. For instance, the applicability of the Laplacian spectrum has been extended from undirected networks to directed-weighted networks.

1.5 Contributions

1. In this thesis we build a N-persons model on complex networks. Both linear and nonlinear dynamics are studied on the models. We investigated how the network topology manipulates the synchronization of opinions on different networks. The analytical and numerical results are presented.
2. We apply the synchronization process to network partitions and develop a new partition algorithm.
3. We introduce degree correlation as a new statistical physical characteristic to describe the network topology, and investigate the relations between the charac-

teristics of topology and the synchronization speed.

4. We develop a systematic methodology to apply mathematical methods on social problems by complex networks.

1.6 Thesis Structure

Chapter 1 introduces the motivation of this thesis. The problem is defined and the objectives of the research are laid out, as well as main contributions. In Chapter 2, we establish the networked opinion models. The spectral analysis is used to measure the speed of synchronization in several kinds of topologies. Meanwhile, the master stability functions are tested. Finally, the results from the spectrum analysis, master stability functions and computer simulations are compared. In Chapter 3, the social outcast is introduced in opinion models. The stability against a social outcast in different topologies is tested and analyzed. In Chapter 4, the nonlinear model is established. We study the control of the whole network by manipulating some key nodes which are addressed as 'attractors'. Chapter 5 discusses the process of synchronization. A real project named *RALIC* in UCL is presented. In Chapter 6, two algorithms are given as the extension of the *ODM* matrix from Chapter 5. One is the balanced Min-cut to partition the graph. The other is the feature selection algorithm based on the *ODM* matrix which is more effective than the similar procedure based on the adjacency matrix. In Chapter 7, the exchanges of the majority of the opinions

are investigated. In Chapter 8, the emergence of consensus is observed in a kind of growth network generated by Achlioptas Process. The ability of these kinds of networks against the perturbation from an outcast is investigated. Chapter 9 concludes this work, discusses potential limitations, and puts forward a research agenda for the future.

Chapter 2

The impact of network topology on opinion convergence

We have defined the opinion dynamics and opinion convergence in Introduction. In terms of dynamical systems, the word 'convergence' is equivalent to identical synchronization of the networked system. In this chapter, we will investigate how the nodes holding different opinions communicate with each other and make the opinions converge on a network. The word 'synchronization' will be used instead of 'convergence' at some points when the theory of dynamical systems is used. The understanding of the relations between the opinion convergence and the structure of the complex networks is important. However, there are lacks of methodology to investigate the relationships. Despite all the simplifications that have been made on opinion models, there are several difficulties in the study of network topology. First, it is still unclear how generic features characterize the formation and topology of complex networks. Some properties have been used to describe network topology statistically, including the degree distri-

bution (K), the average shortest path length (L) and the clustering coefficient (C), etc. But there is no evidence that shows that they can reproduce all aspects of the networks or have a direct relation to any of the complex phenomena. Meanwhile, although the features are correlated to each other, none of them can be adjusted and observed as the exclusive factor when fixing the others. Due to the lack of analytical methods in complex networks, it is hard to prove that the statistical results obtained are credible. Once a new feature of complex network is discovered, the correlations between these features will have to be reconsidered.

In this chapter, we propose a methodology to study the time for the nodes to achieve the convergence. We will illustrate the relation between the time to convergence and the Laplacian spectrum of the complex networks. Later on, we attempt to discover the relations between the Laplacian spectrum and some typical statistical features of the network structure. In this way, we investigate the relation between opinion convergence and network structure. In the following, we will introduce how the opinion dynamics have been developed with the knowledge from different disciplines.

2.1 Development of opinion dynamics

Opinion models represent the development of social modelling and the application of mathematics to contribute to modellers of social systems which forms a new discipline of the computational social science. Opinion dynamics models study the evolution of two or more competing opinions through various approaches of mathematics and statistical physics [Saber et al., 2007]. A considerable amount of work has already

been carried out on how people in a system exchange their opinions in a pairwise way [Sznajd-Weron and Sznajd, 2000, Stauffer, 2001, Lambiotte et al., 2009, Lorenz, 2005].

In 2000, Sznajd [Sznajd-Weron and Sznajd, 2000, Sznajd-Weron, 2005] considered a so called binary Ising spin model to simulate a mechanism of decision making in a closed community. Considering N people as nodes within a network model, every node in this model may have one of two opinions, or choices ($A1$ or $A2$) and update its opinion due to its neighbour's opinions. For instance, if at time $t = t_1$, a node n_i holds opinion $A1$ but all neighbours hold $A2$, then at time $t = t_1 + 1$ n_i will change to $A2$. As shown by Sznajd, given a certain initial state and a rule for evolution, the simulation of the model always ends up with the same stable solution. The Sznajd model is not used to investigate the values of final opinions but to analyze the time evolution to the final opinion.

The voter model [Castellano et al., 2009] is a simple stochastic model used to describe the opinion evolution in time. Given N people in a system, at time t, every person holds an opinion $x_i(t), i = 1, 2, \ldots, N$, which came from their own and another person's opinion at time $t - 1$. Continuing backwards this way, we can find a relation of the form

$$x_i(t) = \eta_t x_i(0), \tag{2.1}$$

where η_t represents a random walk with a given transition probability $p(i, j), i =$

$1,2,\ldots N, j = 1,2,\ldots N$. This research introduced stochastic modelling of the voting process. In 2009, R.Lambiotte [Lambiotte et al., 2009] established a latent voter model based on the existing voter models. The new model assumes that the opinion status is not only a function of the previous opinion $x_i = 1$ or -1 and time t, but also by their activity, I (inactive) or A (active). The voter models introduce the stochastic process in the voting process and make the stable condition of the system a function of time t, but the relations between people have not been considered.

There are some other important binary models based on the Ising and the Potts models [Stauffer, 2001, Liggett, 2004, Clifford and Sudbury, 1985]. Usually, in the binary models, it is assumed that an individual is influenced by its nearest neighbours geographically. Monte Carlo simulations have been frequently used to describe the dynamical evolution from a given initial state. In a system of N individuals, at each step, one individual is selected at random to update its state. After m such steps, one Monte Carlo simulation is considered to be completed. These kinds of models focus on the stochastic communications between people, which always lead to the same final opinion condition given the same initial state. The structure of the interactions between people and the difference of influence among people were not discussed in this model.

Lorenz [Lorenz, 2005] established some continuous opinion models which may be considered as the early introduction of ordinary differential equations and dynamical system theory into opinion dynamics. For the model of continuous opinion dynamics, consider N nodes who change their opinions. Given $x_i(t)$ as an opinion profile at time step t, where x_i represents the opinion of node i. A matrix $A(x(t),t)$, called the

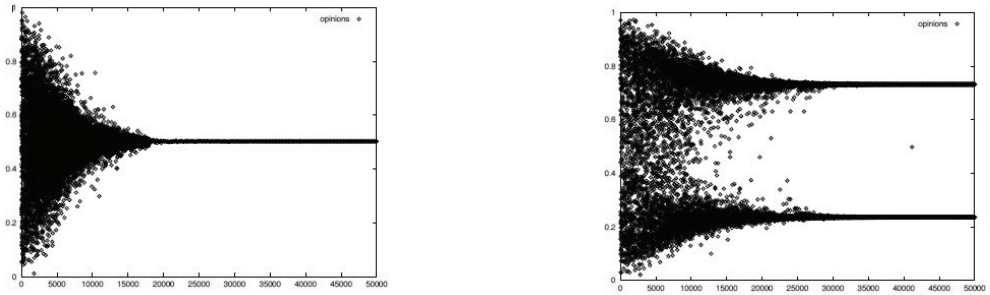
confidence matrix, is defined so that the entry $A(x(t),t)_{ij}$ represents the weight (or confidence) that node i impacts the opinion of node j at time t. The dynamics of the system is governed by

$$x(t+1) = A(x(t),t)x(t). \tag{2.2}$$

If there is a time t_c when $A(x(t_c),t_c) = A(x(t),t)$, it is implied that $x(t)$ converges to a constant consensus.

The confidence model is a leap from linear opinion models to nonlinear ones. The simplest is the bounded confidence model [Hegselmann and Krause, 2000]. In this model, people cannot communicate with others with opinions too close or too far away from their own, which makes the dynamics among all agents a nonlinear one. In [Deffuant et al., 2000] only pairwise interactions are simulated, whereas in [Hegselmann and Krause, 2002] an agent takes into account an average influence of all its neighbours. In the Deffuant model, which has been cited a lot in the recent decade, a population of N agents with continuous opinions $x_i, i = 1,2\cdots,N$ interact with each other under some principles. At each time step any two randomly chosen agents meet. They re-adjust their opinion when their difference of opinions is smaller in magnitude than a threshold. Suppose that two agents have opinions x_i and x_j and that $|x_i - x_j| < d$; the pinions are then adjusted as

$$x_i = x_i + a_{ji}(x_j - x_i), \tag{2.3}$$



(a) $d = 0.2$ (b) $d = 0.5$

Figure 2.1: (a) $d = 0.2$ and (b) $d = 0.5$ in the confidence model of [Deffuant et al., 2000]

$$x_j = x_j + a_{ij}(x_i - x_j), \tag{2.4}$$

where a_{ji} and a_{ij} are the ability of agent j to influence agent i and ability of agent i to influence j respectively, taken between 0 and 1. The rationale for the threshold condition is that agents only interact when their opinions are already close enough; otherwise they do not even bother to discuss their differences. The confidence bound d influences the opinion profile after the system has reached a stable condition. Given $N = 1000$, different values of d such as $d = 0.2$ and $d = 0.5$ lead to different results as shown in Figure 2.1.

Some consequent models are studied by [Lorenz and Urbig, 2007]. More recently, [Hegselmann and Krause, 2004] studied other variants of averaging within the bounded confidence model. [Stauffer et al., 2004] proposed a discrete version of the bounded confidence model and [Urbig, 2003] proposed a version where agents have a discrete expression of their continuous opinions. This model involves by nonlinear opinion dynamics and uses some new methods to analyze the process.

Until now, for the micro level in opinion problems, the pairwise communication has been studied intensively. Researchers wanted to establish a general method to describe how people change their minds as a result of communication. Several mathematical tools have been used, including Monte Carlo methods, matrix theory and stochastic processes. However, studies at the macro level, like the topology of a group of interacting opinions, are still rare. Particularly, heterogeneous models, which are closer to real life, are not well developed due to the lack of tools to analyze this kind of systems beyond the mean-field theory approach. In this thesis, further networked opinion models are established, and the research methodology discussed.

2.1.1 Linear opinion dynamics on complex networks

Recently, attention has been focused on the systematic substrate where the opinion dynamics take place. Complex networks have been used widely to describe the relations between people, and how the network topology affects the opinion dynamics has been studied intensively [Hong et al., 2004, Olfati-Saber, 2005, Saber et al., 2007, Arenas et al., 2008]. In this kind of studies, opinion dynamics are connected to the problem of time scales in synchronization. It has been observed that opinion synchronization becomes an exponential process after some short transients on all kinds of network structures. Thus the distance between opinions is defined as

$$d(t) = \max\{dist\{x_i(t), x_j(t)\}\}, \tag{2.5}$$

which decays towards the synchronous state as

$$d(t) \sim exp(-t/\tau), \tag{2.6}$$

in the long time limit. However, so far the relation between synchronization speed and the complex network structures has not been studied analytically. Therefore, in this chapter, we intend to apply multiple methods to look into this problem.

Curtis and Smith [Curtis and Smith, 2008] developed a linear model to describe persuasion between people. In this model, in a system of N persons, each one holds an initial opinion x_i^0 and interacts with each other according to the model

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} b_j a_{ij}(x_j - x_i), i = 1,2\ldots N, \tag{2.7}$$

where the constant b_j is the person j's ability to influence others. The variable x_j is person j's opinion at any time t. The entry of a_{ij} is 1 if i talks to j, and 0 otherwise. The N people make a network with the adjacency matrix A. In this study, we will analyze the matrix A and its corresponding Laplacian matrix to see how the network structures impact the opinion evolution on complex networks.

In this chapter, we carry out a group of computer simulations followed by an analysis from the point of view of graph theory and matrix theory. Besides variations of the degree distribution (K), the average path length (L) and the clustering coefficient (C), we put the network size (N), the number of connections (m) and degree correlation (Pr)

into consideration. We find that it is hard to claim any general rule experimentally or determine any casual relationships between the network behaviour and the structural features as characterised by these measures. We have discovered how the structural features restrict each other and impact the opinion dynamics integrally.

In this chapter, we will look into the speed of opinion evolution on undirected-unweighted networks. The governing equation is the linear Equation (2.7). The equation set will always achieve a solution x_s, which is the consensus in the opinion dynamics if every constant b_i is positive. This point will be proved in Section 2.5. The network structure will determine how fast the process is. In this chapter, we will obtain some basic understanding of how the network's complexity affects the convergence of opinions. The conclusions and methods in this chapter will be used in the studies of the linear dynamics on directed-weighted networks (Chapter 3), and the nonlinear dynamical networks (Chapter 4).

2.1.2 Research organization

Chapter 2 is structured as follows: In Section 2.2, we give a literature review of complex networks and graph theory. In Section 2.3, we list the definitions and abbreviations which will be used through the research. In Section 2.4, we establish a group of typical complex networks. In Section 2.5, we set linear dynamics to govern the evolution of the opinions on different networks. Then we simulate the opinion evolution on different networks and record the opinion convergence speed. In Section 2.6, we test the relations between the opinion convergence time and a series of network features, like

the average shortest path length (L), the cluster coefficient (C), the degree correlation (Pr), the network size (N) and density (m). We point out some issues from the previous studies and the reasons of them. We give a discussion in terms of graph theory and matrix theory. Section 2.7 is the conclusion of this chapter.

2.2 Research methodology

The methodology of complex network and graph theory is adopted in this research. We use the complex network to simulate a group of people with their opinions. Then the models of dynamical equations are built to describe the opinion evolution on the network. The graph theory and matrix theory are used as tools to transfer the attention from the network and its adjacency matrix to the spectrum of the Laplacian matrix, which makes it possible to analyze the network behaviour.

2.2.1 Graph theory

As established in [Sinabra, 2008], graph theory is the natural framework for the exact mathematical treatment of complex networks. Formally, a complex network is represented as a graph with particular dynamical behaviours. In this section, we will introduce some basic concepts which will be used throughout the whole thesis.

2.2.1.1 Classifications of graphs in the view of networks

Mathematically, a network is a graph with different contexts of the nodes and connections. The definitions of graphs which are used in this thesis are given as follows:

- Undirected graph: A undirected graph is $G = (N, L)$ with two sets N, the nodes

and L, the links in the graph. In these kind of networks, if there is a link from node i to node j denoted as l_{ij}, there always exist $l_{ij} = l_{ji} = 1, l_{ij} \in L, L_{ji} \in L$, otherwise $l_{ij} = l_{ji} = 0$.

- Directed graph: In this case l_{ij} means a link from node i to j. When $l_{ij} = 1$, the link l_{ji} may be 1 or 0. The degree of a node i includes the out-degree $deg_{out}(i) = \sum_{i \neq j}^{N} l_{ij}$ and the in-degree $deg_{in}(i) = \sum_{i \neq j}^{N} l_{ji}$.

- Undirected-unweighted graph: Based on undirected graph, with the weight $w_{ij} = w_{ji} = 1$ on every link, representing the coupling strength between node i and j.

- Directed-weighted graph: Base on directed graph, with the weight w_{ij} on l_{ij} and w_{ji} on l_{ji} respectively.

In this chapter, we will focus on the undirected-unweighted graphs.

2.2.1.2 Adjacency matrix and Laplacian matrix

Any undirected-unweighted graph G can be represented by its adjacency matrix $A(G)$, which is a real symmetric matrix: $a_{ij} = a_{ji} = 1$, if vertices i and j are connected, or 0 otherwise. The main algebraic tool that we will use for the analysis of graphs will be the spectrum, i.e, the set of eigenvalues of the graphs adjacency matrix. The Laplacian matrix generates the information of network structure and the dynamical behaviour between the nodes on the network. The spectral properties of Laplacian matrix are widely used in the analysis of linear models on fixed network [Farkas et al., 2001, Hong et al., 2004, Jost and Joy, 2001] . Given a network with N nodes de-

scribed as a graph $G = (V, E)$, the vertices set (which indicate nodes in the network) is $V = V(G) = \{v_1, v_2, \cdots v_N\}$ and the edges set (which means connections in network) is $E = E(G) = \{e_1, e_2, \cdots e_m\}$. Each edge $e_j = \{v_i, v_k\}$ has two 'ends' v_i and v_k. For undirected and directed graphs, the Laplacian matrix, $L(G)$ is defined by $L(G) = D(G) - A(G)$ [Kelner, 2009]. The diagonal matrix $D(G)$ represents the degree of every node as the diagonal element in the corresponding row. The adjacency matrix $A(G)$ has all elements as 1 or 0.

2.2.1.3 Spectrum of Complex network

The spectrum of the graphs Laplacian matrix is also called the spectrum of the graph [Merris, 1994]. The eigenvalues of $L(G)$, as the spectrum of the graph, reflects the structure and some other aspects of the graph. One of the most important tool in this study is the analysis of the spectrum. The number of zero eigenvalues of the Laplacian matrix is equal to the number of strongly connected components of the network. In this research, we focus on the networks without isolated parts. Therefore, the rank of L is at most $N-1$ and there is always a zero eigenvalue, and the elements in the corresponding eigenvector are equal to each other [Kelner, 2009, Harary, 1969]. Among all nontrivial eigenvalues, the one closest to the zero eigenvalue is called 'algebraic connectivity' λ_{ac} and it predicts whether the system will get synchronized. For the normal Laplacian matrix of an unweighted-undirected in the traditional graph theory, the diagonal elements are all positive and the off-diagonal ones are negative. The eigenvalues of L can be estimated by the theory of Gerschgorin

circle. For the nonzero eigenvalue λ_i of L, $|\lambda_i - a_{ii}| \preceq |\sum_{j=1,j\neq i}^{N} a_{ij}|$. Therefore we have $a_{ii} - |\sum_{j=1,j\neq i}^{N} a_{ij}| \preceq \lambda_i \preceq a_{ii} + |\sum_{j=1,j\neq i}^{N} a_{ij}|$. Since $a_{ii} = |\sum_{j=1,j\neq i}^{N} a_{ij}|$ for L, it is guaranteed that all eigenvalues for L are nonnegative.

2.2.1.4 Measurement of opinion convergence and 'consensus'

The classic measurement of opinion convergence speed and stability is based on a symmetric Laplacian matrix [Li and Chen, 2003, Chavez et al., 2005]. Previously, the Laplacian matrix is defined with all diagonal entries non-negative and all off-diagonal entries non-positive. In recent years, the weighted networks came into the sight of the researchers and the use of the Laplacian matrix has been broadened. When the entries of an adjacency matrix are all positive, the eigenvalues of the Laplacian matrix are $0 = \lambda_1 \leq \lambda_2 \preceq \ldots \preceq \lambda_N$. In the theory of dynamical systems, it is typical that the Laplacian matrix L_d of a system is expressed as $L_d = -L$. Hence, the eigenvalues are $0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_N$. In this thesis, we will refer the Laplacian matrix in the way of L_d. We will discuss this point in Section 2.5 and 2.6. The ratio of the largest nonzero eigenvalue to the smallest one $R = \lambda_2/\lambda_N$ can be used to judge the synchronizing ability of the network if the synchronized region is bounded. The larger value means the better opinion convergence. For the boundless case, the largest nonzero eigenvalue λ_2 can play the same role, that is, the larger value of λ_2 implies stronger opinion convergence. However, when the entries of the adjacency matrix are not all positive, the negative eigenvalues may occur in the system. In that case, we will use λ_{ac}, the algebraic connectivity instead of λ_2 to represent the largest nontrivial eigenvalue.

2.2.2 Complex networks

A complex network is a graph (network) with physical statistical features. It consists of nodes executing particular behaviours and connections representing relations and interactions between nodes. For example, in biology, a cell may be described as a complex network of chemicals connected by chemical reactions; the Internet is a complex network of routers and computers linked by various physical or wireless links; the opinion model in our research is also a complex network where agents (people) talk to each other through the connections. The topology of a complex network produces statistical features that do not occur in simple networks such as lattices but often occur in real life. Hence a complex network is an instrumental tool to describe social and natural complex phenomena, with matrix theory, spectrum analysis and control theory as subsequent analyzing tools [Newman, 2003, Albert and Barabasi, 2001, Olfati-Saber, 2005]. In this section, we will focus on the physical features of complex networks relevant to our research.

Some definitions are taken directly from [Newman, 2003, Albert and Barabasi, 2001].

2.2.2.1 Degree distribution

The degree of a node in an N-node network is the number of connections the node has and through which it communicates with other nodes. The degree distribution $P(k)$ of a N-node network is then defined to be the fraction of nodes in the network with degree k. Thus if there are N nodes in a network and n_k of them have degree k,

we have $P(k) = n_k/N$. We will introduce the random graph, one of the several typical complex networks first and then take it as an example to discuss the relevance of degree distribution.

2.2.2.2 Random graph

A random graph (*RG*) consists of N nodes and some links between them at random. A random graph can be generated by a probability distribution, or by a random process. A typical random graph is the *ER* network, developed by Erdos and Renyi [Erdos and Ranyi, 1959]. It is defined as N nodes and a constant probability p_{er} that any two nodes are connected. When the network is large enough, $N \to \infty$ and $N-1 \approx N$, the average degree of nodes is $\langle k \rangle = p_{er}(N)$, and the degree distribution $P(k)$, is a Poisson distribution.

In an *RG* network, with connection probability p_{er}, the degree k_i of a node i follows a Poisson Binomial distribution with parameters $N-1$ and p_{er}

$$P(k_i = k) = C_{N-1}^{k}(1-p_{er})^{N-1-k}p_{er}^{k}. \tag{2.8}$$

This probability represents the number of ways in which k edges can be drawn from a certain node. In Equation (2.8) the probability of k edges is p_{er}^{k}, the probability of the absence of additional edges is $(1-p_{er})^{N-1-k}$, and there are C_{N-1}^{k} equivalent ways of selecting the k end points for these edges. Furthermore, if i and j are different nodes, it is highly possible for $P(k_i = k)$ and $P(k_j = k)$ to be close to each other. The degree of nodes and the degree distribution will be important measurements of the coupling

ability, the density and some other features of a complex network. We classify complex networks as follows.

2.2.2.3 Small world

The word 'small world' has two definitions. Usually it appears as a 'small-world network' , which is a type of a mathematical graph in which most nodes are not neighbours of one another, but most nodes can be reached from every other node by a small number of steps via the links of the network. Sometimes, this is a feature of a network combining a high clustering coefficient and short path length. Watts and Strogatz [Watts and Strogatz, 1998] proposed a novel graph model, currently named the Watts and Strogatz model (WS). In this section, we discuss the 'small-world network', which is based on a rewiring procedure of the edges implemented with a probability p_{ws}. Start from a regular network where each node connects to its $2m$ neighbours, $m = 1, 2, \cdots N-1$ in a N node network. Then, for every node, each connection is rewired to a randomly chosen node with a probability p_{ws}, and preserved with a probability $1 - p_{ws}$. Obviously, when $p_{ws} = 0$, the regular lattice remains the same, and when $p_{ws} = 1$, the network evolves to another extreme and becomes a random graph.

The networks with small world feature tend to contain cliques, and near-cliques, by which we mean sub-networks with connections between almost any two nodes within them. Secondly, most pairs of nodes will be connected by at least one short path.

Several other properties are often associated with small-world networks. Typically, the network does not contain a small number of hubs with high degrees, which serve

as the common connections mediating the short path lengths between other edges.

2.2.2.4 Power-law and scale-free networks

A power-law is a mathematical relationship between two quantities. When the frequency of an event varies as a power of some attribute of that event (e.g. its size), the frequency is said to follow a power law. A complex network may have small number of nodes with large degree and large number of other nodes with small degree. A scale-free network is a network whose degree distribution $P(k)$ follows a power law, at least asymptotically when $k \to \infty$. That is, the fraction $P(k)$ of nodes in the network having k connections to other nodes for large values of k behaves as $P(k) \sim k^{-\gamma}$, with $\gamma \in (2,4)$ approximately as a statistical result.

A lot of real networks, such as the Internet, are characterized by a power-law distribution. This observation leads to research on the dynamics, such as opinion dynamics of the disordered organization of real-world systems. The main theory of complex network is a focus on the probability p of connection among nodes, large amounts of nodes holding minority of connections while some 'hubs' connect to a significant fraction of the total number of connections. An explicit introduction of how to generate a scale-free network will be given in the next Section.

2.2.2.5 Subgraphs

A random graph is obtained by starting with a set of N vertices and adding edges between them at random. The emergence of subgraphs within random graphs was first studied by Erdos and Renyi in 1959.

A graph G_1 consisting of a set V_1 of nodes and a set E_1 of edges is a subgraph of a graph $G = (V, E)$ if all nodes in V_1 are also nodes of V and all edges in E_1 are also edges of E.

2.2.2.6 Clustering coefficient

The clustering coefficient measures the extent a node's neighbours connecting to each other. In social life, if the friends of one person also know that person's friends, we may say the group (or the social network) has a high clustering coefficient. Complex networks exhibit a large degree of clustering. In a random graph, the probability that two neighbours of one node are connected is equal to the probability that two randomly selected nodes are connected. Consequently the clustering coefficient of the graph is

$$C_{rand} = P = \langle k \rangle / N. \tag{2.9}$$

If we use the concept of subgraph from the last section, the clustering coefficient of any of the subgraph G_1 can also be obtained from this equation. Only the connections between nodes in G_1 will be included in the calculation of the degree of a node.

2.2.2.7 Average path length

In the complex network, the path length between two nodes indicates the number of nodes one particular node has to pass when finding another particular node. For instance, in small world networks (Figure 2.2), the shortcuts effectively reduce the path when a node searches for another one within the network. The average path length in a

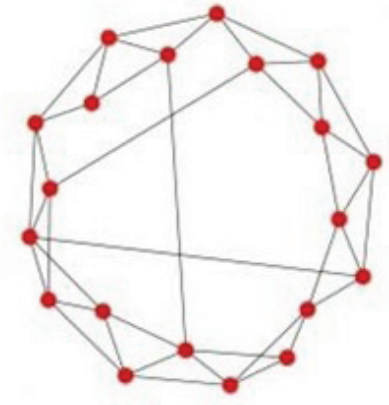


Figure 2.2: Small world network with short path length

network is often used to measure the coupling ability of the network. Numerical analysis we undertook on random graphs and small world models suggests that lower overall average shortest path length leads to a faster consensus of opinions. A discussion about this will be given in Section 2.5.1.

2.2.2.8 Degree correlation

The degree correlation is how a node's degree is correlated with the degree of the neighbouring nodes. It is calculated by the Pearson correlation coefficient Pr, in a network with N nodes and m connections as

$$Pr = \frac{\sum_{i=1}^{N} m^{-1} j_i k_i - \sum_{i=1}^{N} [m^{-1}(j_i + k_i)]^2}{\sum_{i=1}^{N} m^{-1}(j_i^2 + k_i^2) - \sum_{i=1}^{N} [m^{-1}(j_i + k_i)]^2}, \tag{2.10}$$

where j_i and k_i are the degree of vertices at the end of the ith connection, $i = 1, 2, \cdots m$ and m is the number of connections.

2.3 Definitions and abbreviations

- RG: Random graph.
- WS: Watts and Strogatz model as one example of small world network.
- SF: Scale-free network.
- BA: The Barabasi-Albert model.
- $ASSF$: The assortative scale free network.
- $DSSF$: The disassortative scale free network.
- k: The degree of node.
- L : The average shortest path length, or the average shortest distance.
- C: The clustering coefficient.
- Pr: The degree correlation.
- M: The negative Laplacian matrix.
- R: The eigenratio, the ratio of the largest nontrivial eigenvalue of M and the smallest.
- t_c: The time for the opinions to converge.
- N: Network size.
- m: Total number of connections.

2.4 Architecture and the statistical characteristics of complex networks

In this section, we will build five typical types of complex networks. We simulate the process of opinion evolutions towards consensus on these five networks. The time t_c for the opinions to converge on each network is recorded and compared. We prove that the eigenratio R of the Laplacian matrix determines the ability for a network to support opinion convergence. We will show that the larger R is, the shorter t_c will be. Subsequent research on how the topology impacts t will be replaced by how the topology impacts R. In the next section, we will use some statistical characteristics, like the average path length L to represent network topology. We classify the networks into two groups. In each group, the networks have some fixed characteristics in common, and we build the two groups of the networks by adjusting the parameters

The first group the WS (small world network) and the RG (random graph), which are both derived from the regular network in this study.

The second group the BA (Barabasi-Albert model), the $ASSF$ (assortative scale free network) and the $DSSF$ (disassortative scale free network).

2.4.1 The RG and WS networks

Figure 2.3 from [Watts and Strogatz, 1998] illustrates the relations between the regular network, the small world network and the random graph. In a N-node regular network, nodes are arranged in a circle or a lattice regularly. For each node in a circle, every

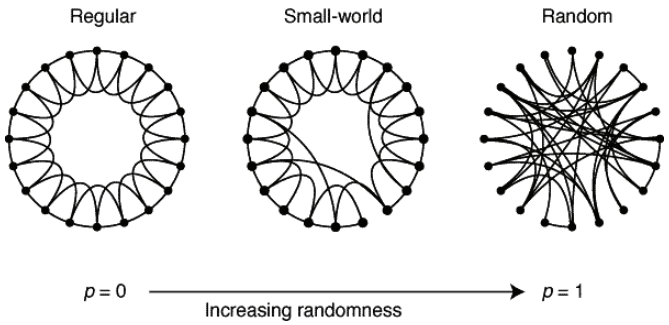
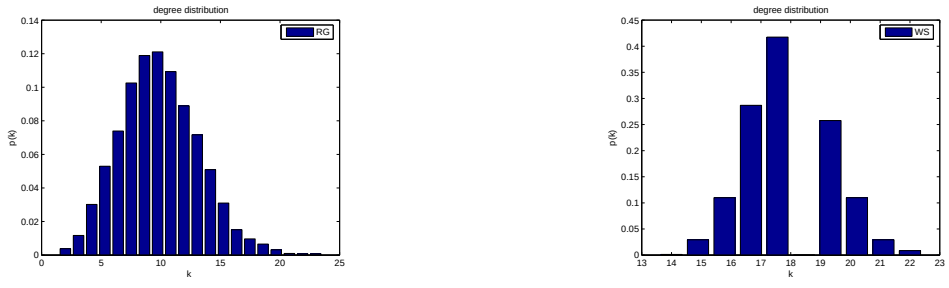


Figure 2.3: From regular network to random graph [Albert and Barabasi, 2001]

other node can be achieved by k steps geographically, $k \leq N-1$. Given a certain value $k \leq N-1$, for each node i, all the nodes within k steps clockwise are its k left neighbours, and those in anticlockwise are its k right neighbours. We connect all the neighbours and build the regular network.

Then, for each node, every connection will lose the other vertex and be rewired to a randomly chosen node with a probability p, and be preserved with a probability $1-p$. Obviously, when $p = 0$, the regular network remains the same, and when $p = 1$, the network evolves to another extreme and becomes a random graph. The word 'small' in 'small network' describes the average path length L of the network. If we fix the network size N, the total connection number m and the neighbour number $2k$ of every node, when p increases and crosses the critical point $p \geq 1/2mN$, the WS behaviour emerges and the L decreases to $L \sim \frac{ln2mpN}{4m^2p}$. The phenomena will be explained in Section 2.5.1. The degree distributions of RG and WS are shown in Figure 2.4. They are both built by rewiring the initial 1000-node regular network with an identical sequence of degrees. Every connection that is broken and rewired increases the inequality of degrees. For these 1000-node networks, we consider that RG holds a Poisson distribu-



(a) RG (b) WS

Figure 2.4: Degree distributions of *RG* and *WS*.

tion and the *WS* holds the binomial distribution for the degrees. When $N \to \infty$, both of them can be regarded as Gaussian distribution.

2.4.2 The *BA*, *ASSF* and *DSSF* networks

Before we generate any of the *SF* networks, we discuss the degree sequence held by all of them. The power-law, which defines the *SF* networks, has the discrete form $P(k) = k^{-\lambda}$ ($\lambda = 3$ in this study). The distribution can be obtained by normalization $P(k) \propto \frac{k^{-\lambda}}{\sum_{n=0}^{\infty}(n+k_{min})^{-\lambda}}$ and $\sum_n^{\infty} P(n) = 1$. Since it's meaningless if $k \to \infty$ in real networks, the k_{min} should be defined to estimate a degree sequence obeying the power-law. In this study, to avert the possibility of the existence of an isolated cluster with two nodes, we define $k_{min} = 4$. We choose $N = 1000$ and determine the degree sequence $D = \{d_1, d_2, \cdots, d_{1000}\}$ by the normalization we mentioned before. The node i holds d_i numbers of prospective links initially. The prospective links connect randomly to make an actual link. The *BA* is established after all the half links are connected.

Then we generate the *ASSF* by rewiring the *BA*, as shown in Figure 2.5:

1. At each step, two links with their four ends of the network are chosen at random;

(a) ASSF (b) DSSF

Figure 2.5: (a)the *ASSF* algorithm and (b)the *DSSF* algorithm.

2. The links between these nodes are rewired in such a way that one new link connects the two nodes with the smaller degrees and the other connects the two nodes with the larger degrees;

3. Repeat the two previous steps until a desired degree correlation Pr, as a measure of the assortativity is achieved

$$Pr = \frac{\sum_{i=1}^{N} m^{-1} j_i k_i - \sum_{i=1}^{N} [m^{-1}(j_i + k_i)]^2}{\sum_{i=1}^{N} m^{-1}(j_i^2 + k_i^2) - \sum_{i=1}^{N} [m^{-1}(j_i + k_i)]^2}, \tag{2.11}$$

where j_i and k_i are the degrees of the nodes at the end of the ith connection, $i = 1, 2, \cdots m$. When $Pr = 0$, the probability that a link is connected to a node with a certain degree is independent from the degree of the attached node and the network is uncorrelated. On the contrary, $Pr = 1$ indicates the network is totally assortative.

In Figure 2.5, choosing two random connections as shown in Figure 2.5(a), we rewire them to guarantee one of them connects the two nodes with larger degrees while the other connects the two nodes with smaller degree as shown in Figure 2.5(b). In

contrary, the algorithm of *DSSF* is to take the step (b) before (a): choosing two random connections as shown in Figure 2.5(b), rewiring to connect the largest degree node and the smallest degree one. Then the two remaining nodes are connected.

To generate a *DSSF* network, two random links are chosen for the *ASSF* process. However, the links are now rewired to make sure that one link connects the node with the smallest degree to the one with the largest degree while the other connects the two remaining nodes. These steps are repeated until no further change can be made during the rewiring. These algorithms ensure that the *BA*, *ASSF* and *DSSF* networks hold the same degree sequence.

For most real life networks, the degree correlation Pr ranges from -0.3 to 0.3 [Newman, 2002]. Therefore, in this study, we set $Pr = 0.3$ for *ASSF* network and $Pr = -0.3$ for the *DSSF* network. The degree distributions shared by all the three networks are shown in Figure 2.6. However, the way to make connections causes significant difference in their topology and dynamical behaviour.

2.5 Formulation of the linear models

The opinion model from [Curtis and Smith, 2008] will be used through the whole study. Consider that a N-node network with the adjacency matrix is $A = a_{ij}$. The entry of a_{ij} is 1 when there is a connection between nodes i and j, and 0 otherwise. The influence ability b_i of node i is set as 1 for all nodes in this study. Every node i holds

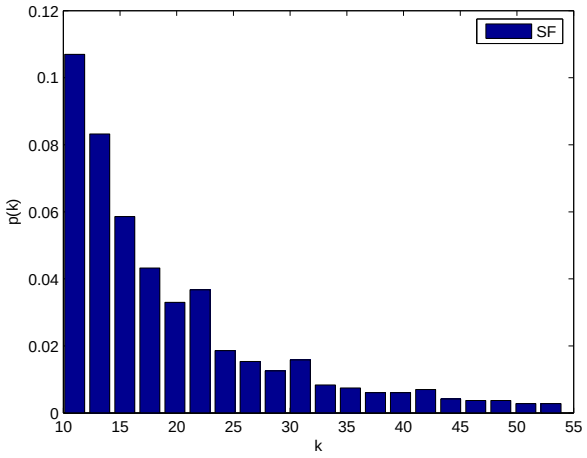


Figure 2.6: The *BA*, *ASSF* and *DSSF* share the same power-law distribution.

an opinion x_i evolving by time t and the rule is

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} b_j a_{ij}(x_j - x_i), i = 1,2\ldots N, \tag{2.12}$$

where the constant b_i is the ith entry of vector b and represents the influence ability which the ith node holds.

As defined in the graph theory [Merris, 1994, Li and Chen, 2003, Chavez et al., 2005], the Laplacian matrix $L = D - A$, where D is the diagonal matrix of degrees and A is the adjacency matrix of the graph [Kelner, 2009]. In this thesis, with regards to the original model from [Curtis and Smith, 2008] and the typical way of expression in the dynamical system, we have the influence matrix $M = -L$ as the Laplacian matrix of the opinion system. We build M with a diagonal matrix $B = diag(b)$ and an all one

vector **1** as follows

$$M = BA - diag(BA\mathbf{1}) = B\widetilde{M}, \tag{2.13}$$

which contains the information about the adjacency matrix and the influence ability between nodes. The matrix $\widetilde{M}$ is the symmetric Laplacian matrix. Consequently, the system can be written as the linear vector Equation (2.10)

$$\frac{dX}{dt} = MX = B\widetilde{M}X \tag{2.14}$$

where the ith entry of the vector X is the opinion x_i of node i.

We obtain from the traditional graph theory that there is always an all 0 column in L, which guaranteed a zero eigenvalue, and the elements in its corresponding eigenvector are equal to each other. If the graph is connected, the null space is 1-dimensional and spanned by the vector 1.

Proof : Let $x \in null(L)$, i.e. $Lx = 0$. It can be obtained from [Kelner, 2009] that

$$x^T Lx = \sum_{(i,j \in E)} (x_i - x_j)^2 = 0, \tag{2.15}$$

which means all x_is are equal. Thus every member of the mull space is a multiple of **1**.

The L is symmetric and the off-diagonal elements are all negative. Except for the zero eigenvalue, the other eigenvalues are all positive [Kelner, 2009]. In Equation (2.13), if we set every b_i positive, the matrix M will have one eigenvalue as zero and

the other eigenvalues all negative. It is necessary that the b_i equals to each other to make A and M symmetric. Therefore, the eigenvalues and eigenvectors of M are all real [Kelner, 2009]. The b_is are all set as 1 in this chapter. There exist a consensus x_s so that

$$\lim_{t\to\infty} |x_i(t) - x_s| = 0. \tag{2.16}$$

We will prove this point in Equations (2.18-2.20).

2.5.1 Simulation

We record the opinion evolutions on the five networks. Each of the networks consists of 1000 nodes. We use the standard deviation *sd* of the opinions at time t to measure the difference between opinions

$$\sigma = \sqrt{\sum_{i=1}^{N} (x_i - \overline{X})^2}, \tag{2.17}$$

where $\overline{X}$ is the mean of the opinions. When $\sigma = 0$, the opinions are identical and the consensus is achieved. With the same group of initial opinions $X^0 = (x_1^0, x_2^0, \cdots, x_N^0)$, the various topologies lead to same consensus $x_s = x_1^t = x_2^t = \cdots = x_N^t$ at time t_c, which is the average of the initial opinions. Given the dynamical Equation(2.12), when $t \to \infty$, consensus is achieved

$$X = e^{Mt} X^0 = P e^{\Lambda t} P^{-1} X^0, \tag{2.18}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$, the diagonal matrix of eigenvalues of M, and P has the corresponding eigenvectors as columns. As M is symmetric, all eigenvectors are orthogonal and the one corresponding to zero eigenvalue can be written as $(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots \frac{1}{\sqrt{N}}, \cdots \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}})^T$. So

$$Pe^{\Lambda t}P^T\mathbf{1} = Pe^{\Lambda t}(0,0,\cdots 0,\cdots 0,1)^T = P(0,0,\cdots 0,\cdots 0,1)^T = \mathbf{1}, \tag{2.19}$$

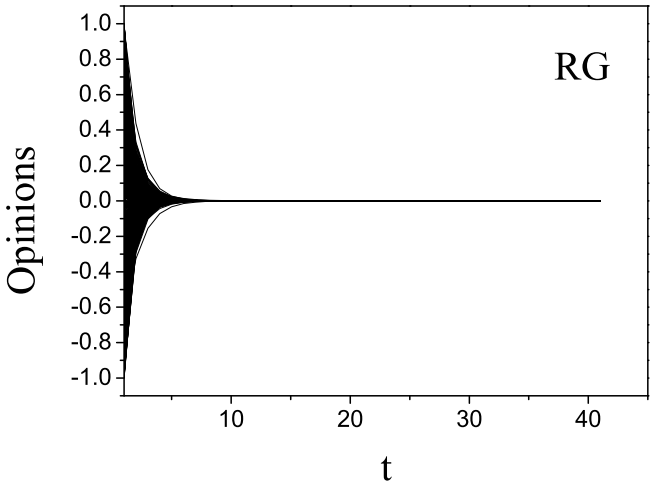
for all t.

Since we have $P^{-1} = P^T$, then

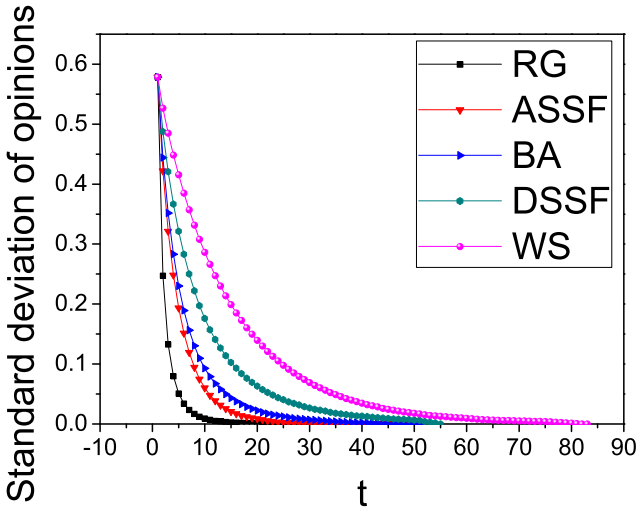
$$N\overline{X} = X^T\mathbf{1} = (P^{-1}X^0)^T e^{\Lambda t}P^T\mathbf{1} = (X^0)^T\mathbf{1} = N\overline{X^0}. \tag{2.20}$$

It is not difficult to find that the consensus is $x_E = x_1^t = x_2^t = \cdots = x_N^t = \overline{X^0}/N$ when the system evolution is stable.

We use $ODE45$ in Matlab to solve Equation (2.12). In Figure 2.7, t denotes the $ODE45$. If the time is long enough, all the opinions will converge at a value close to 0. Since the consensus can't be exactly 0, we restrict the opinion values in 8 digits after decimal points and the opinions can converge at 0. In Figure 2.7, the *RG* supports the fastest convergence of opinions among all networks. For the standard deviations σ of the opinions change along the processes on the five networks, when σ in a network becomes zero, all the opinions become identical and the system is stable.



(a) Opinion evolution



(b) Standard deviations σ of the opinions

Figure 2.7: The figure a) shows the opinion evolution process in the *RG* network. The figure b) shows how the standard deviations σ of the opinions change along the processes on the five networks.

2.5.2 The relation between the Laplacian spectrum and the convergence time t_c

An important hypothesis in this thesis is that all the networks we talk about are connected without any isolated nodes. In the N-node fully-connected network, if the influence ability between all the nodes are positive, the Laplacian matrix M has only one zero eigenvalue as the largest one. Every isolated node will correspond to another zero eigenvalue of M.

In some references, the eigenratio $R = \lambda_{ac}/\lambda_N$ is used instead of λ_{ac}. A test is taken on Matlab to compare between $-R$ and λ_{ac} in our model (see Figure 2.8). We use $-R$ instead of R to make it easier to observe.

We use a directed random graph with a strong positive node of degree K and influence ability α. The algorithm to compare λ_{ac} and R is:

1. Given the ER network, we find the degree of each node and list them in decreasing order. We then divide the list into ten equal intervals. For each interval, we pick at random one node from and use this to generate the vector K.

2. We made the choice $b_i = 1$ for every node i, except one with the influence ability $\alpha < 0$.

3. Start from the first node in K. This node will be given 10 possible integer values influence ability α from -1 to -10. So fix the degree, there are 10 individual tests in every round, which makes 100 individual tests in the total when we go through all the possible degrees.

4. Record the eigenvalues of the Laplacian matrix in every round with a particular degree and influence ability.

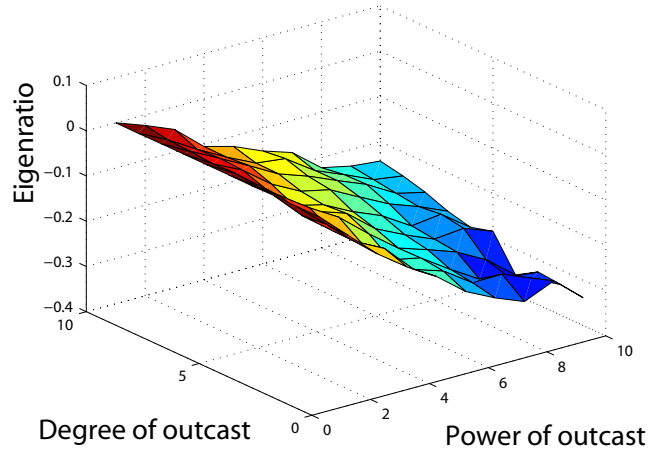
5. Illustrate the λ_{ac} and R in two graphs.

Since the change of λ_{ac} is much faster than λ_N, the two parameters show almost the same tendency when the factors are adjusted. We can observe the information from both figures of whether the network is synchronizing and how it is impacted by the degree K and the influence ability α. As far as we want to observe from this parameter, the R and λ_{ac} don't show significant difference from each other.

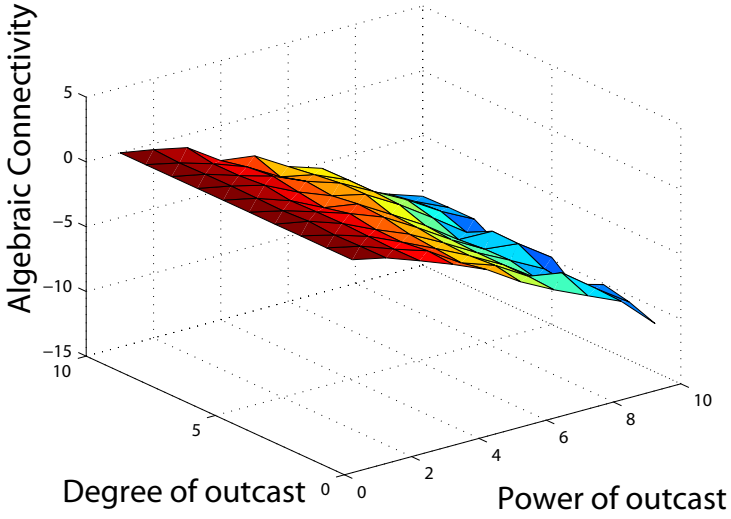
The λ_N controls the quickest convergence in the system while the λ_{ac} controls the slowest convergence. We choose the eigenratio $R = \lambda_{ac}/\lambda_N$ to measure the ability of the networks to support the opinion convergence. Given the same node number N and m connections, for the five networks, the larger R is, the faster the convergence occurs.

In Figure 2.9, we illustrate the eigenvalues of the five networks and the relation between R and t.

Since it is impossible to build a relation between time t and the topology directly, we will use R as an alternative to t in the next section. The relations between several topological characteristics and R will be studied.



(a) $-R$



(b) λ_{ac}

Figure 2.8: Comparison between $-R$ and λ_{ac} in same ER.

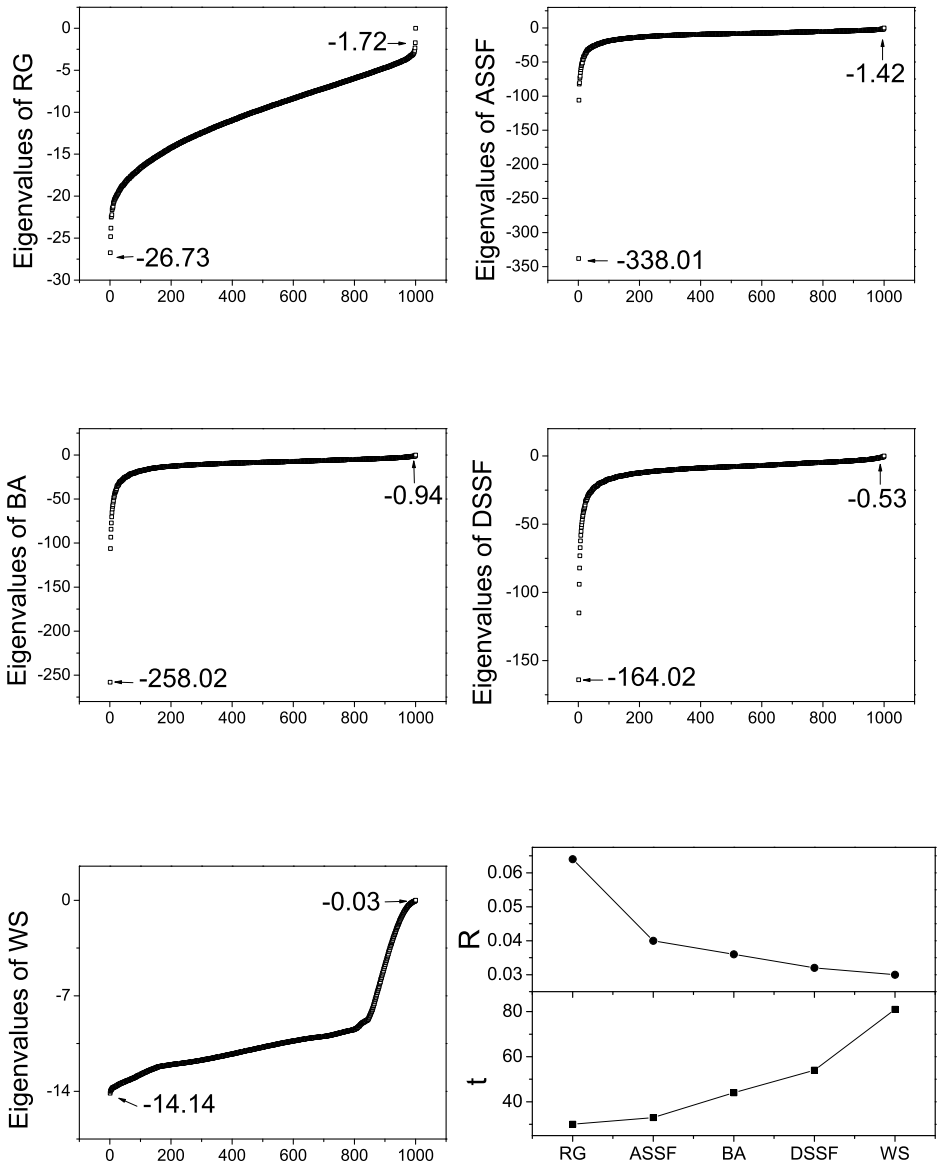


Figure 2.9: The eigenvalues are ranked from small to large for each network. The last figure shows the relation between eigenratio R and convergence time t.

2.6 The impact of the network topology on the opinion convergence

The relations between structural characteristics and the opinion convergence time have been explored intensively. But the observations summarized from the simulations are still in need of further investigation. In this section, we will look into the relations between convergence time and some typical network features. We will give a more objective simulation followed by a graph theoretical analysis.

2.6.1 The average shortest path length

The concept of the average shortest path length (L) originated from the concept of 'distance' in the graph theory [Harary, 1969]. The distance between two vertices in a graph is the number of edges in a shortest path connecting them. Later it was introduced into the theory of complex networks to define the average number of steps along the shortest paths for all possible pairs of network nodes [Albert and Barabasi, 2001]. The average shortest path length is a measure of the efficiency of transport on a network. Some examples use the average number of clicks which will lead you from one website to another, or the number of people you will have to communicate through on average, to contact a complete stranger. Several important problems relevant to L remain unsolved, including the calculation of the shortest path length analytically or numerically, and how it impacts the network behaviour.

The computational complexity to calculate L of an N-node network directly is $O(N^2)$ [Dijkstra, 1959, Mohring et al., 2007], which is huge for large networks. A

lot of researchers attempted to estimate L based on the particular structure of the complex networks [Barahona and Pecora, 2002, Mao and Zhang, 2013, Gulyas et al., 2011, Cohen and Havlin, 2003]. It was suggested by a method [Barahona and Pecora, 2002] to analyze the small-world network(WS) and the random graph(RG) by rewiring a regular lattice and adjusting its adjacency matrix. [Gulyas et al., 2011] estimated the L in a network with given density by deleting edges from a complete network. [Cohen and Havlin, 2003] discovered that L of scale-free networks change dramatically given the different exponents γ of the power-law $p(k) \sim k^{-\gamma}$. When $2 < \gamma < 3$, $L \sim \ln\ln N$, and when $\gamma > 3$, $L \sim \ln N / \ln\ln N$. The researches of this kind develop a deeper understanding of the network features. However, there is no method to evaluate all kinds of networks.

Therefore, it is difficult to investigate the mechanisms how the different network topologies influence the opinion convergence. The previous works have focused on the numerical approaches. Some showed that the shorter average path length L leads to the faster opinion convergence [Gulyas et al., 2011, Arruda et al., 2013] while some obtained the opposite example [Cohen and Havlin, 2003]. A widely accepted result is that L has a negative correlation with convergence time t_c under some particular conditions. In this study, we not only repeat the classic comparison between RG, WS and SF, but consider the comparison within the classification of the SF networks. The SF concept describes a class of complex networks with power-law distribution. A large number of algorithms to generate a SF network exist. Even with the same sequence of node degree, the networks can have variety of structures, which causes

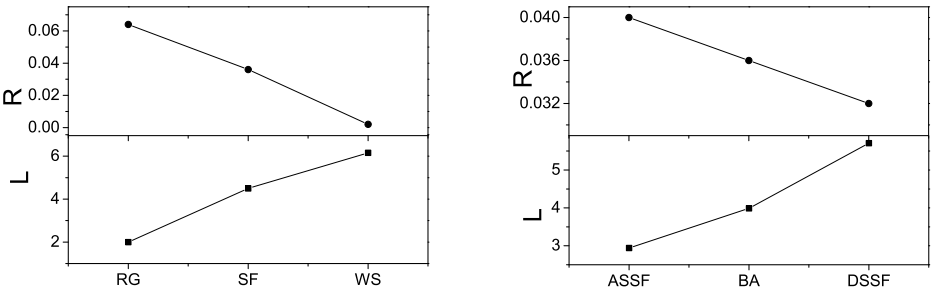


Figure 2.10: The eigenratio R and average path length L of the four 1000-node SF networks.

the difference in opinion convergence speed. In this study, we only adjust the degree correlation from high to low: $ASSF$, BA, and $DSSF$. Their Pearson coefficients are 0.3, 0, -0.1 and -0.3 respectively. In Figure 2.10, we illustrate the comparison between networks. The RG with the shortest L supports the fastest opinion convergence. The WS is in contrary and the SF stands in the middle. In the lower graph, we illustrate the eigenratio R and average path length L of the three SF networks separately. The L rises as the degree correlation goes down from $Pr = 0.3$ to $Pr = -0.3$, which we will discuss in Section 2.6.3.

Now we turn to the RG and WS. These two networks have a common parameter: the rewire probability p. Given a regular lattice, break and rewire connections with probability p. If $p = 1$, the lattice becomes a random graph. So we start from a regular lattice with each node connected to its $2k$ nearest neighbors for a total of nk edges. The Laplacian matrix is L^0, $L^0_{ii} = 2k$,$L^0_{ij} = L^0_{ji} = -1$. The eigenvalues of the matrix are

$$\lambda_l = -2k + 2\sum_{j=1}^{k} \cos(2\pi(l-1)j/N). \tag{2.21}$$

When $l = 2$,

$$\lambda_2 = 2k - 2\sum_{j=1}^{k} \cos(2\pi j/N). \tag{2.22}$$

We use Taylor series on the assumption that N is large

$$\begin{aligned}
\lambda_2 &= 2k - 2(\cos 2\pi/N + \cos 4\pi/N + \cdots \cos 2\pi(k-1)/N + \cos 2\pi k/N) && (2.23)\\
\lambda_2 &= 2k - (\sum_{n=0}^{\infty} \frac{(-1)^n}{2n!}(\frac{2\pi}{N})^{2n} + \frac{(-1)^n}{2n!}(\frac{4\pi}{N})^{2n} + \cdots + \frac{(-1)^n}{2n!}(\frac{2\pi k}{N})^{2n}), && (2.24)\\
\lambda_2 &= 2k - 2\sum_{n=0}^{\infty} \frac{(-1)^n}{2n!}(\frac{2\pi}{N})^{2n}(1^{2n} + 2^{2n} + \cdots + k^{2n}). && (2.25)
\end{aligned}$$

Consider the summation about the Taylor term. When $n = 0$, it equals $-2k$. If expanding it to $n = 1$, we get

$$\lambda_2 \approx \frac{-4\pi^2}{N^2}(1^2 + 2^2 + \cdots + k^2) = -2\pi^2 k(k+1)(2k+1)/3N^2. \tag{2.26}$$

Therefore, we have

$$\lambda_{ac} = \lambda_2 \approx -2\pi^2 k(k+1)(2k+1)/3N^2. \tag{2.27}$$

The WS or RG) network is obtained from the lattice by adding Ns new connections randomly, so that the average number of shortcut for every node is s. Then we get a random and symmetric matrix L^r to the L^0. In L^r, connections are put in the $N(N-2k-1)$ entries with probability $p = 2s/(N-2k-1)$. Each new connection between node i

and node j gives the L^0 off-diagonal $\Delta L^{ij} = \Delta L^{ji} = 1$ while on-diagonal $\Delta L^{ii} = \Delta L^{jj} = -1$. Since in our study the Laplacian matrix $M = -L$. In M, the off-diagonal $\Delta L^{ij} = \Delta L^{ji} = -1$ while on-diagonal $\Delta L^{ii} = \Delta L^{jj} = 1$. We get the new algebraic connectivity λ_{ac} [Boyce, 1968]

$$\lambda_{ac} = \lambda_{ac}^{(0)} + \varepsilon\lambda_{ac}^{(1)}, \tag{2.28}$$

and

$$\varepsilon\lambda_{ac}^{(1)} \approx -2s. \tag{2.29}$$

Obviously, the larger s is, the smaller λ_{ac} is and the greater R is. We try to keep the connection numbers in RG and WS the same, so we just remove a connection from the regular lattice every time a shortcut is added in the network (while keeping it as regular as we can). Therefore, we have

$$\lambda_{ac} = \lambda_{ac}^{(0)} + \varepsilon_1\lambda_{ac}^{(1)} - \varepsilon_2\lambda_{ac}^{(2)}, \tag{2.30}$$

where

$$\varepsilon_2\lambda_{ac}^{(2)} \approx 2s/N. \tag{2.31}$$

Obviously, the negative correlation between s and λ_{ac} remains even if the total number

of connections never changes. Therefore *RG* with a higher value of s will always support faster synchronization than WS.

However, it has not been determined analytically when the small world feature emerges as p grows from 0 to 1. The emergence of Equation (2.31) is impacted by the network size N, the degree of the nodes in the original regular network k, and the randomness parameter p.

$$L(N,k,p) \sim \frac{N}{k} f(pkN), \tag{2.32}$$

where $f(pkN)$ is an universal scaling function and $f(pkN)$ is a constant when $pkN \ll 1$ and $f(pkN) = \ln pkN/pkN$ when $pkN \gg 1$. From this result, it turns out that L begins to decrease with p, and consequently the small world behaviour emerges, for $p \geq p_{ws} = 1/Nk$. This shows that at this point the synchronizability is not enhanced by the rewiring. To achieve such an enhancement, the density of shortcuts has to be independent of N, which happens for $p \geq p_{sync} = 1/k$, that is deep in the small world regime. In other words, in the intermediate region $p_{ws} < p < p_{sync}$, L decreases while the synchronizability of the system remains roughly the same. Figures 2.11-12 illustrate the regime in the networks sized 500, 1000, 2000 respectively. In Figure 2.14, the eigenratio R increases slightly and stably. The average path length L drops markedly from $p = 0$ to $p = 0.1$, then falls off slightly from $p = 0.1$ to $p = 1$. The opinion convergence time t drops significantly from $p = 0$ to $p = 0.4$ then flats up until $p = 0.75$. It stops changing after $p = 0.75$. In Figure 2.15, the similar results have

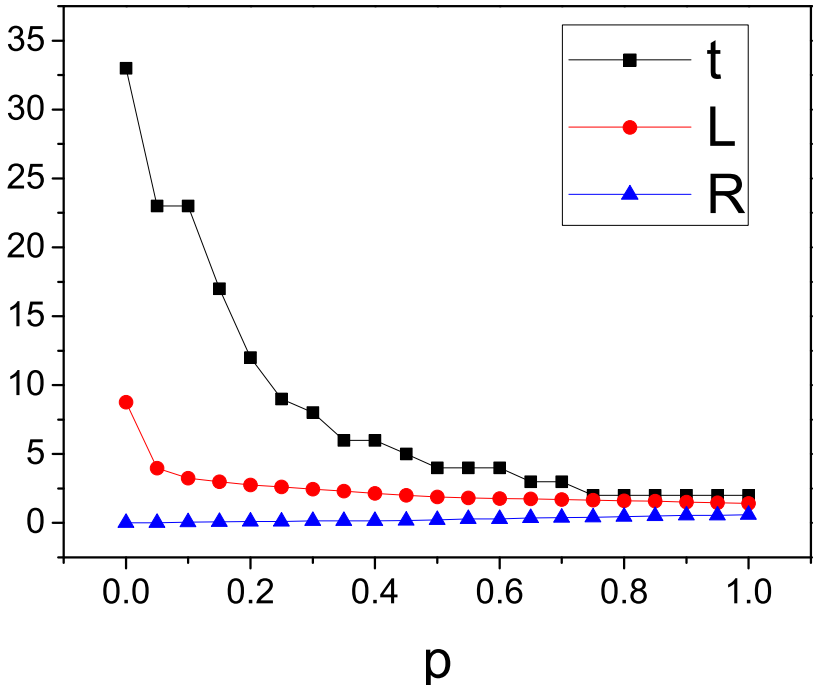


Figure 2.11: The simulation of the opinion dynamics on a group of 500-node networks from regular lattice($p = 0$) to $ER(P = 1)$.

been obtained. In both groups, the average path length L drops and the eigenratio R grows when p goes from 0 to 1. However, the changes of opinion convergence time t have stopped at around $p = 0.75$ for both groups.

It has been observed that for the three networks in different sizes the values of L reduce and R is increasing monotonically when p changes from 0 to 1. However, the opinion convergence time t stops decreasing after the p_{sync} mentioned before.

2.6.2 The clustering coefficient

The clustering coefficient(C) is a measure of the degree to which nodes in a graph tend to cluster together [Holland and Leinhardt, Watts and Strogatz, 1998]. It is not

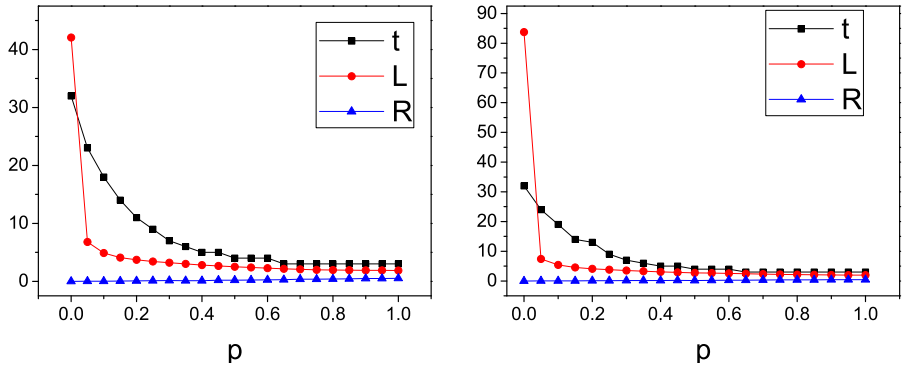


Figure 2.12: The same experiment as in Figure.2.14 in 1000-node networks and 2000 networks respectively.

possible to analyze C separately because it is strongly related to L. The high clustering coefficient indicates the existence of hubs, which makes the L shorter. Meanwhile, the large clusters causing high clustering coefficient may delay the convergence.

Previous studies [McGraw and Menzinger, 2005, Gomez-Gardenes and Moreno, 2007, Zhao et al., 2006] have shown that the opinion convergence time t increases with clustering coefficient, while the eigenratio R drops. However, we find a counter example in the comparison between the SF networks. See Figure 2.13. For R, it is obtained that $RG > SF > WS$, which indicates that RG gives the fastest opinion convergence and WS the slowest. However, the SF in the middle holds the largest clustering coefficient and the WS follows it as the second largest. The same experiments have been taken on the 2000-node and 5000-node networks respectively and the same results are found. In the lower graph, the C falls to the bottom at $Pr = 0$ before it goes up again and achieves the climax at $Pr = -0.3$.

The clustering coefficient C shows how clear the natural communities are in the

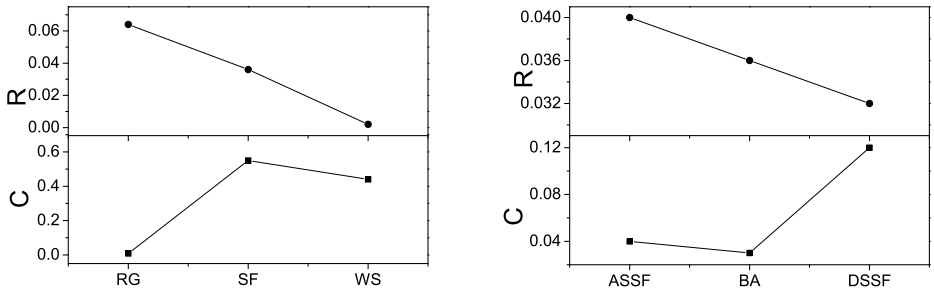


Figure 2.13: The eigenratio R and clustering coefficient C of the four 1000-node *SF* networks.

network. The *RG* is the most homogeneous network among all and has the lowest values of C. For the *SF* group, the *ASSF* has small numbers of natural communities with majority of nodes in them, and the *DSSF* has a lot of small communities. Both of them hold stronger community structure than the randomly-connected *BA*. Apparently, the C cannot be used as a single factor in the observed dependency.

2.6.3 The degree correlation

In many real-world networks, the degree of a node is often correlated with the degree of the neighboring nodes. Correlated networks show assortative (disassortative) mixing when high degree nodes are mostly attached to nodes with high (low) degree. From the last two sections, we have found that the degree correlation has significant influence to the opinion convergence time. See Figures 2.9, 2.10 and 2.13. In Figure 2.9, we can see apparent gaps between the eigenvalues of Laplacian matrices of the *SF* networks. It has been discovered [Arenas et al., 2008] that those kinds of gaps indicate natural communities in networks. As in Figure 2.14, for *ASSF*, it is highly possible to have one single large community with the high degree node as the hub, while the *DSSF*

tends to have several small communities. In Figure 2.9, the *ASSF* has the largest gap between the smallest and the second smallest eigenvalues, which indicates that it will take longer time than *DSSF* and *BA* to have the first local opinion convergence. But after that, the speed for *ASSF* to have the global convergence is faster than *DSSF* and *BA*.

For large networks, we can use the concept of edge connectivity $e(G)$ to estimate the R of *SF* networks. It is the smallest number of connections to remove from a network to make it disconnected. Take the two networks in Figure 2.14 for example, $e(G) = 1$ for the *DSSF*. If we remove one of the connections between node 1 and nodes $2, 3, 4$ or 5, an isolated community will appear. For the *DSSF* in Figure 2.14, $e(G) = 2$. If we remove the only two connections from node 6, it will become isolated. Usually, the $e(G)$ decreases with the increase of assortativity.

Given the degree sequence of a network $k_{min} = k_1 \leq k_2 \cdots \leq k_N = k_{max}$, a general approximation of the eigenvalue bounds are [Fiedler, 1973, Anderson and Morley, 1985]

$$2(1-\cos(\frac{\pi}{N}))e(G) \leq \lambda_{ac} \leq \frac{N}{N-1}k_{min}, \tag{2.33}$$

and

$$\frac{N}{N-1}k_{max} \leq \lambda_N \leq 2k_{max}. \tag{2.34}$$

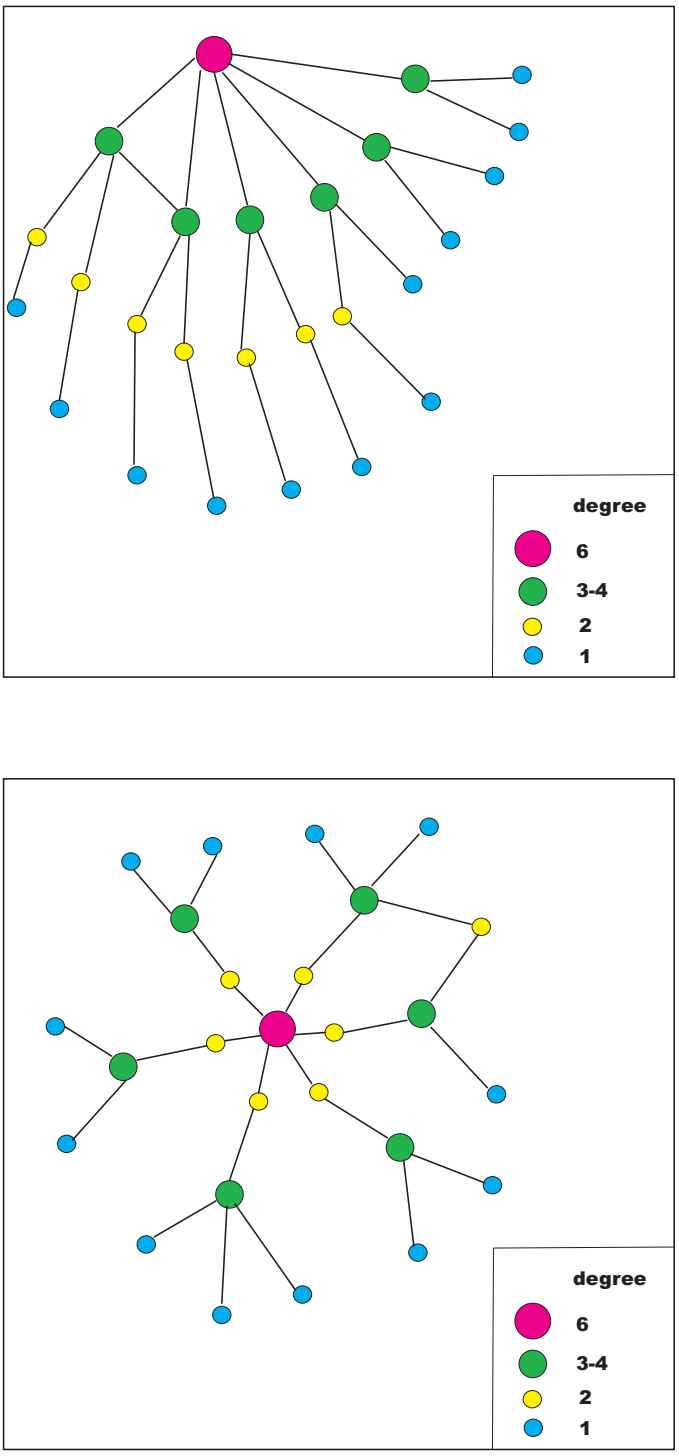


Figure 2.14: The upper graph shows the assortative connectivity, where the large degree nodes tend to connect with each other. The lower graph shows the disassortative connectivity, where large degree nodes and small degree ones mix to connect with each other. Each network consists of 26 nodes and 26 connections.

The eigenratio $R = \frac{\lambda_{ac}}{\lambda_N}$ follows

$$\frac{(1-\cos(\frac{\pi}{N}))e(G)}{k_{max}} \leq R \leq \frac{k_{min}}{k_{max}}. \tag{2.35}$$

2.7 Remarks

In this chapter, we have discussed five types of networks and simulated the opinion dynamics on the five networks. We discussed some disagreements in the previous studies and investigated the relations between network topology and opinion convergence time. We have the following remarks:

1. For most complex networks, a short average path length L indicates faster opinion convergence. However, for the group of regular network, WS and ER, the L drops from $p = 0$ to $p = 1$, the convergence time t_c goes through a rapid drop and stops changing far before $p = 1$.

2. The clustering coefficient C has no monotonic relation with t_c nor with L.

3. The Pearson coefficient Pr as the measure of degree correlation describes how the communities in the network impact the convergence.

The analysis from graph theory is provided to solve the problem of opinion convergence time. However, it is still unclear how to control the network behaviors by adding or deleting the nodes and connections. The further study on this topic will focus on the network control and the applications to real life systems.

Chapter 3

The opinion networks with a social outcast

3.1 Introduction

In the last chapter, we built a group of complex networks with the opinion dynamics [Curtis and Smith, 2008] as in Equation (2.12) and (2.13). After that, we only discussed unweighted-undirected networks where $b_i = 1$ for every node i. Since M is balanced and all off-diagonal elements are positive, the eigenvalues of M take the form $0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_N$. Therefore, the opinions will always converge on a consensus x_s. The eigenratio $R = \lambda_2/\lambda_N$ of M is used to measure the ability of the opinions to converge on a network. In this chapter, we will build and analyze the opinion dynamics on weighted-directed networks.

3.1.1 The weighted-directed networks with an outcast

It is clear that most real-world complex networks, e.g., World Wide Web and mobile communication networks, are directed networks. However, many existing tools developed for the study of system convergence in complex networks can only be applied to undirected networks. So in this chapter, we will set every node to have $b_i = 1$ random except for one of them with $b_j < 0$. The owner j is an outcast among people, and always provides negative influence to others. Now we turn to the weighted-directed networks, which was studied in both the areas of graph theory and complex networks.

We simulate the opinion evolution with an outcast on some networks. It is possible that a divergence occurs during the process, see Figure 3.1. In this group of experiments, we build a 20-node random network with 50 links. It is guaranteed that there is at least one path between every two nodes.

In this chapter, we investigate how the degree and the influence ability of the outcast affect the opinion dynamics. We will analyze how the parameters of the outcast impact the main characters of the weighted-directed networks. Further more, how the network characters impact the dynamical behaviours on them will be revealed. In this way, we develop a method to study the weighted-directed networks.

3.1.2 Research organization

The research in Chapter 3 is structured as follows: In Section 3.1, we give a introduction of the background and motivation of the research. In Section 3.2, we investigate the algebraic connectivity λ_{ac} of weighted-directed networks as a measure of the net-

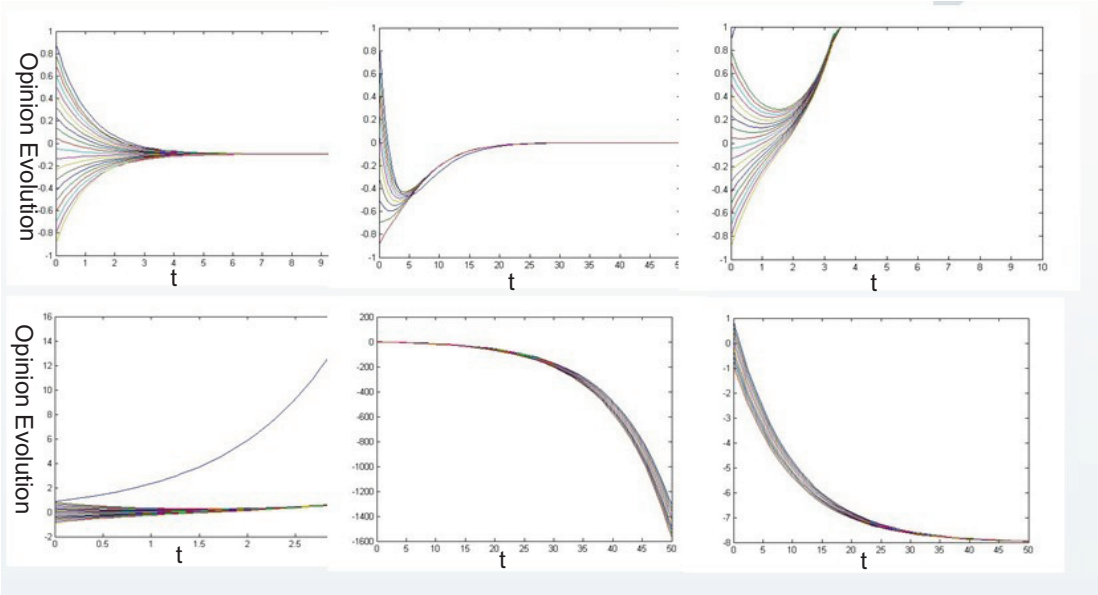


Figure 3.1: The outcast in different position of the network and with varying power may lead to several results during the system evolution.

work stability. In Section 3.3, the simulations of opinion dynamics with an outcast are described on five typical complex networks. In Section 3.4, we give an analysis based on an asymptotic method. In Section 3.5, we provide the remarks of this chapter.

3.2 The algebraic connectivity of a weighted-directed network

In the last chapter, we use the algebraic connectivity $R_{ac} = R_2$ to measure the ability for the network to support the synchronization on it. In this chapter, since neither the adjacency matrix A nor Laplacian matrix M is symmetric, it is necessary to investigate how we can use R_{ac} as follows.

3.2.1 The weighted and directed graph

A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. A weighted-directed graph is a graph where the edges are directed, i.e. each edge is an ordered pair of vertices with $i, j \in V$ denoting an edge $E_{ij} \in E$ which starts at vertex i and ends at vertex j, with a weight associated to it. In this study, we have some hypothesis as follows:

1. A self-loop means two vertices of an edge is the same. There is no self-loop in the graph to make sure the diagonal elements of the adjacency matrix is zero. Then it is guaranteed that the row sum of Laplacian matrix zero. In terms of opinion dynamics, it means that a node can not impact itself.

2. There are no multiple edges in the graph. In real-life network, people may have multiple circumstance to communicate with each other. But in this study, we only consider the influence ability. So we build single edge with weight.

3. The graph is connected. There is always a path between any two vertices. Every isolated part in a network will add one more zero eigenvalue to the Laplacian matrix. We do not put this case in consideration.

3.2.2 Algebraic connectivity of influence matrix

Most researches in this area have focused on the unweighted-undirected networks. A few studies have investigated the phenomenon on weighted-directed networks nor developed the methods for them. In Chapter 2, the eigenvalues of the influence matrix

are $0 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_N$. The algebraic connectivity $\lambda_2 = \lambda_{ac}$ [Fiedler, 1973] is the second largest eigenvalue of M next to the only zero eigenvalue, which guarantees the convergence of opinions. In Chapter 3, the participant of outcast may make the λ_2 closer to zero and sometimes larger than zero. The system dynamic can be written as

$$X = e^{Mt}X^0 = Pe^{\Lambda t}P^{-1}X^0, \tag{3.1}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$, the diagonal matrix of eigenvalues of M, and P has the corresponding eigenvectors as columns. From Equation (3.5), we can get opinion x_i of node i at time t

$$x_i^t = \sum_{j=1}^{N} P_{ij}e^{\Lambda_j t}x_j^0. \tag{3.2}$$

So, the difference of opinions between any two nodes i and m is

$$|x_i^t - x_m^t| \leq \sum_{j=1,k=1}^{N} |P_{ij} - P_{mj}|e^{\Lambda_j t}P_{jk}^{\dagger}x_k^0 \tag{3.3}$$

$$\leq \sum_{j=1}^{N} |P_{ij} - P_{mj}|e^{\Lambda_j t}. \tag{3.4}$$

Apparently, if any $\lambda_i > 0$, the difference between at least two nodes can become larger when $t \to \infty$. Therefore, the largest eigenvalue except for zero in this system remains a suitable measure of the ability to determine the convergence, even if it's positive.

We investigate the same problem in a view of graph theory. Let K be the set $\{\alpha \in$

$R^V, \alpha \perp \mathbf{1}, \|\alpha\| = 1\}$. The Fiedler's algebraic connectivity of M in Frobenius normal form can be expressed as [Wu, 2007]

$$\lambda_{ac} = \max_{\alpha \in K} \frac{\alpha^T M \alpha}{\alpha^T \alpha} \tag{3.5}$$

We know from the graph theory [Kelner, 2009, Ren, 2015] that if M_1 and M_2 are two graphs on the same vertex set with disjoint edge sets, we have $M_{M_1 \cup M_2} = M_1 + M_2$. If M_1 has eigenvalues $\lambda_1, \cdots, \lambda_n$ with eigenvectors $v_1, \cdots, v_n$, and M_2 has eigenvalues $w_1, \cdots, w_n$ with eigenvectors $u_1, \cdots, u_n$, then M has eigenvalues $\lambda_1, \cdots, \lambda_n, w_1, \cdots, w_n$ with corresponding eigenvectors $v_1, \cdots, v_n, u_1, \cdots, u_n$. For a weighted-directed network with an outcast, we can separate the influence matrix M to two parts: M_1 to represent the interactions without the outcast and M_2 to represent the influence made by the outcast. In this way, we rewrite the Equation (3.8) as

$$\lambda_{ac} = \max_{\alpha \in K} \frac{\alpha^T (M_1 + M_2) \alpha}{\alpha^T \alpha} \tag{3.6}$$

In Equation (3.9), due to the participation of M_2, the value of λ_{ac} is not guaranteed to be negative as in Chapter 2. We will illustrate this point in Section 3.4.

Since the matrix M is not a real and positive matrix, there can be real parts and image parts in the eigenvalues. Since the rotations happening in the real space provide enough information about the physical meaning of the eigenvalues [Ren, 2015], we will only consider the real part of the eigenvalues in this study.

3.3 Periodic solutions

In this section, we attempt to consider the opinion dynamics with the participation of a social outcast as a periodic process.

$$\frac{dx_i}{dt}^{j=n}_{j\neq i,j=1} = \sum_{j=1}^{N} m_{ij}(x_j - x_i) + \alpha_i(T_i - x_i), i = 1,2,\ldots,N, \tag{3.7}$$

and in vector form

$$\frac{dX}{dt} = (M-D)X + Dt, \tag{3.8}$$

where $M_{ij} = m_{ij}$, $i \neq j$, $D_{ii} = \alpha_i$ as the influence ability of the outcast, $t_i = T_i$ as the opinion of the outcast. The matrix M is not invertible with a null space spanned by $\mathbf{1}$ with the vector $\mathbf{1}$ defined by $1_i = 1$.

In the N-agent system, we set $m_{i1} = m_{1i} = -1$ and

$m_{ij} = m_{ji} = 1$ for $j = 1,2,\ldots,N$. The agent 1 is against the views of other agents and is reacted by opposing equally the view of agent 1. A period of cooperation of duration $t = 1$ was followed by a period, again of unit duration, and then the pattern repeats indefinitely. After several cycles the opinions of agents $2,\ldots,N$ converge and this group acts as a single agent, repeatedly attracting and being attracted to the opinion of agent 1, and then move away. Here we analyze this situation for a group of N agents.

At the end of the Nth cycle of cooperation and antagonism, the system is given by $X^{(N)} = G^N X^{(0)}$, where $G = e^N e^M = e^{N+M}$ in this case. As N and M commute, G has an

eigenvalue of 1, repeated twice with orthogonal eigenvectors $e_1 = (1,0,0\ldots,0)^T$, and $e_2 = (0,1,1\ldots,1)^T$. The remaining $N-2$ eigenvalues are $e^{-2(n-1)}$. The eigenvectors associated with these are not mutually orthogonal but are orthogonal to e_1 and e_2. After several cycles the components associated with these eigenvalues decay rapidly. If N is large, a periodic solution emerges at the end of each cycle $X^{(N)} = X_e = \alpha_1 e_1 + \alpha_2 e_2$ for all N. Since $e_{1,2}$ are orthogonal to the remaining eigenvectors, the values of $\alpha_{1,2}$ can be found from the projection of the initial data X^0 onto $e_{1,2}$ giving $\alpha_1 = X_1^0$, $(n-1)\alpha_2 = \sum_{i=2}^n X_i^0$. The value of α_1 is the initial opinion of agent one and α_2 is the mean initial opinion of agent 2 to N, which we write as X^0, so that $X_e = X_1^0 e_1 + X_2^0 e_2$.

We now examine the evolution of the system in the two separate stages of each cycle. The operator e^M has eigenvalue 1 with eigenvector $M_1 = 1 = (1,1,1\ldots,1)^T$ and the operator e^{-n} repeated $n-1$ times with eigenvectors $M_i, i = 2,\ldots,N$ such as $M_2 = 1 = (-1,0,\ldots,0,1)^T$, $M_3 = 1 = (-1,0,0\ldots,1,0)^T$, …, $M_N = 1 = (-1,1,0\ldots,0,0)^T$. The vectors $M_{2,3,\ldots,N}$ are not mutually orthogonal, but they are orthogonal to M_1. We write $X_e = \sum_{i=1}^N \mu_i M_i$. Taking the scalar product of this with $M_j, j = 1,\ldots,N$ in turn yields $N\mu_1 = \alpha_1 + (n-1)\alpha_2$ for $j = 1$ and $-\alpha_1 + \alpha_2 = 2\mu_j + \sum_{i=2, i\neq j}^N mu_i$. Hence we have $mu_1 = N^{-1}\sum_{i=1}^N X_i^0 = \overline{X^{(0)}}$, the mean initial opinion of all the agents. Also $Nmu_j = -\alpha_1 + \alpha_2 = -\Delta$ with $\Delta = X_{1^{(0)}} - \overline{X^{(0)\prime}}$ for all $j > 2$. These different eigenmodes develop independently so that at $t = 1$, $X = X_M = \mu_1 M_1 + e^{-N}\sum_{i=2}^N \mu_i M_i = \overline{X^{(0)}}\mathbf{1} - N^{-1}e^{-N}\Delta N_1$, where $N_1 = \sum_{i=2}^N M_i$. In terms of $e_{1,2}$, $X_M = (\overline{X^{(0)}} + N^{-1}(N-1)e^{-N}\Delta)e_1 + (\overline{X^{(0)}} + N^{-1}e^{-N}\Delta)e_2$. The difference between the two active opinions, i.e, the opinion of agent one and the common viewpoint of the remainder, varies from

a maximum of Δ at the start of a cycle, to a minimum midcycle of $e^{-N}\Delta$. The opinion of agent one ranges from $X_{1^{(0)}}$ to $\overline{X^{(0)}} + N^{-1}(N-1)e^{-N}\Delta$, exponentially to the mean of all initial opinions as N increases. The other opinion varies from $\overline{X^{(0)'}}$ to $\overline{X^{(0)}} + N^{-1}e^{-N}\Delta$ also exponentially close to the mean initial opinion.

The next part of the development is governed by the operator e^N, which has eigenvalues 1 and e^{-N+2} repeated $N-2$ times with eigenvectors $N_1 = (-(N-1), 1, \ldots, 1, 1)^T$, $N_2 = (1, 1, \ldots, 1, 1)^T$, $N_3 = (0, -1, \ldots, 0, 1)^T$,
$N_4 = (0, -1, \ldots, 1, 0)^T$, and $N_N = (0, -1, 1 \ldots, 0, 1)^T$. We start the solution in terms of these eigenvectors $X_N = \sum_{i=1}^{N} v_i N_i$, where the equations for v_i can be found by taking the scalar product of this equation with N_j. We find $v_1 = N^{-1}e^{-N}\Delta$, $v_2 = \mu_1 = \overline{X^{(0)}}$, and others all 0. Allowing the eigenmodes to develop, so that N_1 grows by a factor e^N and N_2 grows by a factor unity, gives $X = (\overline{X^{(0)}} + N^{-1}(N-1)\Delta)e_1 + (\overline{X^{(0)}} + N^{-1}\Delta)e_2 = X_e$.

More generally, if the coefficients M_{ij} are relevant for a time $\sigma\Pi$ and N_{ij} for the remainder of the period Π and $(1-\sigma)\Pi$, then a periodic solution will emerge if $G_\sigma = e^{(1-\sigma)\Pi N} e^{\sigma\Pi M}$ has two eigenvalues of magnitude unity with all other eigenvalues less than one in magnitude. There will always be at least one eigenvalue of unity, inherited from the fact that zero is an eigenvalue of both M and N. Whether there is a second depends on the values of M and N.

3.4 Simulation

3.4.1 Simulation settings

In this section, we simulate the opinion evolution on the network with an outcast. We use the five 1000-node networks we built in the last chapter, including the small world network (WS), the random graph (RG), the Barabasi-Albert model (BA), the assortative scale free network ($ASSF$) and the disassortative scale free network ($DSSF$). Each network consists of 1000 nodes and 5000 connections. In each network, we conduct the simulation for 100 rounds. The algorithm is as follows:

1. The nodes are ranked by degree in ascendency order from small to large, and divided into 10 groups. We choose a random node from the first group as the outcast i.

2. For the outcast i with the certain degree, we give it the influence ability b_i from -5 to -0.5. We add 0.5 each round.

3. After we finish the 10 rounds for the outcast, we go on to the next group and repeat the process.

3.4.2 Simulation results

We expect to observe how the position and strength of the outcast impacts the opinion evolution. The experiments results are shown in Figure 3.2-6. We choose 10 nodes with degrees from lowest to highest and make one of them the outcast every time. For each outcast with a certain degree, we give it the strength to impact others from -5

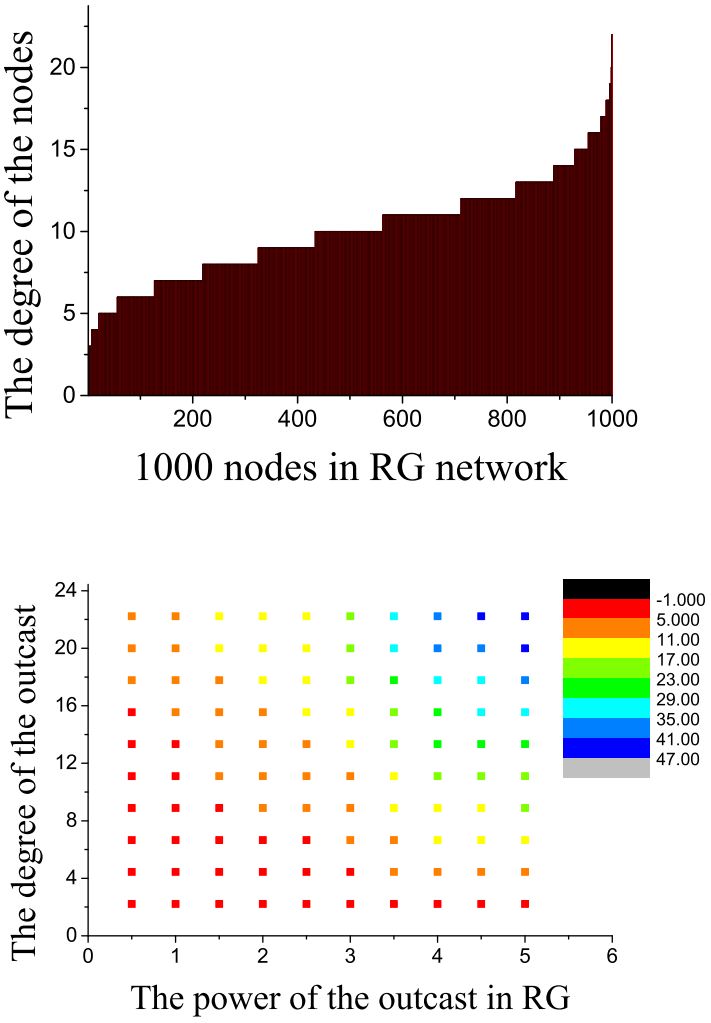


Figure 3.2: Random graph (RG) of 1000 nodes. The change of λ_{ac} when adjusting the degree and strength of the outcast. The blocks in red illustrate negative λ_{ac}, otherwise, positive λ_{ac}.

to -0.5 and we record the algebraic connectivity λ_{ac} in each round. In Figure 3.2, when the degree is really low, the increasing strength doesn't change the convergence of opinions. When the degree goes up to 16, even a very small outcast strength can cause divergence.

In Figure 3.3, we break and rewire a regular graph with probability p. We stop at

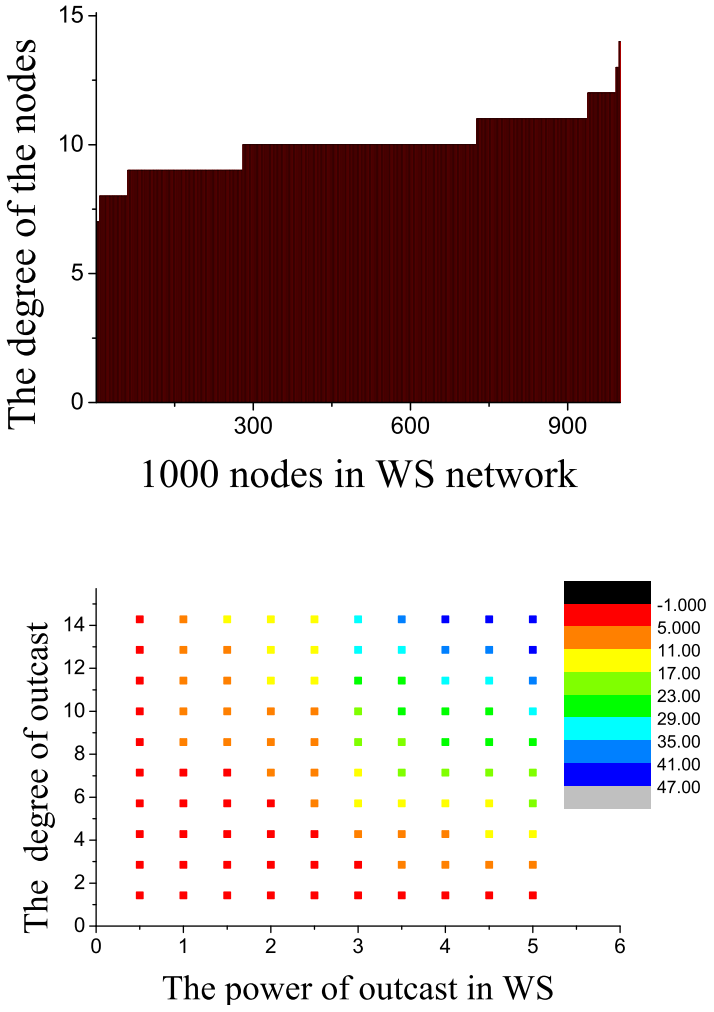


Figure 3.3: The degrees and the change of λ_{ac} in WS network.

$p \approx 0.1$ and consider it a WS network. The WS in between has degrees less heterogeneous than RG. The largest degree in the WS is only 14. No matter where we put the outcast, it will cost higher strength to cause the divergence than in RG.

In Figure 3.4, we can see that the BA has much larger range of degrees than RG and WS. The degrees are generated by power-law $p(k) = k^{-\lambda}$ where $\lambda = 3$. The lowest degree is 4 and the highest is around 160. A node with degree 16 is the hub in RG but

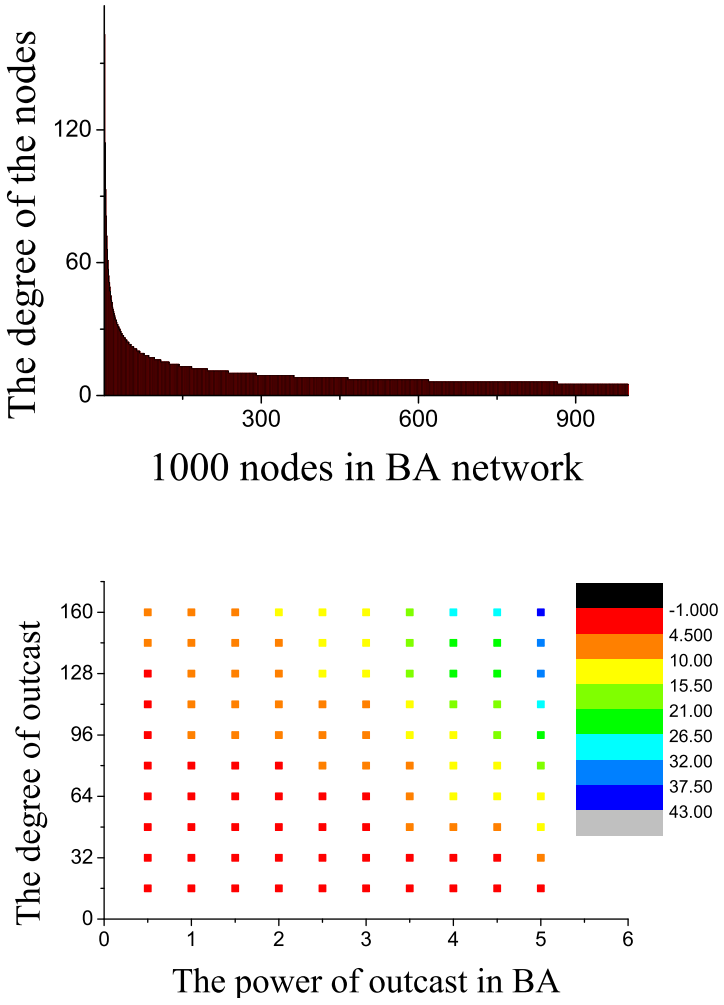


Figure 3.4: The degrees and the change of λ_{ac} in *BA* network.

only a branch around a hub in *BA*. It is shown that there are three levels of nodes in the *BA*. The 'branch' with degree lower than 32 can hardly impact the convergence as an outcast. The 'hub' with degree between 32 and 80, whose ability to make the system diverge, depends on its strength as an outcast. The 'critical hub' with degree higher than 80 can easily perturb the system with an extremely low strength.

In Figure 3.5, the *ASSF* is generated from *BA* by adjusting the Pierson coefficient

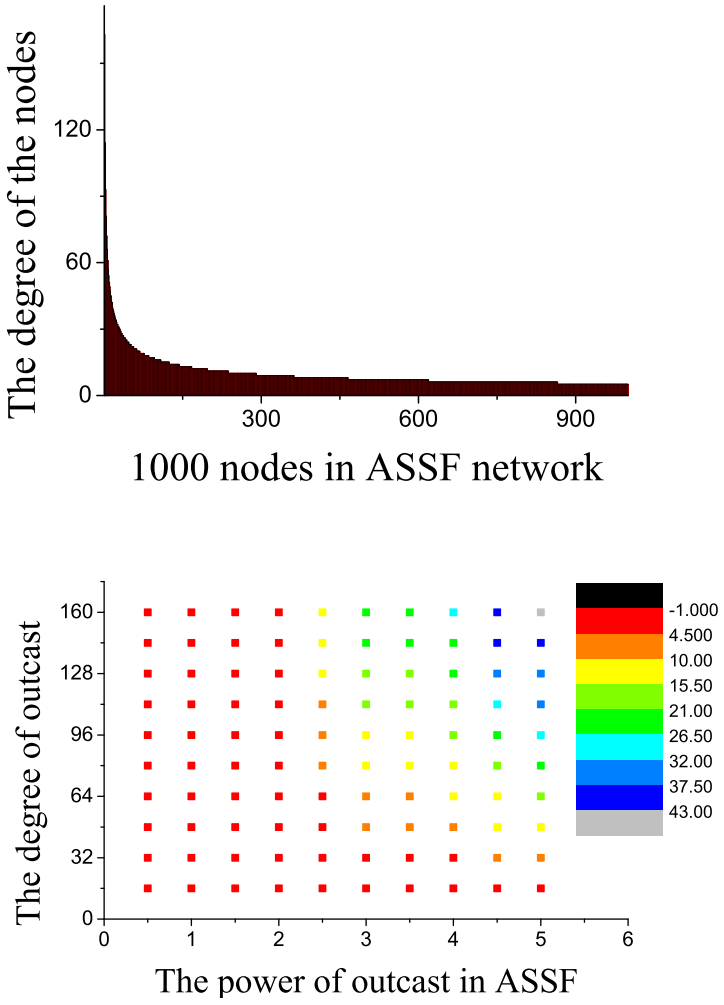


Figure 3.5: The degrees and the change of λ_{ac} in *ASSF* network.

Pr from -0.1 to 0.3. A significant improvement of network stability can be observed in *ASSF*. The 'hubs' need higher strength to change others' opinions. Unless the strength of those 'critical hubs' is higher than 2, the system remains converging. However, when the strength of outcast with degree 160 achieves 10, it drags the λ_{ac} further from zero than in *BA*.

In Figure 3.6, the *DSSF* is generated from *BA* by adjusting the Pierson coefficient

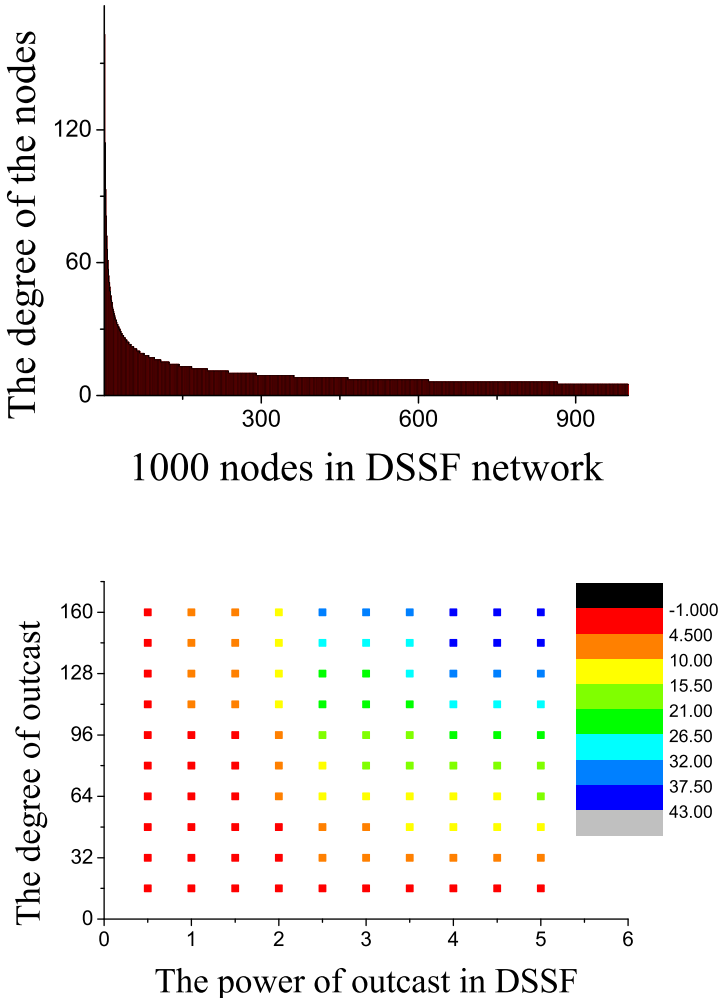


Figure 3.6: The degrees and the change of λ_{ac} in *DSSF* network.

Pr from -0.1 to -0.3. Compared with *BA*, it is easier for the 'branches' in *DSSF* to perturb the convergence, while more difficult for the 'hubs' and 'critical hubs' to do that. When both the strength and degree of the outcast are high, the λ_{ac} flats up around 37.

3.4.3 Simulation discussion

In Figure 3.2, we illustrate how the perturbation from the outcast changes on *RG*. We have concluded that *RG* has significantly shorter average path length L and smaller clustering coefficient C than other networks. The Pearson coefficient to represent the degree correlation is close to 0. In *RG*, the high randomness and short L make it convenient for every two nodes to reach each other, including the outcast. When the degree of the outcast grows, the *RG* shows vulnerability and the λ_{ac} is driven away from 0 rapidly.

In WS, the obvious clustering structure delays the exchange of opinions, see Figure 3.3. From Equation (3.3) we have the solution in terms of the normal modes

$$x_i^t = x_i^0 e^{\lambda_i t}, i = 1, 2 \ldots N, \tag{3.9}$$

which is satisfied at time t. If we rank the system of equations in descending order of the eigenvalues, when all the eigenvalue λ_i of M are negative, the right hand side of Equation (3.9) will achieve zero in a hierarchical way. In other words, a small cluster achieves a local consensus before it communicates with other clusters. If the clustering structure is not clear in the network, the time spent between hierarchies will be shortened. However, when there is one $\lambda_i > 0$, it will impact those who are in the same cluster first. The WS with clearer clusters can limit the damage of the outcast within a cluster when a strong local consensus is formatted in other clusters. It enhances the stability against $\lambda_i > 0$.

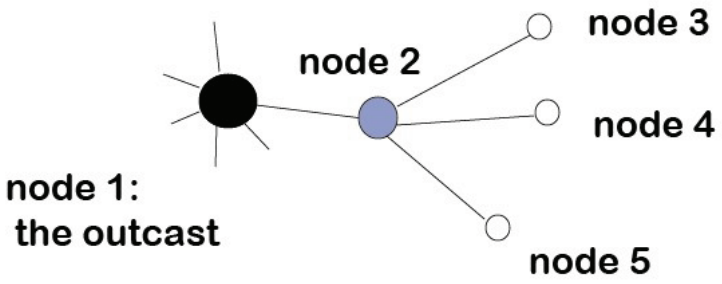


Figure 3.7: The node 1 is the outcast with the highest degree. The node 2 is one of the outcast's neighbours, with degree lower than 1 but still high in the network. The nodes $3-5$ are three single degree branches connected to 2.

Now we turn to the three *SF* networks. From Chapter 2, we have already known that, for the average path length L, $ASSF > BA > DSSF$. For the clustering coefficient C, $DSSF > ASSF > BA$. For degree correlation, $ASSF > BA > DSSF$. In Figures 3.4-6, the *ASSF* shows the highest stability when the outcast strength is low. No significant difference has been observed between *BA* and *DSSF*. But when the strength and degree are both high, the *ASSF* shows the weakest stability and its λ_{ac} soon gets away from zero. In *ASSF*, when the outcast owns the highest degree, it will impact directly the nodes with less-high degrees, like in Figure 3.7.

For node 1, the evolving equation is

$$\frac{dx_1}{dt} = m_{12}(x_2 - x_1) + \cdots . \tag{3.10}$$

For node 2, it is

$$\frac{dx_2}{dt} = \frac{m_{21}}{\varepsilon}(x_1 - x_2) + \sum_{i=3}^{5} m_{2i}(x_i - x_2). \tag{3.11}$$

For nodes $3-5$, it is

$$\frac{dx_i}{dt} = m_{i2}(x_2 - x_i), i = 3,4,5. \tag{3.12}$$

Since the outcast is impacted by a lot of neighbours, we cannot predict how it changes its opinion. When $\frac{m_{21}}{\varepsilon} < 0$ is high enough in value, the difference between node 1 and node 2 is enlarged rapidly and constantly. Meanwhile, it will be difficult for the nodes $3-5$ to converge with node 2. The damage from the outcast spread through the high degree nodes like node 2. Not only the outcast can not converge with it's neighbours, the other parts of the network can hardly achieve a local consensus as well. But in *DSSF*, the outcast with high degree can only impact the small degree branches, while some clusters far away from the outcast may converge and impact the other nodes as a strong group. Therefore, the damage from the outcast is reduced. We will discuss this condition based on asymptotic method in the next section.

3.5 Small α

If t is $\mathbf{O}(1)$ and the α_i are small, then we write $\alpha_i = \varepsilon\overline{\alpha_i}$, $D = \varepsilon\overline{D}$, $\overline{D_{ii}} = \overline{\alpha_i}$, $X = X_0(t) + \varepsilon X_1(t) + \cdots$. By examining the limit $\varepsilon \to 0$, we find

$$\frac{dX_0}{dt} = MX_0, \tag{3.13}$$

and

$$\frac{dX_1}{dt} = MX_1 - \overline{D}X_0 + \overline{D}t. \tag{3.14}$$

The solution to Equation (3.16) is $X_{0^t} = e^{Mt}X(0)$. The eigenvalues of M are all negative and we assume that they are distinct with the exception of the zero eigenvalue corresponding to the null space of M. Let $P = (M_1, M_2, \cdots, M_N)$ be a matrix of eigenvectors m_i of M so that $Mmi = \lambda mi$, $\lambda_i < \lambda_j$ if $i < j$ and $\lambda_N = 0$, $m_n = \mathbf{1}m$ where the value of m depends only on the normalization of the eigenvectors and could be replaced by unity later on. The $X_{0^t} = Pe^{\widehat{D}t}P^{-1}X(0)$ where $\widehat{D}$ is diagonal and so also is its exponential with $e^{\widehat{D}t}_{ii} = e^{\lambda_i t}$. Since all but the largest eigenvalue is negative, this exponential approaches the matrix Q with $Q_{ij} = 0$ with the exception $Q_{NN} = 1$. Hence at least $t \to \infty$, and

$$X_{0^t} \sim \mathbf{1}E_0, \tag{3.15}$$

and

$$E_0 = mw \cdot X(0) = w \cdot X(0)/w \cdot \mathbf{1}, \tag{3.16}$$

and

$$w_i = P_{Ni}^{-1}, \tag{3.17}$$

where the $\cdot$ represents the usual dot product. Note that since $P^{-1}P = I$, the identity matrix, $w \cdot m_i = 0$, $i = 1, 2, \cdots, N-1$ and $w \cdot \mathbf{1}m = 1$. The next order term X_1 satisfies Equation (3.16) and the solution with $X_1(0) = 0$ is

$$X_1(t) = P \int_0^t e^{\widehat{D}(t-\tau)} d\tau P^{-1} \overline{D} t - P \int_0^t e^{\widehat{D}(t-\tau)} P^{-1} \overline{D} P e^{\widehat{D}\tau} d\tau P^{-1} X(0). \tag{3.18}$$

Analysis of this as $t \to \infty$ is possible and is helped by the fact that the exponential terms all decal since $\lambda_i < 0$, but the details become complicated. However, it is clear that $t \to \infty$,

$$X_1(t) = P(S_d + Qt)P^{-1}\overline{D}t - P(U_d + Q(w \cdot \overline{\alpha})t)P^{-1}X(0), \tag{3.19}$$

and hence,

$$X \sim \mathbf{1}(E_0 + \varepsilon t E_1) + \varepsilon v + \cdots, \tag{3.20}$$

where $E_1 = (w \cdot \overline{D})t - (w \cdot \alpha)E_0)/(w\cdot)\mathbf{1})$ and $v = PS_dP^{-1}\overline{D}t - PU_dP^{-1}X(0)$.

Over the long time scale $\mathbf{O}(\varepsilon^{-1})$, the weak influence of α_i has time to influence the solution to $\mathbf{O}(1)$. We write $t = T/\varepsilon$ and $X(t) = \widehat{X_0}(T) + \varepsilon\widehat{X_1}(T) + \cdots$ to find

$$M\widehat{X_0} = 0, \tag{3.21}$$

and

$$M\widehat{X_1} = -\frac{d\widehat{X_0}}{dt} - \overline{D}\widehat{X_0} + \overline{D}t. \tag{3.22}$$

Hence, $\widehat{X_0}(T)$ is part of the null space of M giving $\widehat{X_0}(T) = \mathbf{1}L(T)$ for some function $L(T)$ which we must find

$$M\widehat{X_1} = -\mathbf{1}L' - \overline{\alpha}L + \overline{D}t. \tag{3.23}$$

A solution to this equation will only be possible if the right hand side is normal to the solution $w^\star$ of $M^\star w^\star = 0$ where $M^\star$ is the adjoint of M. Hence, $L(T)$ develops according to the equation

$$(\mathbf{1} \cdot w^\star)L + (\overline{\alpha} \cdot w^\star)L = w^\star \cdot \overline{D}t. \tag{3.24}$$

The two vectors w and $w^\star$ are in fact proportional. From the definition of $M^\star$, $(Mp)\cdot q = p\cdot(M^\star q)$ for any vectors p and q. If $q = w^\star$, then $w^\star \cdot Mp = 0$ for any p. If p is an eigenvector of M, m_i, $i = 1,2,\cdots,N-1$, then this implies $w^\star \cdot m_i = 0$, as is also true for the vector w. Hence both w and $w^\star$ are N-dimensional vectors normal to the same $N-1$ vectors. Hence they must be proportional. Alternatively, the eigenvectors of M form a complete, but not necessarily orthogonal basis. Writing any vector $q = \sum_{i=1}^{N} q_i m_i$, we have $w\cdot q = q_n m(w\cdot\mathbf{1})$. Similarity $w^\star\cdot q = q_n m(w^\star\cdot\mathbf{1})$. Hence, $(w\cdot q)/(w\cdot\mathbf{1}) = (w^\star\cdot q)/(w^\star\cdot\mathbf{1})$ and in the expressions for E_0 and E_1 above, w may be replaced by $w^\star$

Define $c = (\overline{\alpha}\cdot w^\star)/(\mathbf{1}\cdot w^\star)$ and $L_\infty = w^\star\cdot(\overline{D}t)/(\overline{\alpha}\cdot w^\star)$, we have

$$L_T = L_\infty(1-e^{-cT}) + L_0 e^{-cT}. \tag{3.25}$$

Matching with the inner solution, i.e, that obtained for $t = \mathbf{O}(1)$, gives $L_0 = E_0$. The match with terms proportional to t is automatic.

It is possible to solve the next order terms $\widehat{x}(T)$, matching with the inner region through the condition $\widehat{x}(0) = v$. However, we do not know v explicitly and there is an unknown contribution $L_1(T)\mathbf{1}$ which must be found at next order.

The solution has the following structure. Initially, for $t = \mathbf{O}(1)$, the agents interact entirely through mutual influence and come to an opinion which, to first order ε, is uniform at E_0, a weighted average of the initial distributions $x_i(0)$ and independent of α_i or T_i. Next, over the longer ε^{-1} timescale, the opinions adjust again as one to first

order to the ultimate value L_∞ which is a weighted average of the values T_i.

Note that if M is symmetric which would be the case if the influence of agent A on agent B was the same as that of B on A, then eigenvectors m_i would be orthogonal and $P^{-1} = P^T$. Hence $w = m\mathbf{1}$. In this case, the expressions found above simplify so that $nE_0 = \sum_{i=1}^N x_i(0)$, $nc = \sum_{i=1}^N \overline{\alpha_i}$, $L_\infty = \sum_{i=1}^N \overline{\alpha_i} T_i / \sum_{i=1}^N \overline{\alpha_i}$.

3.6 Large α

Equations (3.12) and (3.13) describe how the opinions of the outcast and the other nodes evolve. We assume the outcast is node 1 and its strength is large enough. If $t = \varepsilon\tau$, we expand

$$x_1 = x_1^0(\tau) + \varepsilon x_1^0(\tau), \tag{3.26}$$

and

$$x_i = x_i^0(\tau) + \varepsilon x_i^0(\tau), i = 2, 3, \cdots, N. \tag{3.27}$$

Since

$$\frac{dx_1^0}{dt} = 0, \tag{3.28}$$

we know

$$x_1^0(\tau) = x_1^0(0), \tag{3.29}$$

and

$$\frac{dx_1^1}{d\tau} = \sum_{j=2}^{N} m_{1j}(x_j^0 - x_1^0(0)), \tag{3.30}$$

$$\frac{dx_i^0}{d\tau} = \frac{m_{i1}}{\varepsilon}(x_1^0(0) - x_i^0), \tag{3.31}$$

$$\frac{dx_i^1}{d\tau} = \frac{m_{i1}}{\varepsilon}(x_1^1(0) - x_i^1) + \sum_{j=2, j\neq i}^{N} m_{ij}(x_j^0 - x_i^0). \tag{3.32}$$

So

$$x_i^0(\tau) = x_1^0(0) + (x_i^0(0) - x_1^0(0))exp(-\frac{m_{i1}}{\varepsilon}\tau), \tag{3.33}$$

$$x_1^1(\tau) = \sum_{j=2, j\neq i}^{N} [(1 - exp(-\frac{m_{j1}}{\varepsilon}\tau))\frac{\varepsilon m_{1j}}{m_{j1}}(x_j^0(0) - x_1^0(0)]. \tag{3.34}$$

So we have

$$\frac{dx_i^1}{d\tau} + \frac{m_{i1}}{\varepsilon} x_i^1 = \frac{m_{i1}}{\varepsilon} \sum_{j=2, j\neq i}^{N} [(1 - exp(-\frac{m_{j1}}{\varepsilon}\tau)) \frac{\varepsilon m_{1j}}{m_{j1}} (x_j^0(0) - x_1^0(0)] +$$

$$\sum_{j=2, j\neq i}^{N} m_{ij} [((x_j^0(0) - x_1^0(0)) exp(-\frac{m_{j1}}{\varepsilon}\tau) - ((x_i^0(0) - x_1^0(0)) exp(-\frac{m_{i1}}{\varepsilon}\tau)], \quad (3.35)$$

which leads to

$$x_i^1 \sim \sum_{j=2, j\neq i}^{N} \frac{\varepsilon m_{1j}}{m_{j1}} (x_j^0(0) - x_1^0(0)) + \frac{\varepsilon m_{1i}}{m_{i1}} (x_i^0(0) - x_1^0(0)), \quad (3.36)$$

when $\tau \to \infty$. Meanwhile,

$$x_1^1 \sim x_1^0 + \varepsilon \sum_{j=2}^{N} [\frac{\varepsilon m_{1j}}{m_{j1}} (x_j^0 - x_1^0)]. \quad (3.37)$$

We may divide all the nodes into three kinds and conclude that:

1. The outcast may be impacted by more than one neighbour with initial opinions larger or smaller than its own opinion x_1^0. So x_1^0 will get smaller or larger constantly.

2. The neighbours of outcast, like node 2 in Figure.3.7, are the key nodes to decide whether the system will converge or not. If the second item of Equation (3.21) contributes more to the *i*th opinion, the outcast may be attracted to the main group. Otherwise, the neighbour's opinion will move against the outcast's. The difference between the opinions will become larger.

3. Those who don't connect to the outcast directly, like the nodeS $3-5$ in Figure 3.7, are only influenced by node 2. If node 2 tends to converge with the outcast, all of the nodes will converge. Otherwise, nodes like nodes $3-5$ will try to approach node 2. It will look like the outcast pushes all the other nodes away. However, it's not guaranteed that node 2 and nodes $3-5$ will have a local consensus.

3.7 Remarks

In this chapter, we have observed the evolution of opinions on networks under the perturbation of an social outcast. We use the five typical networks from the last chapter. In each network, we set a random node as the outcast who gives others negative influence. In terms of graph, we make the weight of the connections from the outcast negative, while all other connections have positive weights. The networks become weighted-directed and asymmetric. The convergence of opinions is not guaranteed in this kind of networks. The eigenratio R to measure the convergence speed in Chapter 2 is no longer proper here. We have investigated the algebraic connectivity λ_{ac} as the measure of stability against the outcast. We have simulated the opinion evolutions on networks with an outcast. During the simulation, we chose nodes with different degrees as the outcast and varied the power of the outcast. The increasing instability of the network has been observed when the degree and the power of the outcast were enhanced. Then, we compared between networks to see the different instability in the five networks caused by the outcast with same degree and power. We discussed how

the structural characteristics of the networks impact the results. In the last part of the chapter, we have used the asymptotic method to deduce the process of the opinion evolution. We have analyzed how the topology impacts the evolution which is consistent with the simulation results. In this way, we explored a method to study the relation between topology and dynamical behaviour in weighted-directed networks. However, we can only relate the opinion evolution with the local structure of the outcast and its neighbours. How the global topology impacts the evolution is still unknown. In the further study, we are going to investigate how the statistical structural characteristics like the average path length(L) work during the opinion dynamics.

Chapter 4

Synchronization control on weighted-directed networks

4.1 Introduction

In Chapters 2 and 3, we investigated the opinion dynamics on complex networks governed by linear equations. In this chapter, we will literally put the opinion dynamics in terms of synchronization and investigate the control of synchronization on weigthted-directed networks. With the development of Internet, the human society is entering an era of networks. People rely on different kinds of networks increasingly, for instance, the transport networks, and the electronic networks. Most of the networks are complex in structures. Researchers in different areas study the effectiveness and stability of the networks from different angles. This thesis is aimed to propose some general methodologies to be used in different applications.

It has been discussed in Chapter 2 that the control of a network can not be achieved

by adjusting one or several structure features due to the complexity. In this chapter, we will investigate a new method to control the network globally by locally controlling some key nodes . We call these key nodes the 'controllers' of the network. Eventually, the 'controllers' are able to attract other nodes to synchronize. This method is called 'traction control'.

In a real world network with a huge number of nodes and complex structure, it is worth finding out the number and positions of the controllers. Previous studies were focused on the unweighted-undirected networks. In the following, we study the synchronization control on weighted-directed networks.

Chapter 4 is organized as follows: In Section 4.1, we introduce the motivation and target of the study. In Section 4.2, we give a brief review of the development of synchronization and synchronization control. In Section 4.3, we introduce the traction control on complex networks. In Section 4.4, we study the process of traction control. Consequently, the method to select controllers is proposed. The Master Stability Function is used to test the stability of the network system. In Section 4.5, we simulate the traction control on networks. The remarks are given in Section 4.6.

4.2 Development of synchronization and synchronization control

In 1673, Huygens [Huygens, 1673] first introduced the concept of synchronization by describing the synchronization between two coupled pendulums. Pikovsky [Pikovsky

et al., 2001] and Boccaletti [Boccaletti, 2008] provided detailed discussions recently.

In 1920s, a group of numerical methods were investigated in the development of electrical and radio wave propagations. Some researches were focused on the limit cycle in self-excited dynamical systems [Van der Pol, 1927].

In 1990s, the attractors to control a flow of dynamical systems were considered. Carroll and Pecora [Pecora and Carroll, 1990] presented the synchronized circuits for chaos. Since then, a lot of attentions have been put to the corresponding control methods and the synchronization of two dynamical systems with constraints.

Since 1950, chaos synchronization has been a topic of great attention[Stocker, 1950, Hayashi, 1964, Jackson, 1991, Pecora and Carroll, 1990, Carroll and Pecora, 1991]. In 1992, Pyragas [Pyragas, 1992] presented two methods for chaos control with a small time continuous perturbation, which can achieve a synchronization of two chaotic dynamical systems. In 1994, Kapitaniak [Kapitaniak et al., 1994] used such a continuous control to present the synchronization of two chaotic systems. In 1999, Yang and Chua [Yang and Chua, 1999] used linear transformations to investigate generalized synchronization. In 2006, Chen [Chen et al., 2006] gave a review on stability of synchronized dynamics and pattern formation in coupled systems. The dynamics and synchronization of coupled systems was also investigated via control schemes (e.g. [Yamapi et al., 2007]).

Synchronization means two or more systems sharing a same periodic frequency [Pecora et al., 1997, Boccaletti et al., 2006, Rosenblum et al., 1996]. So far, the study focuses on four classes of synchronization of two or more dynamical systems:

1. identical or complete synchronization,

2. generalized synchronization,

3. phase synchronization,

4. anticipated and lag synchronization and amplitude envelope synchronization.

In this study, we will focus on the identical synchronization and phase synchronization.

Identical synchronization Assume that $x_i(t)(i = 1,2,\cdots N)$ is a solution of the complex dynamic network

$$\frac{dx_i}{dt} = f(x_i) + g_i(x_1, x_2, \cdots x_N), i = 1,2,\cdots N, \tag{4.1}$$

where $f(x_i)$ is the self-motivating function of i and g_i is the amount that i is affected by others in the system. f and g_i are all continuous and differentiable. $g(x_1, x_2, \cdots x_N) = 0$. If there exists

$$\lim_{(t\to\infty)} \|x_i(t) - s(t)\| = 0, i = 1,2,\cdots N, \tag{4.2}$$

where $s(t)$ is a solution of system, then an identical synchronization can occur in the system with the profile for X as $x_1 = x_2 = \cdots = x_N$.

Phase synchronization If two oscillators' phase x_1 and x_2 evolve by a constant ratio $m : n$, which means $|mx_1 - nx_2| \leq \varepsilon$, where ε is a small and positive constant,

while the amplitudes can be quite different, then we claim the two oscillators have a phase synchronization. It is common in the real world, such that the numbers of rabbits and wolves in the same area may always evolve together with a constant ratio.

4.3 Synchronization by traction control on complex networks

Consider the *i*th node in a network, which is governed by the equation as follows

$$fracdx_idt = f(x_i(t),t) + \sum_{j=1,j\neq i}^{N} g_{ij}H(x_j(t) - x_i(t)), i = 1,2,\cdots N, \tag{4.3}$$

where $x_i(t)$ is the state of the node i and f indicates the dynamical feature of each node individually, which is continuous and differentiable. H is the coupling function within the network and g_{ij} is the coupling strength between node i and j. Distinguished from the l_{ij} in Chapter 2, $g_{ij} > 0$ if there is a connection between i and j, and $g_{ij} = 0$ otherwise. The diagonal elements are defined as

$$g_{ii} = -\sum_{j=1,j\neq i}^{N} g_{ij}. \tag{4.4}$$

Therefore, we have $\sum_{j=1}^{N} g_{ij} = 0$ and Equation (4.1) is equivalent to

$$\frac{dx_i}{dt} = f(x_i(t),t) + \sum_{j=1}^{N} g_{ij} H x_j(t), i = 1,2,\cdots N. \tag{4.5}$$

If there is any isolated node in the system, the solution of this node will be

$$\frac{ds}{dt} = f(s(t),t), \tag{4.6}$$

where $s(t)$ can be a periodic track or a balanced point.

If we have the ability to control every node, the system in Equation (4.3) will synchronize. However, in a complex network of huge size, it is almost impossible to manipulate the nodes individually. Thus, we design a control process, which we call 'traction control' to control a small number of nodes and attract the whole network to synchronize.

We select δN number of nodes as the 'controllers', where $0 < \delta \leq 1$. The 'controllers' are governed by

$$\frac{dx_i}{dt} = f(x_i(t),t) + \sum_{j=1}^{N} g_{ij} H x_j(t) + v_i(x_1(t),x_2(t),\cdots,x_N(t)), i = 1,2,\cdots N, \tag{4.7}$$

where we control the nodes by $v_i(x_1(t),x_2(t),\cdots,x_N(t))$.

Define the error at each node as

$$e_i(t) = x_i(t) - s(t), 1 \leq i \leq N. \tag{4.8}$$

We have the error function of the system

$$\frac{de_i}{dt} = f(x_i,t) - f(s,t) + \sum_{j=1}^{N} g_{ij} H e_j + v_i(x_1, x_2, \cdots, x_N), \quad 1 \leq i \leq N \tag{4.9}$$

To make the system synchronize, the error of the system should be

$$\lim_{t\to\infty} \|e_i(t)\|_2 = 0, 1 \leq i \leq N. \tag{4.10}$$

In this way, we can focus on the 'controllers' instead of the whole network to synchronize the system.

4.4 The stability of synchronization by traction control

In this section, we will design a network system with 'controllers'. The master stability function is used to test the stability of the system, both locally and globally.

4.4.1 The master stability function

Louis M. Pecora and Thomas L. Carroll [Pecora and Carroll, 1990] provided an important method to estimate the stability after a perturbation to a dynamical system based on the Lyapunov theorem. Given the following hypothesis:

1. The coupled oscillators (nodes) are all identical.

2. The same function of the components from each oscillator is used to couple to other oscillators.

3. The nodes are coupled in an arbitrary fashion which is well approximated near the synchronous state by a linear operator.

Given an N-node network, set x_i as the m-dimensional vector of dynamical variables of the ith node and the isolated (uncoupled) dynamics for each node is $F(x_i)$. Let H be an arbitrary function of each node variables that is used in the coupling. The dynamics of the ith node is

$$\dot{x_i} = F(x_i) + \sigma \Sigma_j G_{ij} H(x_j), \tag{4.11}$$

where σ is the coupling strength and G_{ij} is the connection between nodes in mean-field. Since we will study the nonmean-field condition later, we won't consider σ in the model. Instead, we will generate the influence ability of each pairwise nodes in G_{ij}.

The $N-1$ constraints $x_1 = x_2 = \cdots = x_N$ define the synchronization manifold.

Give this equation in matrix form

$$\dot{x} = F(x) + \sigma G \bigotimes H, \tag{4.12}$$

where $\bigotimes$ is the direct product. Let ξ_i be the variations on the ith node and the collection of variations is $\xi = \xi_1, \xi_2, \cdots, \xi_N$. Then,

$$\dot{\xi} = [1_N \bigotimes DF + \sigma G \bigotimes DH]\xi, \tag{4.13}$$

where DF and DH are Jacobian matrices of F and H. The first term in the above equation is block diagonal with $m * m$ blocks. The second term can be treated by diagonalizing G. The transformation for this does not affect the first term since it acts only on the matrix $\mathbf{1_N}$. This leaves us with a block diagonalized variational equation with each block having the form

$$\dot{\xi}_k = [DF + \sigma\gamma_k DH]\xi_k, \tag{4.14}$$

where γ_k is an eigenvalue of G. We can think of these as transverse modes and refer to them as such. This leads us to the following formulation of the master stability equation and the associated master stability function. We calculate the maximum Floquet or Lyapunov exponent $\bigwedge_{max}$ for the generic variational equation as

$$\dot{\zeta}_k = [DF + (\alpha + i\beta)DH]\zeta_k, \tag{4.15}$$

which yields the stability function $\bigwedge_{max}$ as a surface over the complex plane. Hence, we have a master stability function. The matrices DF and DH are Jacobian matrices for $F(x)$ and H. If all of the eigenmodes are stable, then the synchronous state is stable at that coupling strength. If $\bigwedge_{max}$ is positive, the coupling may be too strong and the synchronous state will not be stable or a large imaginary coupling can destabilize the system.

This method will be used to check whether the opinion system may come back to stability with a consensus after a perturbed bifurcation.

4.4.2 The local stability of synchronization

First of all we introduce some necessary hypothesis and lemma.

Hypothesis 1 Suppose that there is a boundary for $\|Df(s)\|_2$, which means, there is a non-negative value α to make $\|Df(s)\|_2 \leq \alpha$. $Df(s)$ is the Jacobian matrix when $x = s$.

Lemma 1 Define $\overline{A} = \begin{pmatrix} A_1 & A_3 \\ A_3^T & A_2 \end{pmatrix}, \overline{B} = \begin{pmatrix} B_1 & 0 \\ 0 & 0 \end{pmatrix}$.
$\overline{A}, \overline{B} \in R^{N*N}$, $\overline{A_1}$, $\overline{B_1} \in R^{r*r} (1 \leq r \leq N)$. $B_1 = diag(b_1, b_2, \cdots, b_r)$, $b_i > 0$, $i = 1, 2, \cdots, r$. $A_1^T = A_1$. $A_2^T = A_2$.

When b_i is large enough, $\overline{A} - \overline{B}$ is equivalent to $A_2 < 0$.

Suppose H, the coupling matrix of the network, is positive-definite, i.e, $\|H\|_2 = \gamma > 0$, the smallest eigenvalue of $(H + H^T)/2$ is ρ_{min}. The largest eigenvalue of $\overline{G_i}$ is λ_i, where $\overline{G_i}$ is the matrix $(\widehat{G} + \widehat{G}^T)/2$ with the first ith rows and columns removed. As we know, $G = g_{ij}$ represents the coupling strength matrix of the network. So $\overline{G_i}$ only includes the nodes we do not consider as 'controllers'. In $\widehat{G}$, the elements are defined as $\widehat{g_{ii}} = \frac{\rho_{min}}{\gamma} g_{ii}$.

Generally speaking, we consider the first l nodes as the 'controllers', which are described as follows

$$\begin{cases} v_i &= -p_i e_i, \dot{p}_i = q_i \|e_i\|_2^2 e^{\mu t}, \qquad 1 \le i \le l \\ v_i &= 0, \qquad l < i \le N \end{cases} \tag{4.16}$$

where $\mu \ge 0$ and q_i is any random positive constant.

Consequently, the whole system becomes

$$\begin{cases} \frac{dx_i}{dt} &= f(x_i,t) + \sum_{j=1}^{N} g_{ij} H e_j - p_i e_i, \qquad 1 \le i \le l \\ \frac{dp_i}{dt} &= q_i \|e_i\|_2^2 e^{\mu t}, \qquad 1 \le i \le l \\ \frac{de_i}{dt} &= f(x_i,t) + \sum_{j=1}^{N} g_{ij} H e_j, \qquad l < i \le N \end{cases} \tag{4.17}$$

and the error system is

$$\begin{cases} \frac{de_i}{dt} &= Df(s) e_i + \sum_{j=1}^{N} g_{ij} H x_j - p_i e_i, \qquad 1 \le i \le l \\ \frac{dp_i}{dt} &= q_i \|e_i\|_2^2 e^{\mu t}, \qquad 1 \le i \le l \\ \frac{dx_i}{dt} &= Df(s) e_i + \sum_{j=1}^{N} g_{ij} H x_j, \qquad l < i \le N \end{cases} \tag{4.18}$$

where $Df(s)$ is the Jacobian matrix of f when $x = s$.

Theorem1 If there exists constant $1 \le l < N$ which makes $\lambda_l < -\frac{\alpha+\mu}{\gamma}$, Equation (4.10) will be locally stable at $s(t)$ under the condition as follows

$$\begin{cases} v_i &= -p_i e_i, \qquad \partial(p_i)/\partial t = q_i \|e_i\|_2^2 e^{\mu t}, 1 \le i \le l \\ v_i &= 0, \qquad l < i \le N \end{cases} \tag{4.19}$$

where $q_i > 0$, $\mu > 0$. λ_l is the largest eigenvalue of the lth block of the coupling matrix.

Proof: The Lyapunov function is as follows

$$\dot{\xi} = [1_N \bigotimes DF + \sigma G \bigotimes DH]\xi, \tag{4.20}$$

where p is a positive constant and $p > \alpha + \mu + \gamma\lambda_0$.

The derivative of V is

$$
\begin{aligned}
\frac{dV}{dt} &= \frac{1}{2}\sum_{i=1}^{N}((\partial(e_i)/\partial t)^T e_i e^{\mu t} + (\partial(e_i)/\partial t)e_i^T e^{\mu t} && (4.21)\\
&\quad +\mu e_i^T e_i e^{\mu t}) + \sum_{i=1}^{l}\frac{p_i - p}{q_i}\partial(p_i)/\partial t && (4.22)\\
&= [((\partial(e_i)/\partial t)^T e_i + (\partial(e_i)/\partial t)e_i^T + \mu e_i^T e_i && (4.23)\\
&\quad +\sum_{i=1}^{l}(p_i - p)e_i^T e_i)]e^{\mu t} && (4.24)\\
&= [\sum_{i=1}^{N} e_i^T(\frac{Df(s) + Df(s)^T}{2} + \mu I_N)e_i + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij}e_i^T H e_j && (4.25)\\
&\quad -\sum_{i=1}^{l} p e_i^T e_i]e^{\mu t} && (4.26)\\
&= [\sum_{i=1}^{N} e_i^T(\frac{Df(s) + Df(s)^T}{2} + \mu I_N)e_i + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij}e_i^T H e_j && (4.27)\\
&\quad +\sum_{i=1}^{N} g_{ii}e_i^T\frac{H + H^T}{2}e_i - \sum_{i=1}^{l} p e_i^T e_i]e^{\mu t} && (4.28)\\
&\leq [\sum_{i=1}^{N}(\alpha + \mu)e_i^T e_i + \sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N}\gamma g_{ij}\|e_i\|_2\|e_j\|_2 && (4.29)\\
&\quad +\sum_{i=1}^{N} g_{ii}\rho_{min}e_i^T e_i - \sum_{i=1}^{l} p e_i^T e_i]e^{\mu t} && (4.30)\\
&= [\eta^T((\alpha + \mu)I_N + \gamma\widehat{G} - D)\eta]e^{\mu t} && (4.31)\\
&= [\eta^T((\alpha + \mu)I_N + \gamma\frac{\widehat{G} + \widehat{G}^T}{2} - D)\eta]e^{\mu t}, && (4.32)
\end{aligned}
$$

where $D = diag(p, \cdots, p, 0, \cdots, 0)$ and $\eta = (\|e_1\|_2, \|e_2\|_2, \cdots, \|e_N\|_2)^T$.

From Lemma 1, we know that if p is large enough, we will have $(\alpha + \mu)I_N + \gamma\frac{\widehat{G} + \widehat{G}^T}{2} - D < 0$, which is equivalent to $(\alpha + \mu)I_N + \gamma\overline{G_l} < 0$. $\overline{G_l}$ is the matrix $\frac{\widehat{G} + \widehat{G}^T}{2}$

with the first l rows and columns removed. It is not difficult to achieve

$$\frac{1}{2}\|e_i(t)\|_2^2 e^{\mu t} = \frac{1}{2}e_i^T(t)e_i(t)e^{\mu t} \leq V(t) < V(0). \tag{4.33}$$

Hence, we have

$$\|e_i(t)\|_2 < \sqrt{2V(0)}e^{-\frac{\mu}{2}t}. \tag{4.34}$$

The local stability exponent of the error system in Equation (4.11) is stable, which means that the system in Equation (4.10) is stable locally.

For a network with a fixed structure, it is possible to choose the controllers by Theorem 1 and make the exponent of the network synchronization stable asymptotically.

4.4.3 The global stability of synchronization

First, we give a hypothesis about $f(x,t)$ for an individual system:

Hypothesis 2 If there exits a constant matrix K and $\|K\|_2 = \theta$, then we have

$$(x-y)^T(f(x,t)-f(y,t))) \leq (x-y)^T KH(x-y) \tag{4.35}$$

where H is a positive-definite matrix.

In this way, we can expand what we have achieved from the local stability to the global stability.

Theorem1 If there exists a constant $1 \leq l < N$ which makes $\lambda_l < -\theta - \mu/\gamma$ and

$\psi > 0$, then the system charged by Equation (4.10) will be globally stable under the condition as follows

$$\begin{cases} v_i = -p_i e_i, \qquad \partial(p_i)/\partial t = q_i \|e_i\|_2^2 e^{\mu t}, 1 \le i \le l \\ v_i = 0, \qquad l < i \le N \end{cases} \tag{4.36}$$

where $q_i > 0$, $\mu > 0$ and λ_l is the largest eigenvalue of the lth block of the coupling matrix.

Proof: The Lyapunov function as follows

$$\dot{\xi} = [1_N \bigotimes DF + \sigma G \bigotimes DH]\xi \tag{4.37}$$

where p is a positive constant and $p > \gamma\theta + \mu + \gamma\lambda_0$.

The derivative of V is

$$
\begin{aligned}
\partial(V)/\partial t &= \sum_{i=1}^{N} e_i^T(\partial(e_i)/\partial t)e^{\mu t} + \frac{1}{2}\mu e_i^T e_i e^{\mu t}) + \sum_{i=1}^{l}\frac{p_i - p}{q_i}\partial(p_i)/\partial t && (4.38)\\
&= [\sum_{i=1}^{N}(e_i^T((f(x(t),t) - f(y(t),t))) + \mu e_i^T e_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij} e_i^T H e_j && (4.39)\\
&\quad - \sum_{i=1}^{l} p e_i^T e_i + \sum_{i=1}^{l}(p_i - p)e_i^T e_i]e^{\mu t} && (4.40)\\
&\leq [\sum_{i=1}^{N}(e_i^T K H e_i + \mu e_i^T e_i + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij} e_i^T H e_j - \sum_{i=1}^{l} p e_i^T e_i]e^{\mu t} && (4.41)\\
&\leq [\sum_{i=1}^{N}(e_i^T(\gamma\|K\| + \mu)I_N e_i + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij} e_i^T H e_j && (4.42)\\
&\quad \sum_{i=1}^{N} g_{ii} e_i^T \frac{H + H^T}{2} e_i - \sum_{i=1}^{l} p e_i^T e_i]e^{\mu t} && (4.43)\\
&= [\eta^T((\gamma\theta + \mu)I_N + \gamma\widehat{G} - D)\eta]e^{\mu t} && (4.44)\\
&= [\eta^T((\gamma\theta + \mu)I_N + \gamma\frac{\widehat{G} + \widehat{G}^T}{2} - D)\eta]e^{\mu t} && (4.45)
\end{aligned}
$$

where $D = diag(p, \cdots, p, 0, \cdots, 0)$ and $\eta = (\|e_1\|_2, \|e_2\|_2, \cdots, \|e_N\|_2)^T$.

From Lemma 1, we know that if p is large enough, we will have $(\gamma\theta + \mu)I_N + \gamma\frac{\widehat{G}+\widehat{G}^T}{2} - D < 0$, which is equivalent to $(\gamma\theta + \mu)I_N + \gamma\overline{G_l} < 0$. $\overline{G_l}$ is the matrix $\frac{\widehat{G}+\widehat{G}^T}{2}$ with the first l rows and columns removed. It is not difficult to achieve

$$
\frac{1}{2}\|e_i(t)\|_2^2 e^{\mu t} = \frac{1}{2}e_i^T(t)e_i(t)e^{\mu t} \leq V(t) < V(0) \qquad (4.46)
$$

Hence, we have

$$\|e_i(t)\|_2 < \sqrt{2V(0)}e^{-\frac{\mu}{2}t} \tag{4.47}$$

The global stability exponent of the error system in Equation (4.11) is stable, which means that the system in Equation (4.10) is stable globally. In Theorems 1 and 2 , we assume that we choose the first l nodes as the controllers, and achieve the local and global synchronization. It will be significantly different if we rank the nodes in different ways.

4.5 Remarks

In this chapter, we have reviewed the history of the area of synchronization in the dynamical system. We discussed the Lyapunov exponent and the master stability function. Then we developed a method to choose some nodes in a network as the 'controllers'. The other nodes in the network are attracted by the 'controller' which are governed by particular dynamics. In this way, we achieved the local and global stability in synchronization. The further study will target at the relation between the network structure and the selection of 'controllers'.

Chapter 5

Synchronization process to partition networks

5.1 Introduction

5.1.1 Communities in a network

Communities in a network are defined as the subgraphs with higher density of connections within them than between them. In Figure 5.1, the three modules in circles with apparently higher inner-density can be considered as communities. The problem of community detection originated from graph theory decades ago [MacQueen, 1967, Barnes, 1981]. The algorithms have been developed and used by researchers from various disciplines [Pothen, 1997, Junker and Schreiber, 2008, Sen et al., 2006], especially the complex networks [Newman, 2006b, Lancichinetti and Fortunato, 2012].

Community structures occur in many networked systems from society, computer science, biology and economics. The examples include the working and friendship cir-

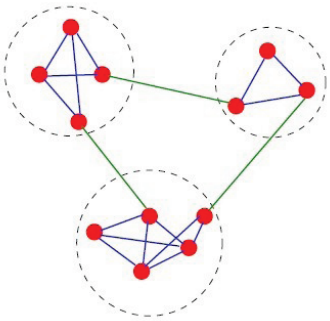


Figure 5.1: The communities in a network

cles, the clients with similar interests on online retailers such as Amazon, the proteins having the same characters within the cell [Sen et al., 2006]. When modelling them mathematically, the nodes and connections may have specific definitions and weights. The communities may act with specific functions in a network. The online customers divided into one group may maintain the similar habit of purchase, which enables the recommendation system to provide effective guide and advertisements.

The community detection gives insight into the network structure and helps to control the network. The node at the central position of a cluster to connect most of the group partners is crucial to the stability of the cluster. The nodes at the boundaries between clusters are important to the partitions of networks and the exchanges of opinions.

In some networks, the community detection displays the hierarchial structure. For instance, the town consisting of small villages is an element of a city. The human body is composed of organs and organs are composed of tissues.

The purpose of community detection is to find the modules and hierarchy using

the network topology. Most of the previous research have dealt with the unweighted-undirected networks. The algorithms from graph theory can not cover all kinds of real networks [Von Luxburg, 2007, Yu et al., 2006, Newman, 2000]. The frequently used Laplacian matrices consist of zero and positive elements as defined. However, in a social network, a negative connection is allowed, which presents a negative element in the adjacency matrix and Laplacian matrix.

The structures may be another barrier when detecting the communities in a dynamical network. When the network is homogeneous in degree, any node to put in a cluster will contribute the similar numbers of connections to the inner and outside density of the cluster (Figure 5.2). When the networks are inhomogeneous, those less connected nodes tend to be divided as individual clusters by algorithms both from the Laplacian spectrum and the graph cut. In some applications of networks, despite of all these difficulties, it is necessary to uncover the similarity of nodes, the organizations of the networks, etc. An example is RALIC network (Figure 5.3) which we will introduce in Section 5.5. Therefore, we have to investigate the information hidden in the combination of connections, graph structure and weighings to ensure that the network is clear enough to partition.

5.1.2 Research objective

Recently there has been a trend to reveal the hidden relations of nodes before partitioning the network [Lancichinetti and Fortunato, 2012]. The network has some sort of precondition process used to help reveal its structure and the conditioned network

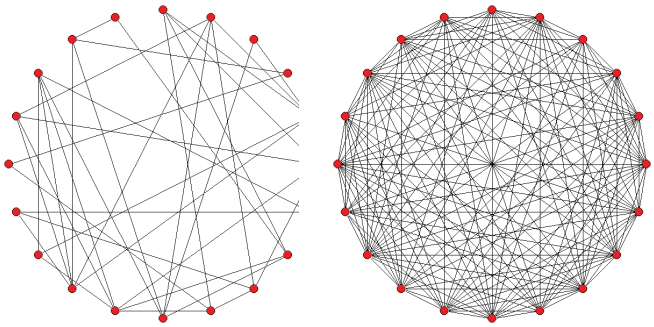


Figure 5.2: Homogeneous structures

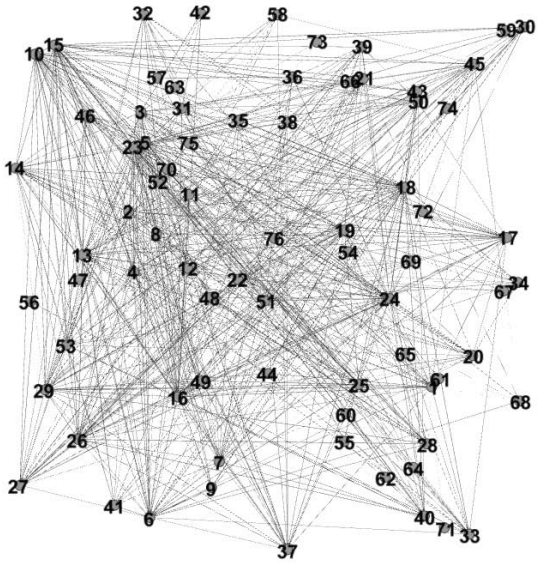


Figure 5.3: The network of RALIC project

is then divided using a standard partitioning algorithm. In this research, we intend to develop a method of this kind.

A new dynamical method *ODM* (Opinion Dynamics Matrix) will be developed based on the opinion dynamics to investigate not only the connections underlying communities, but also the hierarchy of communities within a directed-weighted network as well.

5.1.3 Research structure

The introduction is followed by Section 5.2, where a brief review of the related work in community detection, network partition,eing is given. In Section 5.3, we introduce the optimization rule and algorithm of the *ODM* method which develops a dynamical matrix by the opinion convergence process on a network. The dynamical matrix can be used in network partition instead of the adjacency matrix. In Section 5.4, we illustrate and compare several possible algorithms. Three benchmark networks are built to test the algorithms with and without the *ODM* method. In Section 5.5, the RALIC project in UCL is introduced, which provides a real network to be partitioned. In Section 5.6 , we describe the RALIC network and apply the algorithms from Section 5.4 to compare their performances to partition the networks. In Section 5.7, an optimization of the *ODM* is tested and analyzed. In Section 5.8, some aspects from the point of view of software engineering are discussed. Section 5.9 is the remark of the chapter.

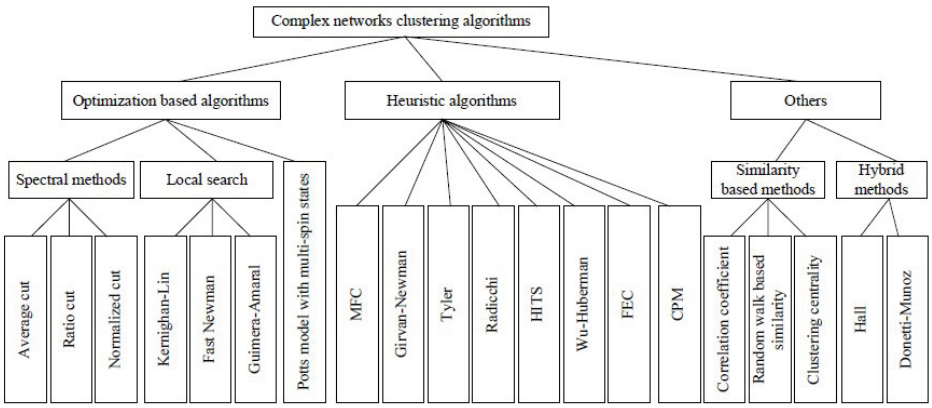


Figure 5.4: Classification chart of algorithms from [Yang et al., 2009]

5.2 Related work

Various terms are used in this area, such as 'community detection', 'network clustering', or 'network partition'. In this study, we put them together as a class of algorithms to investigate the subgraphs of a graph. Figure 5.4 shows a classification of algorithms in this kind which have been extensively tested [Yang et al., 2009, Fortunato, 2010].

Here, we discuss the spectral methods, local search and heuristic algorithms respectively, which will be used later in Sections 4.4-4.6.

5.2.1 The spectral method

Historically, the spectral method was first used to partition graphs [Barnes, 1981] and then applied to the complex network clustering [Newman, 2006b, Motoki et al., 2007].

In graph theory [Euler, 1736, Newman, 2000], a graph $G = \{E, V\}$ is defined by the edge set E and vertex set V [MacQueen, 1967, Newman, 2000]. In a random subgraph $C = \{E_C, V_C\}$, $E_C \in E, V_C \in V$, an internal edge e_{ij} has two vertices $i, j \in V_C$. An external edge e_{kl} has one vertex $k \in V_C$ and the other outside the subgraph. In

this study, we define the total number of internal edges inside C as the internal density ρ_{in}^{C}. We divide the graph into m disjoint subgraphs $C_1, C_2, \cdots C_m$, $C_p \cap C_q = 0, p, q \in m$ and $C_1 \cup C_2 \cdots C_m = G$. The sum of all internal densities of subgraphs makes the total internal density ρ_{in}. The remainer in the edge set, ρ_{ex}, connects the subgraphs. For a particular subgraph, if the internal density is considerably higher than the external edge density, we consider it a community structure, or a cluster [Albert and Barabasi, 2001, Newman, 2003, Girvan and Newman, 2002].

The spectral method uses ρ_{ex} as the 'cut function' and minimizes it to find a partition. Given the adjacency matrix A of a network, we define an all one vector $\mathbf{1}$ and build a diagonal matrix G with G_{ii} as the degree of node i

$$G = diag(AI). \tag{5.1}$$

So we can define the Laplacian matrix M as

$$M = A - G. \tag{5.2}$$

Some alternations of M are used as 'cut functions' in different researchs. Take the 'standard cut' $M = G^{-1/2}(G-A)G^{-1/2}$ as an example. If we partition the network into two parts, the eigenvector corresponding to the second largest eigenvalue gives the approximation to an optimal cut. Then, the smaller clusterings will be found based on the two big clusterings. Multiple clusters can be found by repeating the process.

Here we will a use bisecting network as an example to explain how the 'optimal cut' works. Assume a network G consists of two disjoint subgraphs G_1 and G_2. The cut between them is defined as

$$\text{cut}(\text{G}_1, \text{G}_2) = \sum_{i \in G_1, j \in G_2} w_{ij} A_{ij}. \tag{5.3}$$

Here A is the adjacency matrix and w_{ij} is the weight on connection a_{ij} . The partition is to minimize the cut. Since some isolated nodes may be divided as single clusterings. The ratiocut is used to balance the number of nodes in two partitions.

$$\text{ratiocut}(\text{G}_1, \text{G}_2) = \frac{\text{cut}(\text{G}_1, \text{G}_2)}{|G_1|} + \frac{\text{cut}(\text{G}_2, \text{G}_1)}{|G_2|}. \tag{5.4}$$

where$|G_1|$ is the number of the nodes in G_1. Given a $1 \times N$ vector f, we have

$$f_i = \begin{cases} \sqrt{|G_2|/|G_1|}, \text{if } i \in G_1 \\ -\sqrt{|G_1|/|G_2|}, \text{if } i \in G_2 \end{cases} \tag{5.5}$$

therefore

$$
\begin{aligned}
f^T M f &= \frac{1}{2}\sum_{i,j=1}^{N} A_{ij}(f_i - f_j)^2 \\
&= \sum_{i\in G_1, j\in G_2} w_{ij}(\sqrt{|G_2|/|G_1|} + \sqrt{|G_1|/|G_2|})^2 + \qquad (5.6) \\
&\quad \sum_{i\in G_2, j\in G_1} w_{ij}(-\sqrt{|G_2|/|G_1|} - \sqrt{|G_1|/|G_2|})^2 \\
&= 2\mathrm{cut}(\mathrm{G}_1,\mathrm{G}_2)(\sqrt{|G_2|/|G_1|} + \sqrt{|G_1|/|G_2|} + 2) \\
&= 2\mathrm{cut}(\mathrm{G}_1,\mathrm{G}_2)(|G_2| + |G_1|/|G_1| + |G_1| + |G_2|/|G_2|) \\
&= 2N * \mathrm{ratiocut}(\mathrm{G}_1,\mathrm{G}_2). \qquad (5.7)
\end{aligned}
$$

In this case, the minimal cut is equivalent to finding the second largest eigenvalue of M. The method can be expanded to the k-cluster condition.

However, this algorithm is effective in an undirected graph represented in relatively dense matrix, but has limitations when the network is directed or sparse. Suppose a directed network with one node i which only has in-degree but no out-degree, then, the ith row of A will be empty, which may cause one more zero eigenvalue of M. If we treat this eigenvalue as the second largest one, it will cut node i as an individual cluster, which may not be a proper partitioning.

5.2.2 The local search

This class of algorithms has frequently been studied [Fortunato, 2010, Guimera and Amaral, 2005, Newman, 2004]. With the target of the fixed number of clusters, they usually consist of three parts:

Seeds Given an N-node network, preassign k clusters. We will pick k nodes randomly as seeds and wait for new nodes to join in.

Cost function A parameter to measure the distance from one node to another node or to a prospective cluster.

Search Strategy Put those nodes which have not been divided into any cluster, one by one, into k clusters to maximize or minimize the cost function.

This kind of algorithms is sensitive to initial conditions. The different choices of seeds may bring totally different results. An example is illustrated in an unweighted network in Figure 5.5. Suppose $k = 3$, the cost function is the distance to seeds and the search strategy is to minimize the new member's distance to the seeds. The partition with seeds as $\{n2, n5, n9\}$ and $\{n1, n6, n9\}$ will give totally different results as shown in Figure 5.5, above and below respectively.

5.2.3 Heuristic algorithms

Heuristic algorithms look for optimized partitions of networks based on some intuitive assumptions. In [Flake et al., 2002], it was assumed that the place where the minimum cut occurs there might be a 'bridge' between clusters. In Girvan-Newman (*GN*) method [Girvan and Newman, 2002, Newman and Girvan, 2004], it was assumed that the edges between clusters should be larger than the edges within them. The 'edge betweenness' means the number of the shortest paths that go through an edge in a graph or network. A detailed algorithm to calculate it and execute the *GN* will be given in Section 5.4. However, there is no analytical proof to guarantee the optimization of the

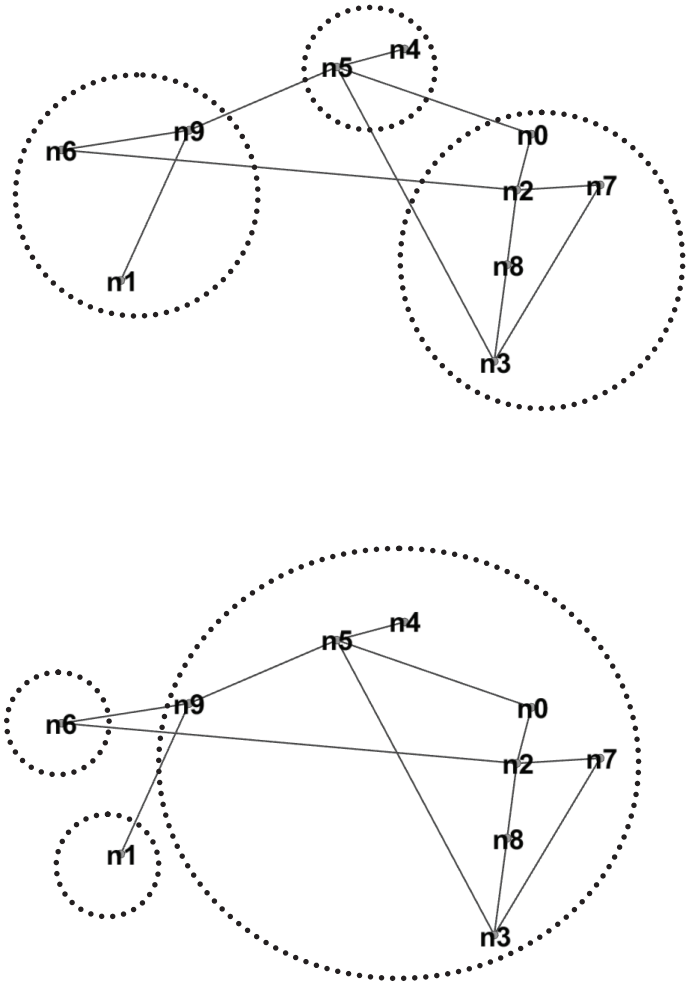


Figure 5.5: Two partitions of one network from different seeds

solutions for every network, especially when it is directed. Besides, this method tends to define some of less-connected nodes to lie in individual clusters with one member. This is rigorous as a solution of a graph problem but makes no sense for a directed-weighted social network.

5.2.4 Conclusion

Many algorithms to cluster (or partition) a network are developed from graph theory and suffer the problems we have listed in Section 5.1. Therefore, we will attempt to develop a new method, the *ODM* (Opinion Dynamical Matrix) method to make the network structure more clear and avoid the possible errors caused by the application of graph theory to real-life networks.

5.3 *ODM* from opinion convergence

In this section we will introduce the *ODM*(Opinion Dynamical Matrix) algorithm. Previously, all methods of network partition use the adjacency matrix as the input. In the *ODM*, we make the opinions evolve on a network, and observe the speed of local convergence of opinions in order to determine the closeness between two nodes. We develop a dynamical matrix, where nodes i and j are connected if their opinions converge fast enough. Therefore, we reveal clearer network structure to partition.

5.3.1 Opinion Dynamics

We will use the linear opinion model of Curtis and Smith [Curtis and Smith, 2008] in this study. The opinion evolution is governed by the equation below

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} b_{ij}a_{ij}(x_j - x_i), i = 1,2\ldots N, t \geq 0. \tag{5.8}$$

5.3.2 The use of opinion convergence to reveal network structures

We consider an N-node network as a single graph in the macroscale or N subgraphs in the microscale. In the middle, there still exist several levels of scale. A simple example is a city as marcoscale, which can be divided into districts, communities, families in communities until every individual in a family is the microscale. As opposed to those methods aiming to partition a network merely at a single scale, synchronization of network may exhibit communities at all possible scales during the process.

Previous researches [Lu et al., 2010, Blasius et al., 1999, Li et al., 2012, Arenas et al., 2006] have done a lot using a Kuramoto model which considers the networks as a mean-field [Barabasi et al., 1999], where all nodes are motivated only by a field force. As in the following dynamics, K is used as mean-field interaction

$$\frac{dx_i}{dt} = \omega_i + K\sum_j l_{ij}sin(x_j - x_i), \tag{5.9}$$

where x_i stands for the phase of oscillator i to indicate all physical properties of it. The ω_i is the natural frequency at which the oscillator tends to vibrate when motivated.

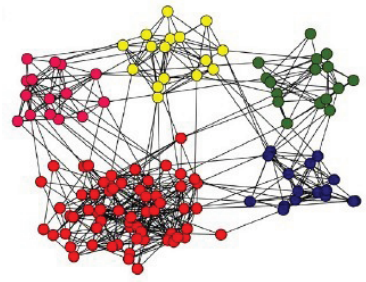


Figure 5.6: A network with 5 clear natural clusterings [Arenas et al., 2006]

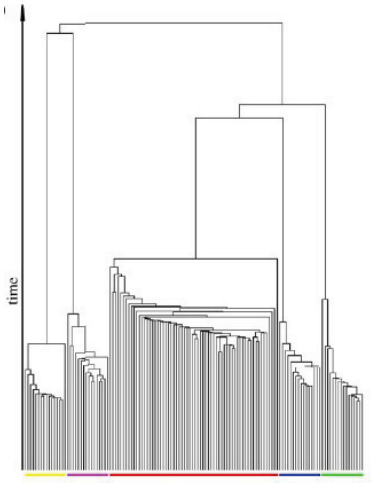


Figure 5.7: Local convergence before global convergence [Arenas et al., 2006]

The l_{ij} is 1 when there is a connection between i and j, otherwise 0. The coupled oscillators were defined in the mechanism system and are used to describe some other phenomenon recently. For instance, the neurons in brain as oscillators with their firing rate as the natural frequency [Cumin and Unsworth, 2007], a person as an oscillator with his/her opinions as the phase [Pluchino et al., 2005]. The kuramoto models have been applied to many areas partially or thoroughly. In this study, we will use the idea of the synchronization process on Kuramoto model to investigate the similar phenomena in opinion dynamics. As can been observed in Figure 5.6-7, with a clear structure as shown in Figure 5.6, the phase of some nodes come identically to each other before all nodes have a same phase as shown in Figure 5.7.

In opinion dynamics, we do not have the global field force K to drive the system to evolve. Instead, there is an influence ability b_i of each node i to others as shown in Equation (5.5). We observe a similar phenomenon here. Those who connect to each

other with smaller cluster coefficient converge faster than the average of a network, by which we want to divide the hierarchy eventually as shown in Figure 5.7. The solution at any time t should always be

$$X^t = X^0 e^{\Lambda t}. \tag{5.10}$$

where λ_i is the ith eigenvalue of M. If we put the equation in the order of eigenvalues, the time for them to achieve zero like shown in Figure 5.7. So two conclusions can be obtained here:

1. Once the Laplacian matrix is fixed, the partition of the network will always be revealed given any initial opinions.

2. The time scale can be revealed by ranking the equations and consequently the sub-structure of the network at different levels.

Here we test on a 32-node regular network with four identical clusters in Figure 5.8. Opinion difference between every pair of nodes changes based on the clusters of network.

When the opinion $x_i \in (0,1)$, the values of $x_i - x_j$ and $sin(x_i - x_j)$ don't show apparent difference during the opinion evolution, as shown in Figure 5.9. The local convergence ahead of the global convergence occurs in the opinion dynamics we build in this study.

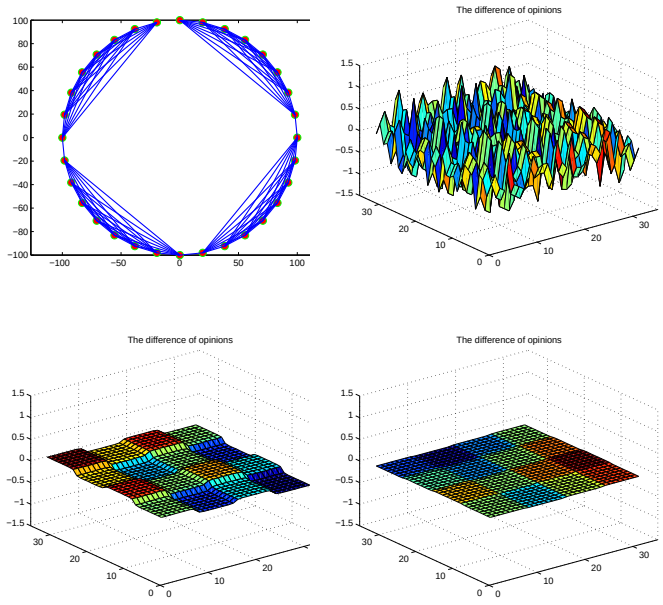


Figure 5.8: Opinion difference between nodes in a regular network

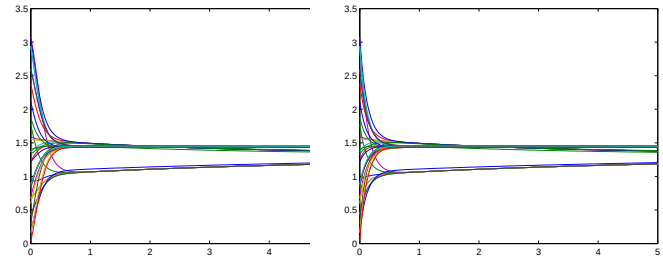


Figure 5.9: Opinion evolving on Kuramoto model and opinion dynamical model

5.3.3 Basic idea of *ODM*

During the evolution of opinions, those who converge faster than the global convergence can be identified as 'closer friend' no matter whether there is a real connection between them. This is the assumption of the *ODM* approach.

In Figure 5.10, it is not difficult to prove that nodes $n1$ and $n3$ without any direct connection may have interactions through their three common 'friends' nodes $n0$, $n2$

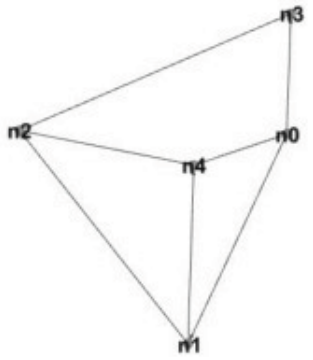


Figure 5.10: The communities in a network

and $n4$ and get a local convergence faster than the global convergence if the local density among nodes $n0-n4$ is higher than the global density. Therefore, it is allowed to put a connection can be put between node $n1$ and $n3$. We present a method of determining a dynamical matrix D instead of A for community detection.

5.3.4 Otimization of *ODM*

As we have illustrated in Chapter 2, the Laplacian matrix of the connected network without isolated part has one zero eigenvalue. Any single isolated part will increase one zero eigenvalue in the spectrum of the Laplacian matrix. An ideal condition for the network clustering is that there are $k(k>1)$ eigenvalues in the Laplacian spectrum, which means there are k exact isolated components of the network. However, in most real-world networks and the simulated complex network, this kind of natural clusters do not exist. The *ODM* is used as a pre-process method to optimize the adjacency matrix to impose a rank constraint. In this way, we build a similarity matrix with eigenvalues very close to zero in order to find the clusters in the network. In Section

1.3.2, we have provided a simple method based on the synchronization process to test the number of clusters k. Next, we show an effective method to describe the two targets we aim at in this Chapter: First, to find a dynamical matrix D with a low cost of calculation; Second, the matrix D should have k eigenvalues which equal or are as close as possible to zero. For the first target, we build the objective function as follows

$$\min \|d_{ij} - a_{ij}\|_2^2. \tag{5.11}$$

When it comes to the second target, a basic equation in spectral analysis is used

$$\sum_{i,j} \|f_i - f_j\|_2^2 a_{ij} = 2Tr(F^T MF), \tag{5.12}$$

where M is the Laplacian matrix and $F \in R^{N\times k}$. It can be proved that if $rank(D) = N-k$, the dynamical matrix D will contain exact k clusters. Thus we add this rule as a constraint to the problem as follows

$$min \sum_{i,j} (\|x_i - x_j\|_2^2 d_{ij} + \gamma d_{ij}^2). \tag{5.13}$$

To tackle it, let $\lambda_i(M)$ denotes the i-th eigenvalues closest to zero. As M is negative semi-definite, we know $\lambda_i(M) \leq 0$. Considering the Ky Fan theorem, we have

$$\sum_{i=1}^{k} \lambda_i(M) = minTr(F^T MF). \tag{5.14}$$

To combine our two targets, we rewrite the objective function as

$$min \sum_{i,j} (\|x_i - x_j\|_2^2 d_{ij} + \gamma d_{ij}^2) + 2\sigma Tr(F^T M F). \tag{5.15}$$

Apparently, as long as σ is large enough, the second part of the function will approach zero and the clusters will be revealed easier than before. In order to investigate the structure of the clusters, we will use the algorithm below to process the clustering.

5.3.5 Algorithm of *ODM* method

In this study, we investigate a new method of community detection, the *ODM*, Opinion Dynamics Matrix, to precondition the networks to be partitioned. As a motivation for this method, one may imagine the opinion evolution that might take place. Those nodes who were more connected by their peers might find that their opinions hold greater weight in the discussion process. We precondition that the network description of the single value opinion or opinion matrix provides with a model of this discussion process, taking inspiration from a developed theory of opinion dynamics. It has been established [Arenas et al., 2006] that the synchronisation of opinions to a consensus within different components of the network proceeds at different rates which reflect the size of the component, or community. Therefore, it is possible that controls over the granularity to which the project engineer wishes to consider the data as their initial opinions, may be obtained by allowing the initial preconditioning to correspond to a discussion of varying lengths.

The main idea of the study is to further increase the internal connections in the pos-

sible clusters when the network is loosely connected and clusters are not clear enough to divide.

We have two algorithms to obtain the communities and opinions in different scales in an N-node system. One is the Evolving algorithm:

1. Consider every person as a node, and if person i rates person j, there is a link between them and they talk to each other through the link;

2. Every node i holds an initial opinion x_i^0;

3. Use the Equation set (5.2) to evolve opinions until everyone has the same opinion x_i^t at time t.

4. Given a threshold speed Th and an $N*N$ matrix D, if $\frac{x_j^t - x_i^t}{x_j^0 - x_i^0}$ is smaller than Th, the opinions of person i and person j converge fast and we determine $D_{ij} = 1$, otherwise $D_{ij} = 0$.

5. Use the Spectral, K-clustering or G-N method to cluster the network.

5.4 Pre-test on normal networks as benchmarks

Before we develop a new algorithm, we present an investigation into what a good algorithm is and what a good partition is. The previous studies [Fortunato, 2010, Lancichinetti et al., 2008] use benchmarks(several designed networks with particular structures, as shown in Figure 5.10 to test the following statistical characteristics of the partitions:

1. Is the inner-density of degree ρ_{in} significantly larger than ρ_{ex}?

2. Are the partitions similar to each other if one algorithm is tested multiple times with different choices of seeds on one same network?

Any possible cluster will be defined by comparison of the actual cluster inner-density regardless of the network structure. A possible error has never been corrected in the first measurement if the algorithm put most of the nodes in one cluster. In this case, the ρ_{in} can be extremely high even the natural clusters are not identified.

The *ODM* method is an attempt to provide the information emerging from the evolution of a network to a chosen algorithm. In this chapter, we will test it followed by three typical algorithms respectively given in Section 5.2.

Spectral partition to represent spectral methods based on an optimal 'cut'.

K-clustering to represent local search method.

Girvan-Newman to represent Heuristic algorithms.

We will compare the pairwise results from the original algorithms and those after preconditioning by *ODM*. The tests are taken on three designed networks (see Figure 5.10) with same numbers of nodes and connections but different structures. Meanwhile, a new measurement is developed to avoid the formation of any oversized clusters.

5.4.1 Algorithm of Spectral method

Given N nodes to put into k clusterings [Von Luxburg, 2007]:

1. Make the affinity matrix F by $F_{ij} = e^{(-|a_{ij}w_{ij}|^2/2\sigma^2)}$. The scaling parameter σ^2 controls how rapidly the F_{ij} falls off with the opinion distance between x_i and x_j. In this study $\sigma = 1$.

2. Define D as a diagonal matrix whose ith diagonal element is the sum of F's ith row and construct the matrix $l = D^{-1/2}FD^{-1/2}$.

3. Find the k largest eigenvalues of L and the corresponding eigenvectors to make a matrix K.

4. Form the matrix Y by normalizing the matrix K like $Y_{ij} = K_{ij}/(\sum_j K_{ij}^2)^{1/2}$

5. Treat each row of Y as a node and put it into the kth cluster by minimizing the distortion.

6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

5.4.2 Algorithm of K-clustering

The methods like K-clustering [MacQueen, 1967, Fortunato, 2010] usually determine N nodes as the centers of clusters. Then the rest of the nodes will be assigned into one of the prospective clusters by minimizing a cost function. K-clustering uses the shortest path length between pairwise nodes as the cost function, which records the minimum of all the possible paths for nodes i to get to node j. In Figure 5.11, all the possible paths from node A to node F include:

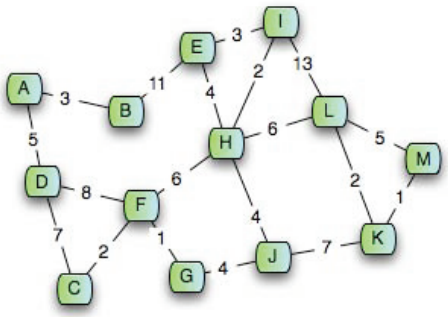


Figure 5.11: The communities in a network

- A-D-F
- A-D-C-F
- A-B-E-H-F
- A-B-E-I-H-F
- A-B-E-I-L-H-F
- ...

The shortest path among all the weighted path is A-D-F and the path length is 13.

Before we apply the K-clustering on any network, we need to calculate the shortest path length between every two nodes:

1. Start from one node i and choose a target node j. The initial distance between any two nodes q and p is $d_{qp} = w_{qp} l_{qp}$, where $l_{qp} = 1$ when there is a connection between q and p and 0 otherwise.

2. Create a set of unvisited nodes called the unvisited set consisting of all the nodes except for i and the visited set.

3. For any node k connected to node i in the unvisited set, put it in the 'visited set'. If there exists $d_{ik} + d_{kj} < d_{ij}$, make $d_{ik} + d_{kj}$ the new tentative distance d_{ij}.

4. When all of the neighbors of node i are done, repeat process 3 for all nodes in the visited set.

5. If node j has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal), then stop. The algorithm has finished.

6. Otherwise, repeat step $1-5$ to all nodes.

With the shortest path length d_{ij} between every pair of nodes i and j obtained, we proceed the K-clustering algorithm:

1. Randomly pick k nodes as centers of k prospective clusters.

2. Among all the rest nodes, we pick one, node i, and calculate $\sum_1^{j \in c_k} d_{ij}$ for all the k clusters. Put node i in the cluster with $\min \sum_1^{j \in clusterk} d_{ij}$.

3. Repeat 2 until all nodes are assigned to a clustering.

5.4.3 Algorithm of Girvan-Newman

This algorithm focuses on those edges which are least central,i.e, the edges which are most between communities [Newman and Girvan, 2004, Girvan and Newman, 2002]. In an unweighted N-node network, there are $N(N-1)/2$ shortest paths for any pair of nodes to find each other. The edge betweenness of the edge e_{ij} counts how many shortest paths are through e_{ij}. Rather than constructing communities by adding the strongest edges to an initially empty vertex set, we

1. Calculate the betweenness for all edges in the network.
2. Remove the edge with the highest betweenness.
3. Recalculate betweennesses for all edges affected by the removal.
4. Repeat from step 2 until no edges remain.

5.4.4 Benchmark networks

In this study, to test the performance of all the algorithms, we build three 20-nodes networks, with clear, normal and homogeneous structures respectively, as shown in Figure 5.12.

We assume the networks are unweighted-undirected in this test. In both the preliminary tests in this section and and real-life data tests in Section 5.6, we choose to partition a network into four parts as examples.

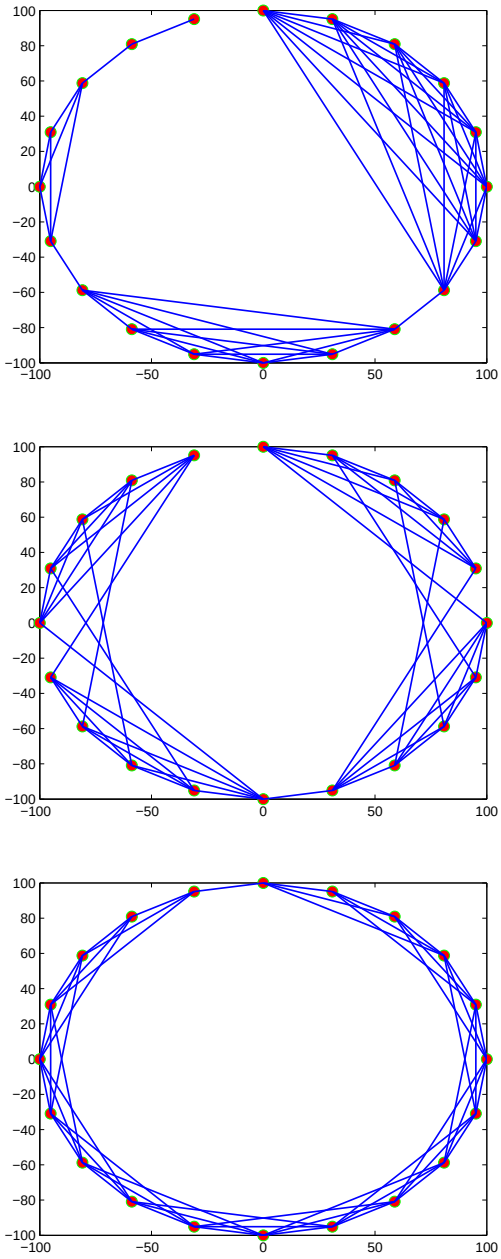


Figure 5.12: Networks with different natural structures

5.4.5 Measurement and Comparison

In this part of study we will use the modularity Q from [Newman, 2006a] to measure the partitions from all algorithms.

$$Q = \frac{1}{2N}\sum_{ij}(a_{ij} - \frac{k_i k_j}{2N})\delta_{ij}, \tag{5.16}$$

where N is the number of nodes in the network, A_{ij} is the corresponding element in the adjacency matrix, k_i is the degree of node i and $\delta_{ij} = 1$ when i and j are assigned in one cluster, otherwise 0.

Here we assume all the weights between nodes are assigned to 1 if there is a connection, and 0 otherwise. In Figure 5.13, the S represents 'Spectral method' while $S-ODM$ is the spectral method following ODM. The spectral method gives precise solution in clear network and reasonable solution in the medium one. However, when it comes to unclear and homogeneous network, the accuracy drops drastically. The connections added by $S-ODM$ make the clear and medium networks more homogeneous, which is unnecessary. But in an unclear network, the new connections enhance the inner-density and emerge the communities. The K represents 'K-clustering method' while $K-ODM$ is the K-clustering method following ODM. Since the K method is seed-sensitive, we put all possible seeds into consideration and average the results to get what are shown in Figure 5.12. As in the comparison between S and $S-ODM$, the $K-ODM$ may break the natural structures and cause errors in clear and medium networks. The extremely high accuracy when it partitions the clear network is a mis-

take which can be identified in Figure 5.10. But the less clear the network is, the more effective the $K-ODM$ will be. The GN represents 'Girvan-Newman method' while $GN-ODM$ is the GN method following ODM. The GN method detects precisely the natural clusterings in networks but cannot identify a single clustering until removing most of the connections to determine some isolated nodes each as one individual cluster. The problem of an unreasonable large cluster is serious by this method, even if it is optimized by the ODM.

Then, we test the weighted network. We give each connection of the three networks in Figure 5.12 a random weight $w_{ij} \in (0,1)$. The three groups of algorithms are used. The measurement of modularity is

$$Q = \frac{1}{2N}\sum_{ij}(A_{ij}w_{ij} - \frac{kw_i kw_j}{2N})\delta_{ij} \tag{5.17}$$

where w_{ij} is the weight on the edge between node i and node j and kw_i indicates the sum of the edge weights with one of the vertices i. The results are shown in Figure 5.14.

The conclusions are:

1. The ODM is not suitable for those networks with natural clusterings. It is effective in enhancing the heterogeneous of degree and decreasing the sparseness of a network.

2. If the network is too homogeneous or sparse, the GN and the S method will both make the mistake to determine several nodes with one node as one cluster and

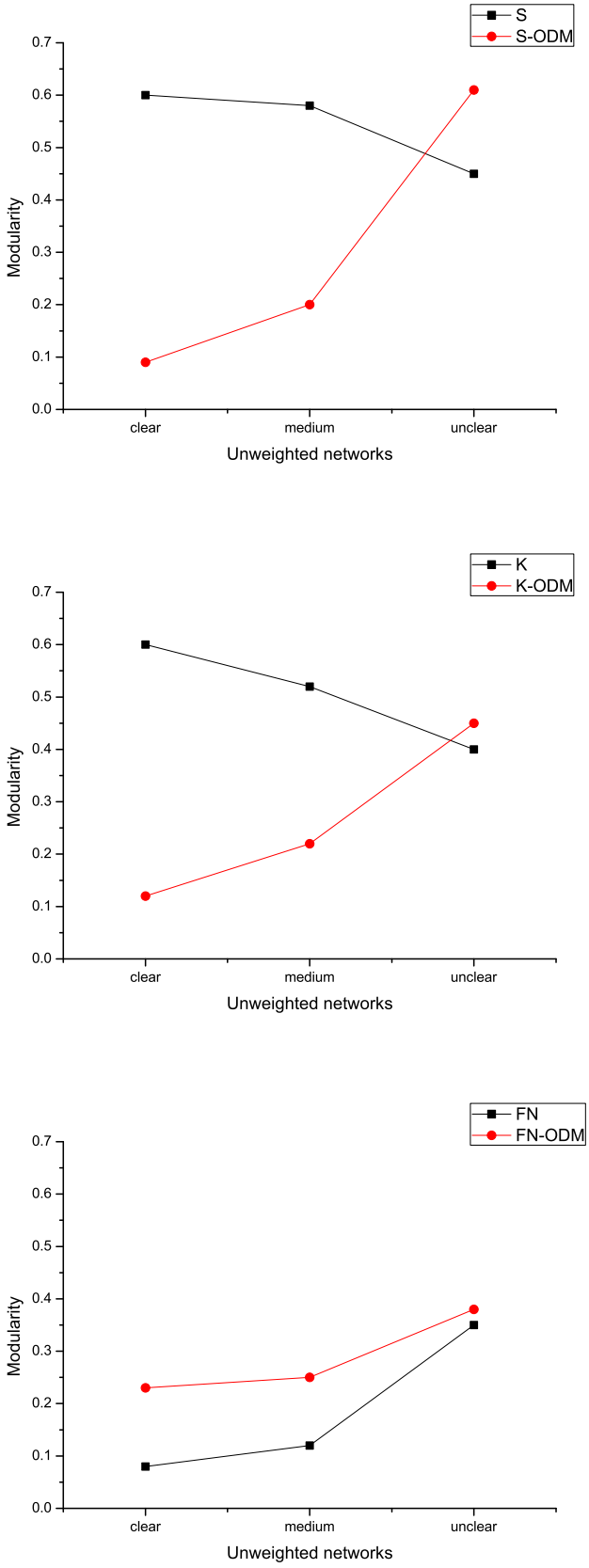


Figure 5.13: Comparisons of modularity on unweighted networks

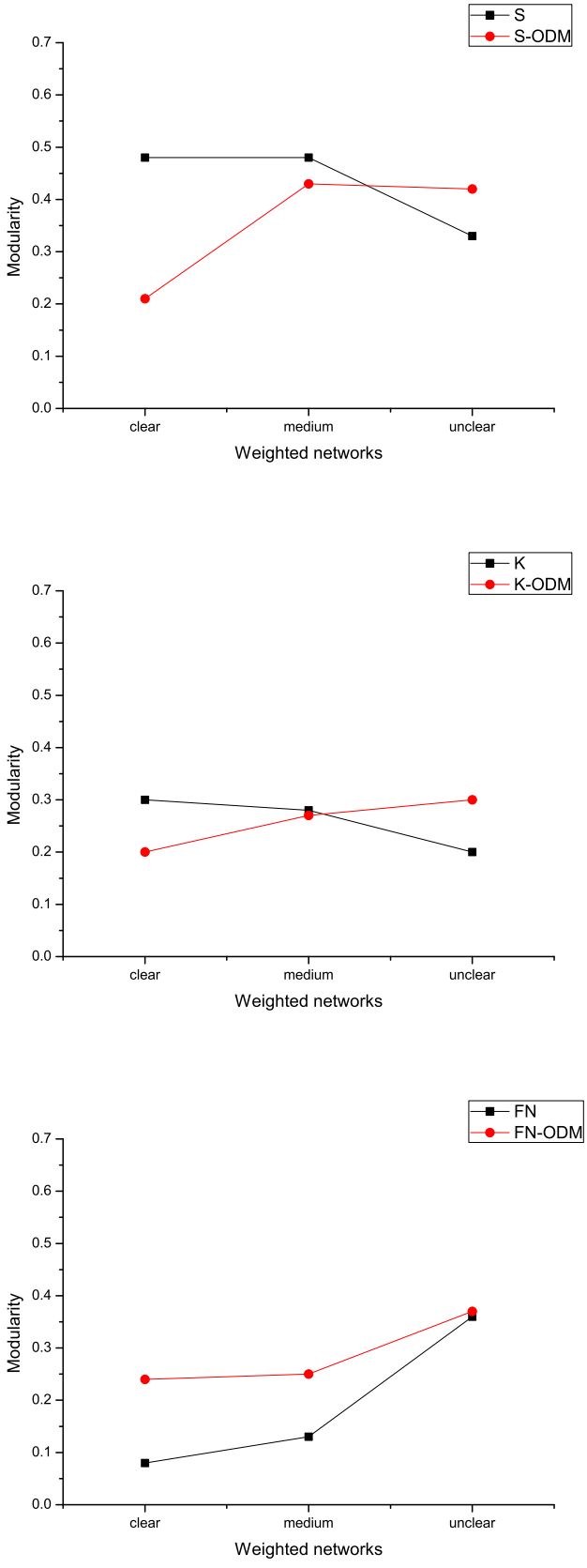


Figure 5.14: Comparisons of modularity on weighted networks

all others as a large cluster.

3. The $K-clustering$, as most of the $K-mode$ algorithms, is time-consuming to find good seeds. However, for the sparse data in RALIC in Section 5.5, it will at least take distance into consideration before making node-sized clusters.

The behavior of the $S-ODM$ group is not affected a lot by the network structure. Since we do not know if there are natural clusters in the RALIC data in Section 5.5, we will apply S and $S-ODM$ to this real-life network to achieve several targets besides the partition in Section 5.6-8.

5.5 RALIC data

RALIC (Replacement Access, Library and ID Card) was the access control system project at University College London (UCL). RALIC was initiated to replace the existing access control systems at UCL based on the prior opinions abstracted from the questionnaires collected from 76 people in UCL [Lim et al., 2010, 2011, Lim and Finkelstein, 2012, Lim and Bentley, 2012, Lim, 2010], see Figure 5.2.

The previous research has already transferred the data in the large set of questionnaires into pure numbers before we build opinion networks based on them. The STAKERARE system referenced in [Lim and Finkelstein, 2012] is a system for collecting project requirements and web-based, importantly for collecting recommendations parties on who they judge to have important stakes in particular aspects of a project. One output of the system is a network of interested parties with connections between

them weighted by the relative number of recommendations they received. The amount and complexity of the information that such a system can generate can imply that an engineer finds it difficult to identify reduced communities of stakeholders whose requirements are most relevant to a particular aspect of a design. Ideally they may want to identify a very small number of experts that they can consult on a particular topic. It may also be helpful if they can identify communities of a particular size or granularity within the data. Besides, this study attempts to dig as much information from the RALIC data as possible:

We collect weight w_{ij} from the database of [Lim, 2010] to represent the recommendation from person i to person j, ranking from 1 to 10, as low to high. The data can be found in Appendix 1-3. The element $A_{ij} = w_{ij}a_{ij}$, where $a_{ij} = 1$ when there is a recommendation from person i to person j. We use the weighted matrix A here apart from the pure adjacency matrix we mentioned before. It is not hard to find the difficulties to partition a network like this with the following properties:

1. It is weighted and directed.

2. The network is sparse.

3. The degree distribution is homogenous.

Therefore, several targets following the previous study are:

1. Compare several clustering methods to test the superiority of *ODM* when dealing with sparse and directed-weighted networks.

2. Develop a systematical method to reveal the hierarchy of the 76-node network layer by layer, from smaller clusters to larger ones.

3. Detect the functional department and the key person responsible for a particular issue in RALIC.

4. Provide opinion pools to represent group opinions, in order to reduce the number of opinions from questionnaires.

5.6 *ODM* in RALIC

In this section, we will apply methods K and $K-ODM$ to the RALIC data. In $K-ODM$, the choice of seeds, evolving time t and opinion difference threshold Th are parameters to impact the accuracy. If k clusters are required in an N-node network, we will list all the possible combinations of k nodes as seeds out of N nodes, in order to find the best seeds. This procedure will be executed for several times with various t and T, followed by the analytical explanation in Section 5.7. The purpose of dynamical partition is to increase the density of the Adjacency Matrix. During the period of the convergence, the D matrix will eventually become all-one matrix. However, the homogeneity of the network will grow at the same time which may also cause the difficult to partition. In any algorithm where a cost function is involved, like in Figure 5.15, the node 7 has the same average distance to cluster with nodes $1-3$ and the other one with 4-6. In any condition like this, a computer simulation program will put node 7 into the cluster $1,2,3$ automatically. The cluster of $1,2,3$ will always have the priority

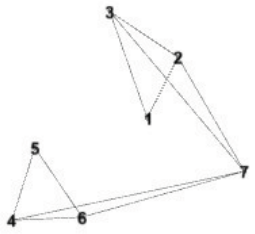


Figure 5.15: Node 7 is looking for a cluster

to obtain nodes like this kind and becomes large.

In case that the irrationality happens, we need to find a D matrix with proper density. Therefore, we test various combinations of T and t. In this study, we only illustrate the results when $T = 0.02,0.2,0.5$. If the system takes $t = t_{sync}$ to synchronize, we take A' matrix at $t = t_{sync}/3, t = 2t_{sync}/3, t = t_{sync}$ respectively and test any combinations of t and T to see the effectiveness of the dynamical method to cluster (see Figure 5.16-18). Since the higher w_{ij} is, the closer two persons i and j are. We use $1/w_{qp}$ instead to calculate the shortest path length.

Meanwhile, the number of value 1 in the 0-1 matrix are recorded, as shown in Figure 5.19.

The accuracy given by the C_N^k times of the partitions with different seeds makes binomial distributions. The $K-ODM$ is more accurate than K. With the concern of unreasonable large clusters, we look into the partitions with high accuracy. In several runs of tests, the same results are illustrated. In $T = 0.02$ group, when $t = 1/3$ and $t = 2/3$, the rare partitions with high ρ_{in} are caused by the same reason shown in Figure

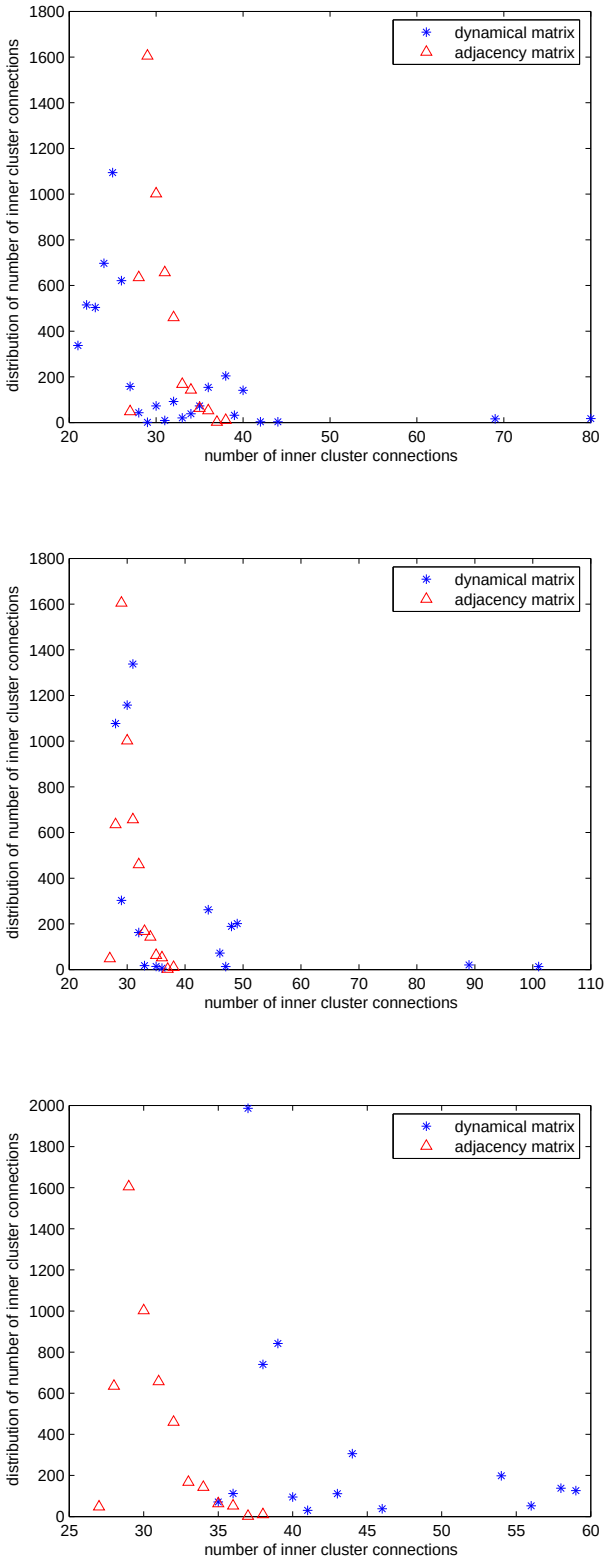
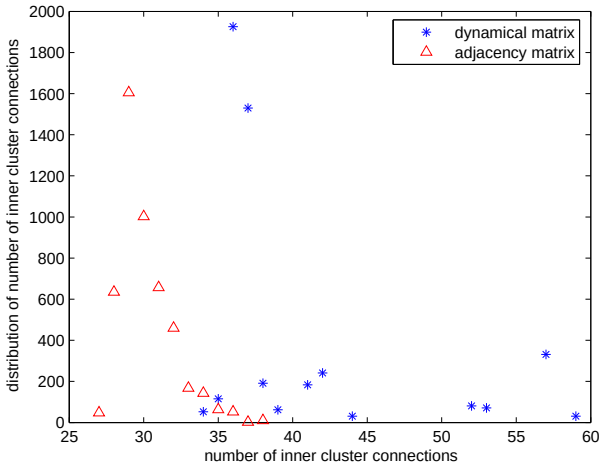
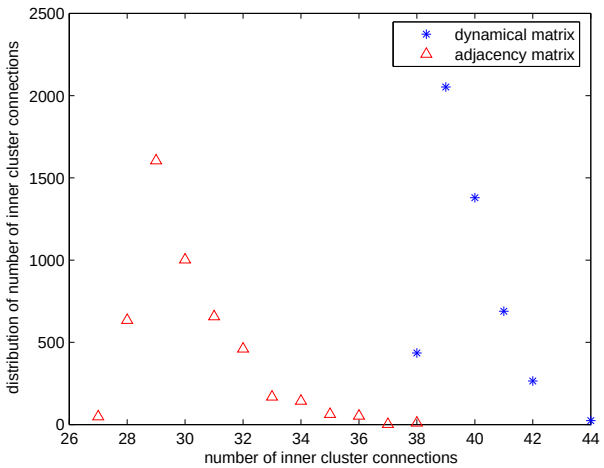


Figure 5.16: when threshold $T = 0.02$





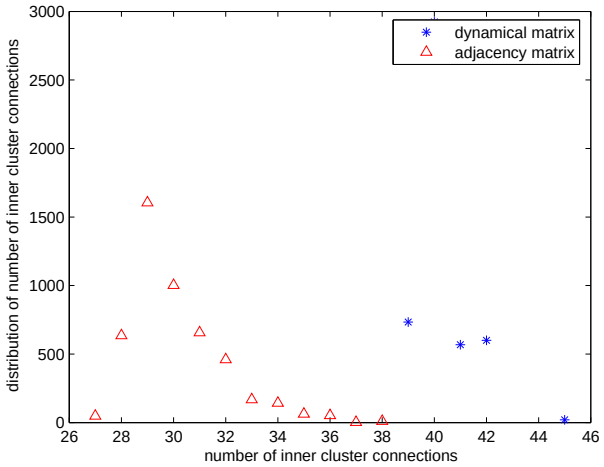
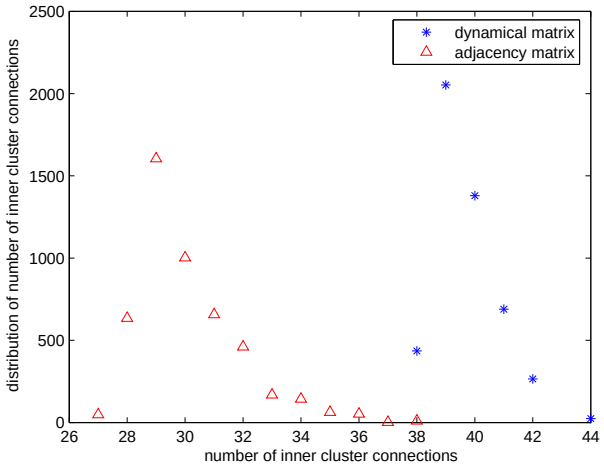
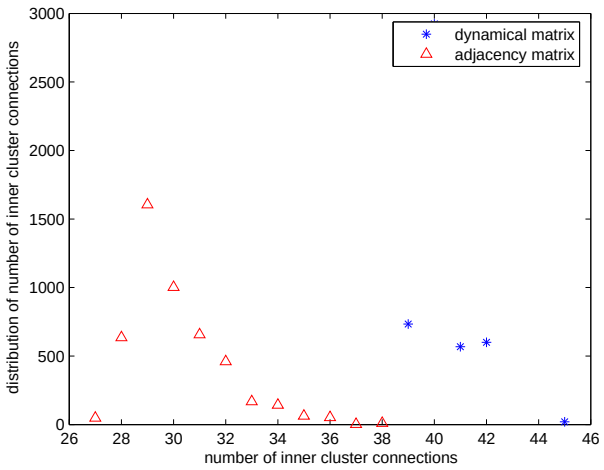


Figure 5.17: when threshold $T = 0.2$





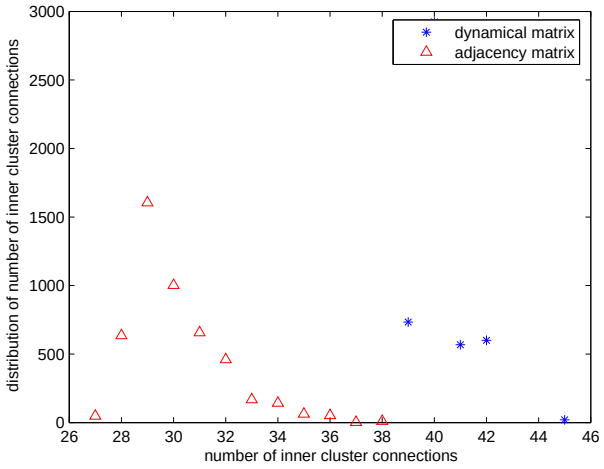


Figure 5.18: when threshold $T = 0.5$

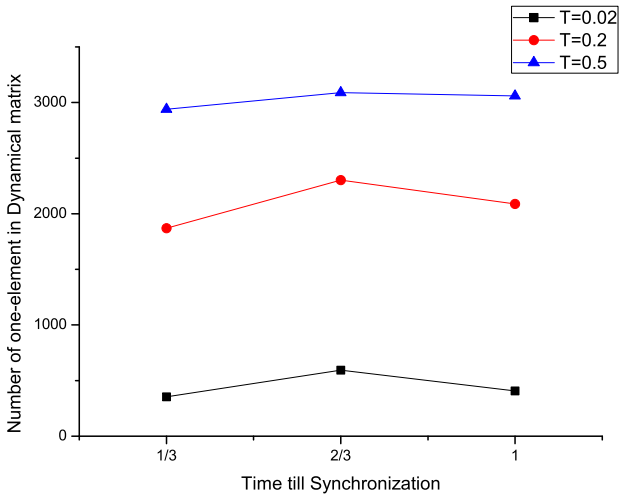


Figure 5.19: One-element in the $76 * 76$ Dynamical $0-1$ Matrix

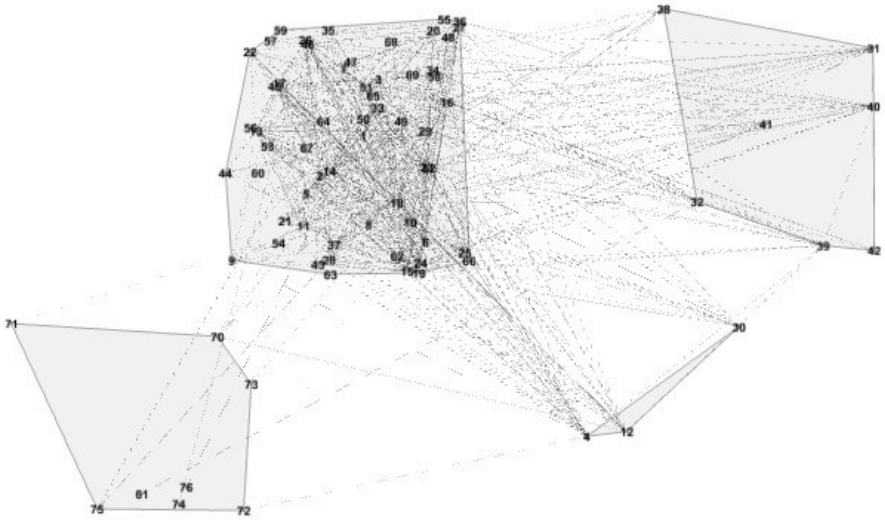


Figure 5.20: An improper partition with high ρ_{in}

5.15. One of the partition with high ρ_{in} is shown in Figure 5.20, in which almost all nodes are put in one cluster and the ρ_{in} are extremely high.

Then, we run the algorithms with $T = 0.02, t = 2/3$ which is supposed to display a

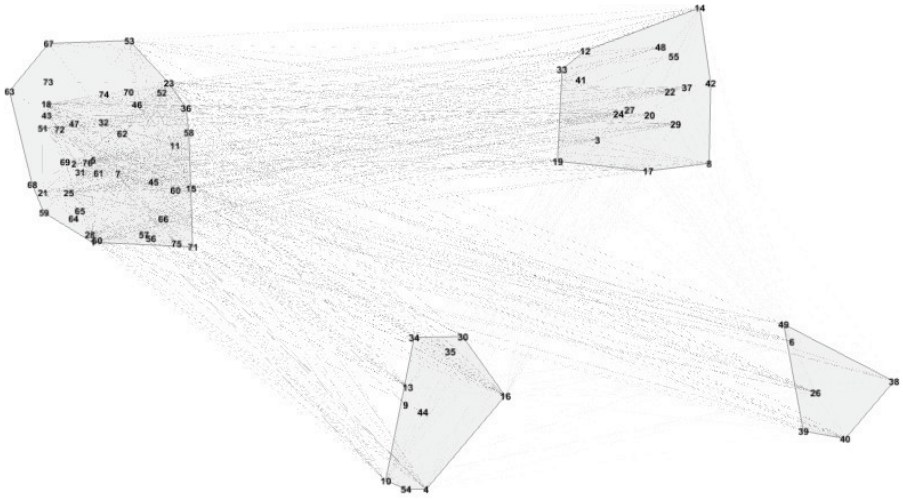


Figure 5.21: Clustering by K-clustering to Dynamical matrix when $T = 0.02$ and $t = 2/3$

good partition. Then we capture one of the network division with highest ρ_{in} as shown in Figure 5.21.

However, the last 26 persons out of 76 ones did not participate in the questionnaires and were brought into the data by one or several recommenders. If the 26 persons were divided before its only friend, the distance from it to all clusters will be zero and the algorithm will automatically put it into the first cluster, which may be wrong. To eliminate the perturbation of the 26 persons, we will introduce an optimization method in Section 5.7 together with another improvement to display hierarchy of the network.

5.7 Improvement of *ODM* method

Among the 76 people in RALIC, only 50 gave and received recommendations. The other 26 did not participate in the questionnaires which left the row of them in the A

matrix blank. As shown in Figure 5.22, we observe the process of the opinion evolving in 76 people and only the 50 of them. The 26 people may cause error when calculating the Dynamical matrix as given in Section 5.6. To avert the perturbation, we optimize the algorithm in Section 5.6 by the following procedures:

1. Generate the adjacency matrix AA of the 50 people.

2. Give every person i of the 50 people random initial opinion x_i.

3. Generate a matrix $DD1$, where $DD1_{ij} = |x_i - x_j|$.

4. As in the right side graph of Figure 17, calculate $d(x_i - x_j)/dt$.

5. Generate a matrix $DD2$, where $DD2_{ij} = |(d(x_i - x_j)/dt)/x_i - x_j|$.

6. Set a threshold T, if any $DD2_{ij} > T$, we can determine that person i and person j are close enough to be set in one cluster.

7. By adjusting T from small values to large ones, we can eventually record clusters layer by layer. Initially, the small clusters with pairwise people first and then larger groups.

8. Put the 26 people one by one to the clusters to which they have the shortest path length (or highest recommending score in the RALIC case).

9. Reset the initial opinions and run the procedure 1-7 for several times to observe the stable clusterings.

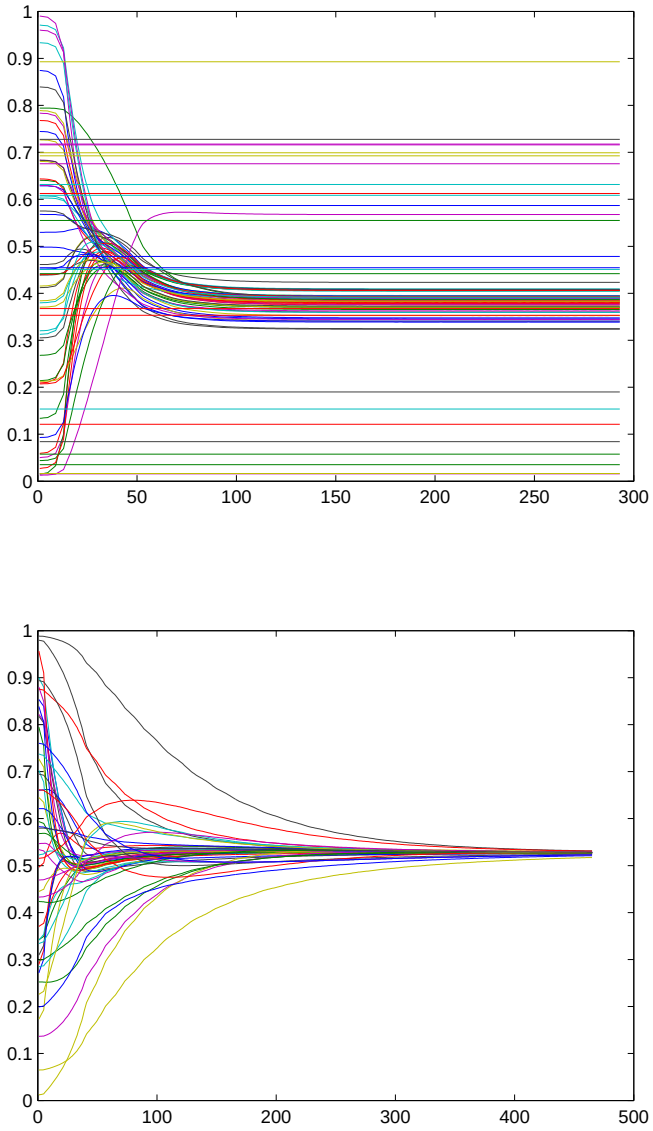


Figure 5.22: Convergence process of 76 people and 50 of them

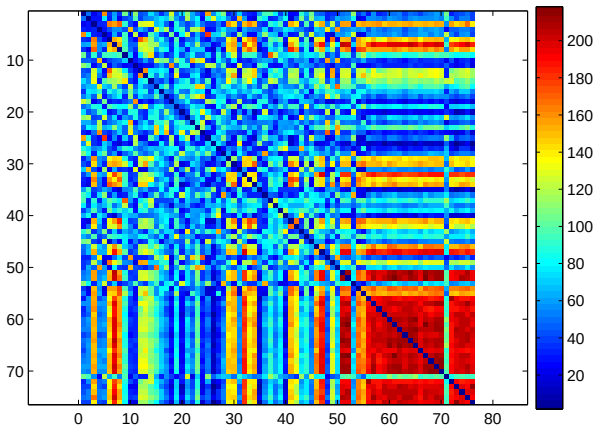


Figure 5.23: The possible speed of local convergence between pairwise nodes

The matrix *DD* to show the possible speed of the local convergence between pairwise nodes is shown in Figure 5.23, by which we can take any algorithms from Section 3 to partition the matrix *DD* instead of the matrix *A* which is too sparse and homogeneous.

In the previous study from Sections 5.4 and 5.6, we partition all the networks into 4 groups. In real life manipulation, various numbers of clustering may be required in one algorithm or software. The optimized algorithm in Section 5.6 makes it possible to detect clusters from small to large. As shown in Figure 5.24, small clusters A, B, C and D have been found respectively before larger clusters $A \bigcup B$ and $C \bigcup D$.

In optimization in this section, if $|(d(x_i - x_j)/dt)/x_i - x_j| > Th$, nodes i and j are connected. Suppose we adjust Th from small values to large ones, more and more pairwise nodes may gain an abstract connection. Consequently, larger clusters will emerge. To maximize the efficiency of the algorithm, two more steps have to be taken:

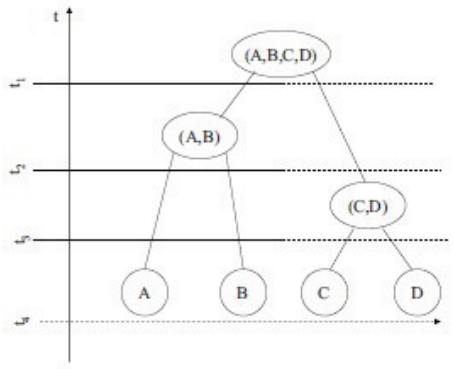


Figure 5.24: From microscale to macroscale

1. Test what the density is to get the best partition for a certain number of clusters.

2. Test how large the T is to give a particular density. For example, when $T \approx \overline{X}$, the density of one-element in an $N * N$ matrix is $1/2 * N^2$, where $\overline{X}$ is the mean of the initial opinions.

We use the third network in Figure 5.12 to test the optimized algorithm and the partitions are shown in Figure 5.25.

The comparison of the inner cluster average distance is shown in Figure 5.26.

5.8 Centroid person responsible for a particular event

Here we give a further thinking in the view of software engineering. We run dynamical partition with various T and t each for several times. The best results for every round are recorded. Figure 5.27 shows the distribution for every node as the centroid. If we use opinions from RALIC data instead of random initial opinions, we can determine the key person for a particular event.

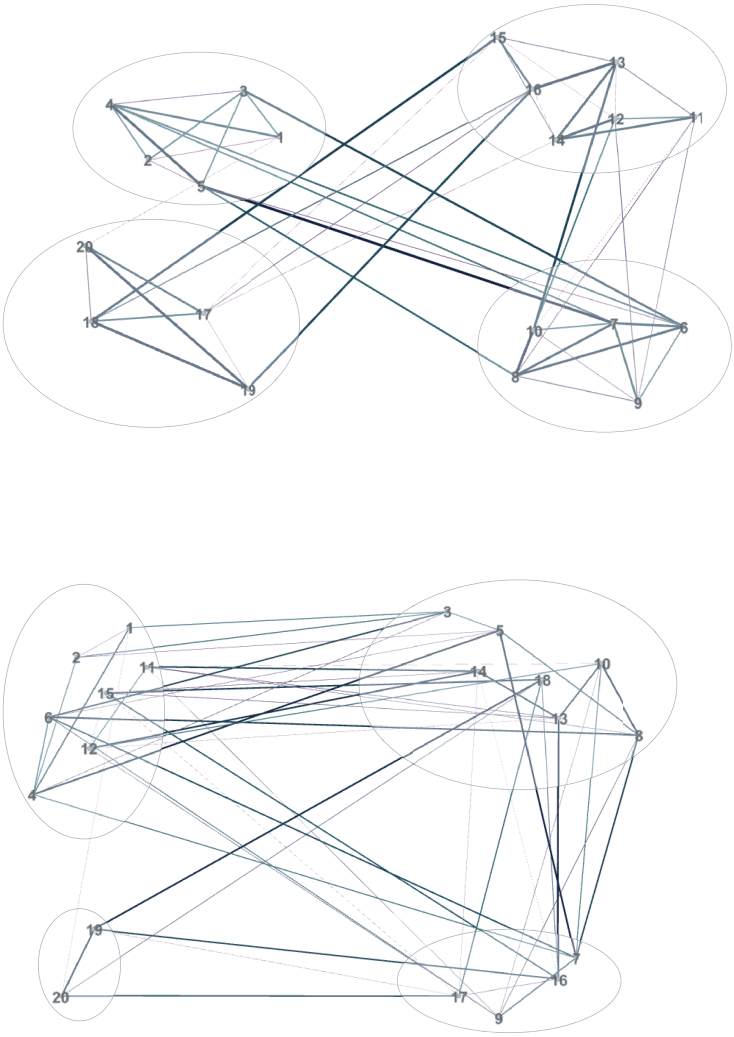


Figure 5.25: Partitions from S and optimized $S - ODM$

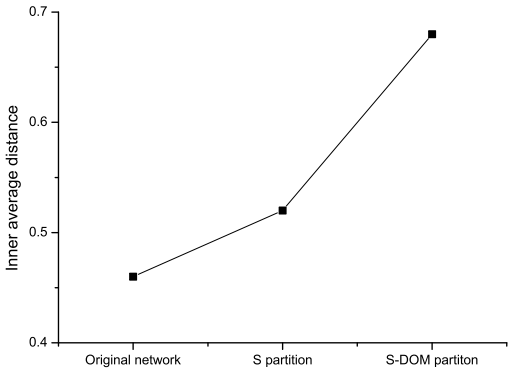


Figure 5.26: The inner cluster average distance

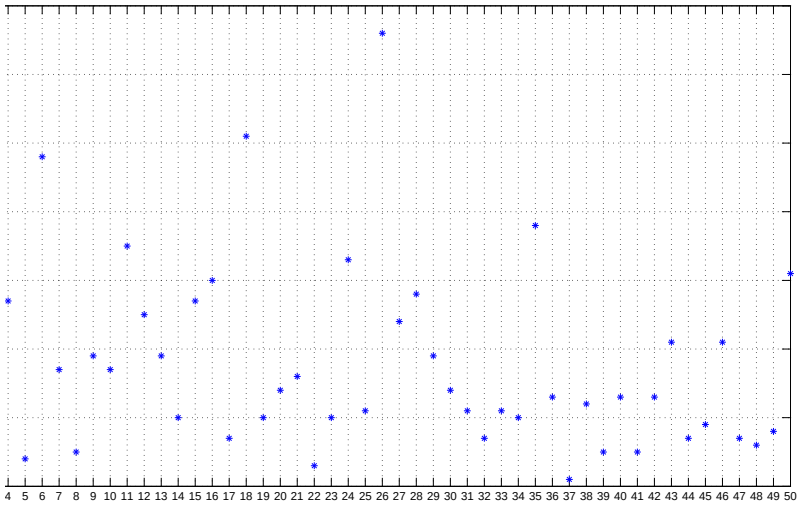


Figure 5.27: The distribution of every node as the best centroid

5.9 Remarks

In this chapter, we have used the linear opinion process to develop an algorithm of network partition. During the opinion evolution towards consensus, some nodes achieve local consensus much earlier than the global consensus. We conclude that these kinds of nodes have closer relations even there is no direct link between them. In this way, we partition the network using the opinion dynamical matrix(ODM) instead of adjacency matrix. The algorithm is tested in benchmark networks and a real network *RALIC* from *UCL*. The partition results are better than what we have obtained from the previous methods based on adjacency matrix.

Chapter 6

Applications of the *ODM* matrix

6.1 Introduction

In Chapter 5, we create a dynamical matrix *ODM*, which illustrates the relations of the nodes underneath. It has been proved that the *ODM* provides clustering structures more clear than the adjacency matrix can do. In this Chapter, we will use the *ODM* instead of adjacency matrix to develop some new algorithms in some other subjects in the area of graph theory. The Chapter is insisting of two parts: the balanced Min-cut based on *ODM* matrix; the supervised feature selection with constrained structured graph optimization. We will use A for short to indicate the *ODM*, which replaces the adjacency matrix.

6.2 The balanced Min-cut based on *ODM* matrix

Clustering is a fundamental research topic in data mining and is widely used for many applications in the field of artificial intelligence, statistics and social sciences. The

objective of clustering is to partition the original data points into a number of groups so that the data points within the same cluster are close to each other while those in different clusters are far away from each other Among various approaches for clustering, K-means and Min-cut are two most popular choices in reality because of their simplicity and effectiveness. The general procedure of traditional K-means (TKM) is to randomly initialize c clustering centers, assign each data point to its nearest cluster and compute a new clustering center iteratively. Some researchers claim that the curse of dimensionality may deteriorate the performance of TKM . A straightforward solution of this problem is to project the original dataset to a low-dimensional subspace by dimensionality reduction, for example, PCA, before performing TKM. Discriminative analysis has been shown effective in enhancing clustering performance. Motivated by this fact, discriminative k-means is proposed to incorporate discriminative analysis and clustering into a single framework to formalize the clustering as a trace maximization problem. By contrast, the min-cut clustering is realized by constructing a weighted undirected graph and then partitioning its vertices into two sets so that the total weight of the set of edges with endpoints in different sets is minimized. Among several graph clustering methods, min-cut tends to provide more balanced clusters as compared to other graph clustering criterion. As the within-cluster similarity in min-cut method is explicitly maximized, solving the Min-cut clustering problem is nontrivial.

In past decades, the methods of clustering based on spectral analysis are well developed. A similarity matrix $\mathbf{A}$ is studied as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the data set. Clustering

of this kind may be done in various ways. They make use of the k eigenvectors corresponding to the k smallest eigenvalues and partition nodes on a graph into k clusters. The algorithm can converge at the globally optimal solution, when the intra-cluster distances are low and inter-cluster distances are high. The algorithms of this kind usually include cluster size in its cost function

$$Sc(C_1,\cdots,C_k) = \sum_{i=1}^{k} \frac{(C_i,\overline{C_i})}{|C_i|}, \tag{6.1}$$

where Sc means spectral clustering and C_i:s are the clusters. Size regularized cut is defined as the sum of the inter-cluster similarity and a regularization term measuring the relative size of two clusters. There is a balancing aiming term in the cost function. There are also application-based solutions in the networking, which aim at network load balancing.

Balancing clustering, in general, is a two-step optimization, in which two aims contradict each other: to minimize the intra-cluster distance and to balance the cluster sizes.

6.2.1 Related work

In balance-constrained clustering, cluster size balance is a mandatory requirement that must be met, and minimizing intra-distance is a secondary criterion. In balanced riven clustering, balance is an aim but not mandatory. It is a compromise between these two goals, namely the balance and the intra-distance. The solution can be a weighted compromise between intra-distance and the balance, or a heuristic that aims at minimizing

intra-distance but indirectly creates a more balanced result than standard k-means.

6.2.2 Methodologies

6.2.2.1 About Min-cut

The principle of Min-cut is original from graph theory. The affinity matrix **A** is built from N data points $\{x_1, \cdots, x_N\}$. The Min-cut graph clustering objective function can be generalized as

$$J = \sum_{1 \leq p < q \leq K} s(C_p, C_q) + s(C_p, C_q) = \sum_{k=1}^{K} s(C_k, \overline{C_k}), \tag{6.2}$$

where K is the number of clusters, C_k is the kth cluster, and $\overline{C_k}$ is the complement of a subset C_k in graph G, and for any set G_1 and G_2

$$s(G_1, G_2) = \sum_{i \in G_1} \sum_{j \in G_2} A_{ij}, \tag{6.3}$$

$$d_i = \sum_j A_{ij}. \tag{6.4}$$

We denote $q_k (k = 1, \cdots, K)$ as the cluster indicators where the ith element of q_k is set to 1 if the ith data point x_i belongs to the kth cluster, and 0 otherwise. For example, if the data points within each cluster are adjacent,

$$q_k = (0, \cdots, 0, 1, \cdots, 1, 0, \cdots, 0)^T. \tag{6.5}$$

After simple mathematical deduction, we can find that

$$s(C_k, \overline{C_k}) = \sum_{i \in C_k} \sum_{j \in \overline{C_k}} A_{ij} = q_k T (D - A) q_k, \tag{6.6}$$

$$\sum_{i \in C_k} d_i = q_k T D q_k, \tag{6.7}$$

$$s(C_k, C_k) = q_k T A q_k, \tag{6.8}$$

where D is a diagonal matrix with the ith diagonal element as d_i. The objective function of Min-cut method can therefore be reformulated as:

$$J = \sum_{k=1}^{K} q_k T (D - A) q_k. \tag{6.9}$$

Min-Cut clustering has been applied in various applications. However, none of the existing work on Min-cut is capable of balanced clustering when necessary, which shall be addressed by our newly proposed balanced min-cut algorithm.

6.2.2.2 About Exclusive Lasso

In this paper we will study a cluster indicator matrix F, which is defined as

$$f_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are in the same class,} \\ 0 & \text{otherwise.} \end{cases} \tag{6.10}$$

If F is an indicator matrix, $F \in R^{N*d}$ and $F \in Ind$, it will look like

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix} \tag{6.11}$$

Here we introduce a regularizer that controls the complexity of nodes in different clusters. We assume a competitive nature among the objects shared by all the clusters, i.e, if a node is assigned to the kth cluster, it is reasonable that the value of the same node in other clusters are zeros. To the end, we introduce the following regularizer

$$\|F\|_e = \sqrt{\sum_{j=1}^{c} (\sum_{i=1}^{n} |f_{ij}|)^2}, \tag{6.12}$$

which illustrate the balance of the clustering result.

The regularizer introduces an l_1-norm to combine the weights for the same category used by different data points and an l_2-norm to combine the weights of different categories. Since l_1-norm tends to achieve a sparse solution, the construction in the exclusive lasso essentially introduces a competition among different categories for the same data points. In our work, the exclusive lasso is used as a balance constraint. We will prove that the value of exclusive lasso indicates the balance degree of our clustering algorithms.

6.2.2.3 About Augmented Lagrange Multiplier Method

ALM method may be called as Method of Multiplier or Primal-dual Method. If only consider Lagrangian functional only for equality constraints

$$L(x) = f(x) + \lambda_T h(x), \tag{6.13}$$

then for a Lagrange multiplier vector λ^*, suppose that there is an optimum x^* for the following unconstrained optimization problem

$$\min_x L(x, \lambda^*). \tag{6.14}$$

If x^* satisfy all the equality constrains $h(x^*) = 0$ in the original design problem, x^* is an optimum for the original optimization problem and λ^* is a Lagrange multiplier optimum. Consequently, the original optimization problem can be transformed into

the following problem that have the same optimum x^* and λ^*

$$\min_x L(x,\lambda), \tag{6.15}$$

subject to $h_i(x) = 0, i = 1,2,\cdots,l$.

In order to avoid the unboundness of Lagrangian, a penalty function is introduced. We call it as augmented Lagrangian

$$A(x,\lambda,r) = L(x,\lambda) + \frac{1}{2}\sum_{i=1}^{l} r_i h_i(x)^2, \tag{6.16}$$

where r_i is the penalty parameter for the ith equality constraint. In the *ALM* method, the unconstrained optimization tool sequentially minimize the augmented Lagrangian for the given value of r_i and λ_i. Then, these two parameters are modified to satisfy the optimality condition.

6.2.3 Clustering based on balanced min-cut

6.2.3.1 Balance constraint

Given F in Equation (6.12) as the cluster indicator matrix, the exclusive Lasso of F can be rewritten as:

$$\|F\|_e = Tr(F^T \mathbf{1}\mathbf{1}^T F). \tag{6.17}$$

From this equation, we can observe that the value of exclusive lasso equals the square-sum of the number of data points in each class. In the following, we prove that the most balanced clustering can be achieved by minimizing the exclusive lasso.

Theorem1. Given $n_1 + n_2 + \cdots + n_k = N$, $n_i|_{i=1}^k \geq 0$, $\sum_{i=1}^k n_i^2$ arrives at its minimum when $n_i = N/k$.

Proof. According to the Cauchy inequality, we have

$$(a_1^2 + a_2^2 + \cdots + a_k^2)(b_1^2 + b_2^2 + \cdots + b_k^2) \geq (a_1 b_1 + a_2 b_2 + \cdots + a_k b_k)^2. \quad (6.18)$$

Let $b_i|_{i=1}^k = 1$, the equality hold when $n_1 = n_2 = \cdots = n_k$. Hence , we can easily have the conclusion that when $n_i = N/k$, $\sum_{i=1}^N n_i^2$ achieves the minimal value.

According to the above theorem, by minimizing the exclusive lasso, each cluster will have n/c data points. The most balanced clustering result is thus obtained. Hence, we use the the exclusive lasso as the balance constraint.

6.2.3.2 Balanced Min-cut

We similarly aim to cluster n data points $X = x_1, \cdots, x_n \in \mathbb{R}^{d*n}$ into K clusters. To begin with, we use the Gaussian function to construct an affinity matrix A, which is

defined as:

$$A_{ij} = \begin{cases} e^{(-\frac{\|x_i - x_j\|^2}{\delta^2})}, \\ x_i \text{ and } x_j \text{ are } k \text{ nearest neighbours} \\ 0, \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{6.19}$$

where δ is is utilized to control the spread of neighbors.Given the weight matrix A and the cluster indicator matrix F, the objective function of min-cut graph clustering is formulated as follows

$$\min_{F \in Ind} \mathbf{1}^T A \mathbf{1} - Tr(F^T A F). \tag{6.20}$$

We further incorporate the exclusive lasso into min-cut and get the following objective function

$$\max_{F \in Ind} Tr(F^T A F) - \gamma \|F\|_e, \tag{6.21}$$

which can be rewritten as

$$\max_{F \in Ind} Tr(F^T A F) - Tr(F^T \gamma \mathbf{1}\mathbf{1}^T F), \tag{6.22}$$

The problem is equivalent to

$$\min_{F \in Ind} TrF^T(\gamma \mathbf{1}\mathbf{1}^T - A)F. \tag{6.23}$$

Here we learn a matrix G, with the constraint $F = G$ the function becomes:

$$\min_{F \in Ind, G, F=G} Tr(F^T(\gamma \mathbf{1}\mathbf{1}^T - A)G). \tag{6.24}$$

During the optimization, it is one of the target to make G close to F. With a simple mathematical deduction, the objective function is rewritten as

$$\min_{F \in Ind, G} Tr(F^T(\gamma \mathbf{1}\mathbf{1}^T - A)G) + \frac{\mu}{2}\|F - G + \frac{1}{\mu}\Lambda\|_F^2. \tag{6.25}$$

During the optimization, we will have to update G and F iteratively. When fixing F, the Lagrangian function of G problem (6.25) is

$$\mathcal{L}(G, \mu) = Tr(F^T(\gamma \mathbf{1}\mathbf{1}^T - A)G) + \frac{\mu}{2}\|F - G + \frac{1}{\mu}\Lambda\|_F^2. \tag{6.26}$$

Taking the derivative of $\mathcal{L}(G, \mu)$ and setting it to zero we have

$$\frac{\partial \mathcal{L}(G, \mu)}{\partial G} = F^T(\gamma \mathbf{1}\mathbf{1}^T - A) - \mu(F - G) - \Lambda = 0. \tag{6.27}$$

Hence we have:

$$G = \frac{1}{\mu}(\Lambda - F^T(\gamma \mathbf{1}\mathbf{1}^T - A)) + F. \tag{6.28}$$

The initialization and update of G can be solved by equation (6.28). Similarly, we can update F in this way

$$\frac{\partial \mathcal{L}(F,\mu)}{\partial F} = (\gamma \mathbf{1}\mathbf{1}^T - A)G + \mu(F - G) + \Lambda = 0, \tag{6.29}$$

which leads to

$$F = -\frac{1}{\mu}(\Lambda + (\gamma \mathbf{1}\mathbf{1}^T - A)G) + G. \tag{6.30}$$

The detail of the algorithm is described in Algorithm 1:

1. **Input** Data matrix $X \in \mathcal{R}^{n \times d}$, a large enough μ, a regulation parameter γ, Λ, δ, ρ.
2. **Output** Indicator matrix F
3. Initialize A, $F \in Ind$, δ.
4. Update G by solving Equation (6.28).
5. Update $F \in Ind$ by solving Equation (6.30).
6. Update $\Lambda = \Lambda + \mu(F - G)$, $\mu = \rho\mu$.

7. Repeat until convergence.

6.2.4 Experiments

We establish a set of synthetic data which called 'two moon' as in Figure 6.1. If we use the traditional Min-cut to partition it, the initial point for searching will determine the final partition, which means the method is instable. In our new balanced Min-cut, we do not need to find any initial points and the solution of the partition is stable and effective. Some initial values of the parameters are as follows:

1. X will be attached;
2. $\mu = 10^8$
3. $\gamma = 10^{-6}, 10^{-4}, 10^{-2}, 10^0, 10^2, 10^4, 10^6$, try everyone respectively
4. $\rho = 1.5$
5. Λ is a positive-definite matrix, at present we don't have a clue to determine it, so just try all one matrix $\Lambda \in R^{N*k}$
6. A is calculated by equation (19)
7. $F \in Ind$ means that $F \in R^{N*k}$ is an indicator matrix, we will set it all one initially.
8. δ welllet's put $\delta = 1$ at present.

In Figure 6.1 we can see the result of the clustering, which is accurate. Some further experiments will be taken on different kinds of graph data.

(a) ASSF (b) DSSF

Figure 6.1: The clustering for the synthetic data set 'two moon'.

6.3 Supervised feature selection with constrained structured graph optimization

Feature selection is an important subject in the area of data learning. This kind of techniques have been proposed to select the most relevant subsets of features according to a particular demand [Saeys et al., 2007, Chandrashekar and Sahin, 2014, Huang, 2015]. High-dimensional data present a big challenge to feature selection due to the "curses of high-dimensionality" . The new task is involving many irrelevant and redundant features [Liu and Yu, 2005]. As a typical application, the selection of gene features will be serve us throughout the introduction and the experiment in this paper. In this area, a typical task is to separate healthy patients from cancer patients based on their gene features. Usually, fewer than 100 patients are available for training, but the number of features which may relevant to the cancer ranges from 6000 to 60000. It has been stated that the objectives of feature selection are manifold. The effective method of feature selection is to express the structure of high-dimensional data by a low-dimensional manifold. The methods from various of areas focus on the different benefits of feature

selection: understanding of data, reducing the measurement, defying the curse of dimensionality to improve prediction performance, etc. Among them, feature ranking is a type of popular feature selection method which computes the degrees of dependency of individual features with respect to class and select features according to the degrees. Typical criteria to evaluate the degrees of dependency include the measures of correlation between the feature and the class, or the uncertainty measures used in information theory. However, such methods are most effective for statistically independent features, but have low ability in identifying group features that can be used to predict the class. Recently, classifiers are used to rank features. For example, Relief algorithm, support vector machine (SVM) were used to rank features [Inza et al., 2000, Chang et al., 2008]. However, such methods are computationally intensive. Moreover, the above-mentioned strategies for acquiring feature subsets could be biased toward the highest ranking feature, as the feature with the highest performance will be selected first in the subset. However, low-rank features, if selected in an appropriate subset, could provide better classification performance.

In the view of data, if all the data instances in the data set have class label, the process of feature selection is called "supervised". If some of the data instances have class label and the others do not, it is called "semi-supervised". If none of the data instances have class label, it is called "unsupervised". The depth of treatment of various subjects reflects the proportion of papers covering them: the problem of supervised learning is treated more extensively than that of the other two problems. The supervised feature selection has plenty of potential benefits due to the rich information of the data struc-

ture. Most algorithms of feature selections target at minimize the feature redundancy. Feature redundancy is understood in terms of feature dependency, which measures the correlation between features. It is widely accepted that two perfectly correlated features are redundant to each other since adding one on top of the other will not increase information nor improve model accuracy. However, some recent research indicated that noise reduction and class separation may be obtained by adding variables that are presumably redundancy. Hence, the supervised feature selection is able to make full use of the class label to avoid the problem.

From the previous studies, it appears that the objective of finding a single feature subset that can produce a model with the highest accuracy when evaluated using available data is overly emphasized in current feature selection research. The applications from various areas may call for different objectives, which in term require different approaches for feature selection. The challenge now is how to develop a unified method based on which different structures of data can be analyzed.

To mitigate the impact of the above problem, we proposed a supervised feature selection method with constrained structured graph optimization (CGFR). It is worthwhile to highlight the main contributions of the papers as follows:

1. A dynamical process is proposed to train the similarity matrix **S**, which adaptively learns local manifold structure, and thus can select more valuable features.

2. The optimization of **S** towards the affinity matrix **A** is provided. It is highly possible for the small class to be buried in big class as a sacrifice to achieve the

lower redundance between features. Hence, it is worthwhile to maximize the correction of the real data structure, which means **S** approximates the initial **A**.

3. A low-dimensional manifold is investigated to express the original data with high dimension.

4. Both L1-norm and L2-norm are proposed to form a more structured regulation. The L1-norm tends to give sparse solution.

6.3.1 Related Work

Feature selection, also called as variable selection, is a process to determine the "best" subset of features for prediction. The concept itself can be traced back to 1940's

The feature selection methods can be categorized into tree types:1) Feature ranking method which computes the degrees of dependency of individual features with respect to class and select features according to the degrees, 2) Feature subset method which directly select subset of features which are relevant to the class, and 3) Embedded method which incorporate the feature selection as part of training process. Among these methods, feature ranking is a type of popular feature selection method which computes the degrees of dependency of individual features with respect to class and select features according to the degrees.

A simple way to rank features are based on pair-wise dependency analysis of individual features. Such methods evaluate the degree of dependency between each feature and the class, one feature at a time. Typical criteria to evaluate pair-wise dependency include the measures of correlation between the feature and the class, e.g., Pearsons

product-moment correlation coefficient. Information gain is a another type of popular criteria which measures the reduction of uncertainty about the class with a given feature [Quinlan, 1986]. However, such methods are most effective for statistically independent features, but have low ability in identifying group features that can be used to predict the class.

Recently, classifiers are used to evaluate the features. For example, Relief algorithm was used to calculate relevance weights for all features simultaneously by looking into their joint relationship with the class

6.3.2 Notations and Definitions

We summarize the notations and the definition of norms used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $M = (m_{ij})$, its i-th row is denoted as $\mathbf{m}^i$, and its j-th column is denoted by $\mathbf{m}_j$. The Frobenius norm of the matrix $M \in \mathcal{R}^{n \times m}$ is defined as

$$\mathbf{M}_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} m_{ij}^2}.s \tag{6.31}$$

6.3.3 Supervised feature selection with constrained structured graph optimization

6.3.3.1 Constrained structured graph optimization

Inspired by the development of spectral analysis and manifold learning, many supervised feature selection methods try to use local manifold structure in order to get better performance. In such methods, similarity matrix is crucial for the ultimate performance of spectral methods. Nevertheless, most methods construct similarity matrix simply from original features which contain many redundant and noise samples or features. This will inevitably damage the learned structure, and the feature selection result is surely unreliable and inaccurate. Thus, in this paper, we apply an adaptive process to determine the similarity matrix with probabilistic neighbors through the algorithm [Nie et al., 2014].

Let $\mathbf{X} \in \mathcal{R}^{d\times n}$ be a data set with n objects $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, where $\mathbf{x_i} \in \mathcal{R}^{d\times 1}$. $\mathbf{X}$ is associated with class labels $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_n\}$, and let c be the number of classes. Let $\{\mathbf{W}_1, ..., \mathbf{W}_d\}$ be d features of $\mathbf{X}$, the supervised feature selection is to use both $\mathbf{X}$ and $\mathbf{Y}$ to rank $\mathbf{W}$. We want to build a matrix $\mathbf{S} \in \mathcal{R}^{n\times n}$, which is defined as the probability for x_i to connect to x_j. The probability of two objects to be neighbours can be regarded as the similarity between them. Intuitively, closer samples are likely to have larger probability to connect, thus s_{ij} is inversely proportional to the distance between x_i and x_j. In this paper, we use the square of Euclidean distance for simplicity, i.e., $\mathbf{x}_i - \mathbf{x}_j{}_2^2$. The corresponding Laplacian matrix of $\mathbf{S}$ is $\mathbf{L}_s = \mathbf{D}_s - \frac{\mathbf{S}^T+\mathbf{S}}{2}$, where the degree matrix

$\mathbf{D}_s \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the i-th diagonal element as $\sum_j (s_{ij} + s_{ji})/2$.

In the meanwhile, we also wish that the learned probabilities $\mathbf{S}$ is consistent to the class label $\mathbf{Y}$. To effectively utilize the class label information during feature selection, we translate the class labels $\mathbf{Y}$ into a affiliation matrix, $\mathbf{A}$, which represents the relationships among objects and is defined as

$$a_{ij} = \begin{cases} \frac{1}{|\mathbf{C}_l|} & \text{if } x_i \in \mathbf{C}_l \text{ and } x_j \in \mathbf{C}_l, \\ 0 & \text{otherwise.} \end{cases} \tag{6.32}$$

Here, the similarity between two objects in the same class is set as the reciprocal of size of corresponding class. The objects in smaller class will be assigned with higher similarity than those in bigger class. As such, the small class will be emphasized in order to not be buried in big class. This can effectively solve the imbalanced problem. We wish to learn $\mathbf{S}$ that best approximates the affiliation matrix $\mathbf{A}$.

Therefore, we can learn $\mathbf{S}$ by simultaneously minimize the product of $\mathbf{S}$ and the distance between objects, and the difference between $\mathbf{A}$ and $\mathbf{S}$. Considering two different distances, the L2-norm and the L1-norm, between the given affinity matrix $\mathbf{A}$ and the learned similarity matrix $\mathbf{S}$, we define the Constrained Graph Rank (CGFR) as the

solution to the following two optimization problems

$$
\begin{aligned}
&J_{CGFR-L2} = \min \sum_{i=1}^{n} (\sum_{j=1}^{n} \|x_i - x_j\|_2^2 s_{ij} + \gamma \|s_i - a_i\|_2^2), \\
&s.t. \forall i,\ s_i^T \mathbf{1} = 1,\ s_i \geq 0, rank(L_S) = n - c,
\end{aligned}
\tag{6.33}
$$

$$
\begin{aligned}
&J_{CGFR-L1} = \min \sum_{i=1}^{n} (\sum_{j=1}^{n} \|x_i - x_j\|_2^2 s_{ij} + \gamma \|s_i - a_i\|_1), \\
&s.t. \forall i,\ s_i^T \mathbf{1} = 1,\ s_i \geq 0, rank(L_S) = n - c,
\end{aligned}
\tag{6.34}
$$

where $\mathbf{S}_t = \mathbf{X}^T \mathbf{H} \mathbf{X}$ is the total scatter matrix, and $\mathbf{H}$ is the centering matrix defined as $\mathbf{H} = 1 - \frac{1}{n} \mathbf{1}\mathbf{1}^T$. The constraint $\mathbf{W}^T \mathbf{S}_t \mathbf{W} = I$ is used to force the data on the subspace are statistically uncorrelated. The rank constraint $rank(\mathbf{L}_A) = n - c$ is imposed to $\mathbf{L}_S$, such that the sparse graph constructed from $\mathbf{S}$ only consists of c connected components. γ is the balance parameter used to balance the first and the second term. We show how to determine γ later.

It is difficult to solve problems (6.33) and (6.34), since $\mathbf{L}_S = \mathbf{D}_S - \frac{\mathbf{S}^T + \mathbf{S}}{2}$ and $\mathbf{D}_S$ both depend on $\mathbf{S}$, and the rank constraint $rank(\mathbf{L}_S) = n - c$ is a complex nonlinear constraint. Fortunately, Nie et al. have proved that $rank(\mathbf{L}_S) = n - c$ is equivalent to

the following problem [Nie et al., 2014]

$$\min_{\mathbf{F}\in R^{n\times c},\mathbf{F}^T\mathbf{F}=I} 2\mu Tr(\mathbf{F}^T\mathbf{L}_S\mathbf{F}), \tag{6.35}$$

where μ is a large enough parameter.

Then problems (6.33) and (6.34) are equivalent to the following two problems

$$\begin{aligned} J_{CGFR-L2} = &\min[\sum_{i=1}^{n}(\sum_{j=1}^{n}\|x_i-x_j\|_2^2 s_{ij}+\gamma\|s_i-a_i\|_2^2) \\ &+2\mu Tr(\mathbf{F}^T\mathbf{L}_S\mathbf{F})], \\ &s.t.\forall i,\ s_i^T\mathbf{1}=1,\ s_i\geq 0,\mathbf{F}\in R^{n\times c},\mathbf{F}^T\mathbf{F}=\mathbf{I}, \end{aligned} \tag{6.36}$$

$$\begin{aligned} J_{CGFR-L1} = &\min[\sum_{i=1}^{n}(\sum_{j=1}^{n}\|x_i-x_j\|_2^2 s_{ij}+\gamma\|s_i-a_i\|_1) \\ &+2\mu Tr(\mathbf{F}^T\mathbf{L}_S\mathbf{F})], \\ &s.t.\forall i,\ s_i^T\mathbf{1}=1,\ s_i\geq 0,\mathbf{F}\in R^{n\times c},\mathbf{F}^T\mathbf{F}=\mathbf{I}. \end{aligned} \tag{6.37}$$

6.3.3.2 Constrained structured graph optimization feature selection

According to the theory of manifold learning, there always exists a low-dimensional manifold that can express the structure of high-dimensional data. In this paper, we aim at finding a linear combination of original features to best approximate the low-dimension manifold. Denote $\mathbf{XW}$ as this linear combination, where $\mathbf{W}\in\mathcal{R}^{d\times m}$ is the

projection matrix, m is the projection dimension. Then we rewrite problems (6.36) and (6.37) as the following two problems

$$\begin{aligned} J_{CG_L2} = &\min[\sum_{i=1}^{n}(\sum_{j=1}^{n}\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij} + \gamma\|s_i - a_i\|_2^2) \\ &+ 2\mu Tr(\mathbf{F}^T \mathbf{L}_S \mathbf{F})], \\ &s.t.\ \forall i,\ s_i^T \mathbf{1} = 1,\ s_i \geq 0, \mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{W}^T\mathbf{S}_t\mathbf{W} = \mathbf{I}, \end{aligned} \tag{6.38}$$

$$\begin{aligned} J_{CG_L1} = &\min[\sum_{i=1}^{n}(\sum_{j=1}^{n}\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij} + \gamma\|s_i - a_i\|_1) \\ &+ 2\mu Tr(\mathbf{F}^T \mathbf{L}_S \mathbf{F})], \\ &s.t.\ \forall i,\ s_i^T \mathbf{1} = 1,\ s_i \geq 0, \mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{W}^T\mathbf{S}_t\mathbf{W} = \mathbf{I}. \end{aligned} \tag{6.39}$$

The importance of d features can be ranked according to $\{w^1{}_2^2, ..., w^d{}_2^2\}$. The most important k features can be selected by the sorted $\mathbf{w}^j{}_2^2$, where k is the number of features that need to be selected. $\mathbf{W}$ is used for selecting features and $\mathbf{S}$ is used to capture local structure, thus the proposed approach performs feature selection and local structure learning simultaneously.

6.3.4 Optimization Algorithm for CGFR-L2

In this section, we propose an effective algorithm to solve problem (6.38).

6.3.4.1 Updating $\mathbf{S}$ for fixed $\mathbf{W}$ and $\mathbf{F}$

Note that the problem (6.38) is independent between different i, so we can solve the following problem individually for each i

$$\begin{aligned} &\min(\sum_{j=1}^{n} \|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij} + \gamma \|s_i - a_i\|_2^2 \\ &\quad + 2\lambda \sum_j f_i - f_{j_2}^2 s_{ij}), \\ &s.t.\ s_i^T \mathbf{1} = 1,\ s_i \geq 0. \end{aligned} \tag{6.40}$$

Denote $\mathbf{d}_i \in \mathcal{R}^{n\times 1}$, where $d_{ij} = \mathbf{W}^T x_i - \mathbf{W}^T x_{j_2}^2 - 2\gamma a_{ij} + 2\lambda f_i - f_{j_2}^2$, problem (6.42) can be written in vector form

$$\begin{aligned} &\min \left\| \mathbf{s}_i + \frac{\mathbf{d}_i}{\gamma} \right\|_2^2, \\ &s.t.\ s_i^T \mathbf{1} = 1,\ s_i \geq 0, \end{aligned} \tag{6.41}$$

where $\mathbf{s}_i \in \mathcal{R}^{n\times 1}$ is a vector with the j-th element as s_{ij}.

The Lagrangian function of problem (6.42) is

$$\mathcal{L}(s_i, \lambda, \beta_i) = \left\| s_i + \frac{\mathbf{d}_i}{\gamma} \right\|_2^2 - \lambda(s_i^T \mathbf{1} - 1) - \beta_i^T s_i, \tag{6.42}$$

where λ and $\beta_i > 0$ are the Lagrangian multipliers.

Note that $\beta_i = 0$ according to KKT condition [Boyd and Vandenberghe, 2004], then

it can be verified that the optimal solution s_i should be

$$\mathbf{s}_i = (-\frac{\mathbf{d}_i}{\gamma} + \lambda)_+. \tag{6.43}$$

We define the following function w.r.t. λ

$$\mathbf{h}_i = \sum_{j=1}^{n} (-\frac{\mathbf{d}_i}{\gamma} + \lambda)_+ - 1. \tag{6.44}$$

According to the constraint $\mathbf{s}_i^T \mathbf{1} = 1$, we have

$$\mathbf{h}_i = 0. \tag{6.45}$$

Therefore, the value of η is the root of function $\mathbf{h}_i$. Note that $\mathbf{h}_i$ is a piecewise linear and monotonically increasing function, thus the root can be easily obtained by Newtons method. After computing η, the optimal solution to the problem (6.42) can be obtained by Equation (6.43).

6.3.4.2 Updating $\mathbf{F}$ with fixed $\mathbf{W}$ and $\mathbf{S}$

When S is fixed, problem (6.38) becomes

$$\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \mu Tr(\mathbf{F}^T \mathbf{L}_S \mathbf{F}). \tag{6.46}$$

It is obvious that the optimal solution of F is formed by the c eigenvectors of L_s corresponding to the c smallest eigenvalues, which are determined by S.

6.3.4.3 Updating $\mathbf{W}$ with fixed $\mathbf{S}$ and $\mathbf{F}$

It can be verified that the above problem can be rewritten as

$$\min_{\mathbf{W}^T\mathbf{S}_t\mathbf{W}=\mathbf{I}} Tr(\mathbf{W}^T\mathbf{X}^T\mathbf{L}_S\mathbf{X}\mathbf{W}), \tag{6.47}$$

Let $\widehat{\mathbf{W}^T} = \mathbf{W}^T\mathbf{S}_t^{1/2}$, we have $\widehat{\mathbf{W}} = (\mathbf{S}_t^{1/2})^T\mathbf{W}$. Then problem (6.47) becomes

$$\begin{aligned} &\min Tr(\widehat{\mathbf{W}^T}(\mathbf{S}_t^{1/2})^{-1}\mathbf{X}^T\mathbf{L}_S\mathbf{X}((\mathbf{S}_t^{1/2})^T)^{-1}\widehat{\mathbf{W}}), \\ &s.t.\ \forall i,\ s_i^T\mathbf{1} = 1,\ s_i \in [0,1], \widehat{\mathbf{W}^T}\widehat{\mathbf{W}} = \mathbf{I}. \end{aligned} \tag{6.48}$$

The optimal solution $\mathbf{W}$ to the above problem is formed by the k eigenvectors of $\mathbf{S}_t^{-1}\mathbf{X}^T\mathbf{L}_S\mathbf{X}$ corresponding to its m smallest eigenvalues (we assume the null space of the data $\mathbf{X}$ is removed, i.e., $\mathbf{S}_t$ is invertible).

6.3.4.4 CSFG-L2 algorithm

We summarize the detail algorithm of CGFR-L2 as follows. In this algorithm, $\mathbf{W}$ and $\mathbf{F}$ are alternately updated until convergence. Finally, the important features are selected according to the learned $\mathbf{W}$.

The algorithm of $CGFR-L2$ is as follows:

1. **Input** : Data matrix $X \in \mathcal{R}^{n\times d}$, labels $\mathbf{Y} \in \mathbb{R}^{n\times 1}$, number of projection dimension m, number of select features k, regularization parameter γ.

2. **Output** : k selected features.

3. Compute affiliation matrix $\mathbf{A}$ according to (6.32).

4. Initialize $\mathbf{W}$ such that $w_{ij} = \frac{1}{m}$, $\mathbf{S}$ such that $s_{ij} = (\frac{-W^T x_i - W^T x_{j_2}^2}{\gamma} + \eta)_+$ where η is the root of $\sum_{j=1}^{n} s_{ij} - 1 = 0$.

5. Repeat.

6. Update $\mathbf{F}$ by selecting the first c smallest values from the eigenvector of L_S.

7. For each i, update the i-th row of $\mathbf{S}$ by solving the problem in Equation (6.42).

8. Update $\mathbf{W}$ by selecting m eigenvectors of $S_t^{-1} X^T L_S X$ corresponding to its m smallest eigenvalues.

9. Repeat until convergence.

10. **Return** : Sort $\{w^1{}_2^2, ..., w^d{}_2^2\}$ in descending order, and select top k ranked features as ultimate result.

6.3.5 Optimization Algorithm for CGFR-L1

In this section, we propose an effective algorithm to solve problem (6.39). $\mathbf{W}$ and $\mathbf{F}$ can be solved with the same methods as those in CGFR-L2.

6.3.5.1 Updating $\mathbf{S}$ for fixed $\mathbf{W}$ and $\mathbf{F}$

Denote $\mathbf{g}_i \in \mathcal{R}^n$ where $g_{ij} = \mathbf{W}^T x_i - \mathbf{W}^T x_{j_2}^2$, and $\mathbf{v}_i \in \mathcal{R}^n$ where $\mathbf{v}_{ij} = f_i - f_{j_2}^2$. Since the above problem is independent between different i, we can solve it respectively for

each i

$$\min_{\mathbf{s}_i \geq 0, \mathbf{s}_i^T \mathbf{1}=1} \left(\mathbf{g}_i^T \mathbf{s}_i + \gamma \mathbf{s}_i - \mathbf{a}_{i1} + \lambda \mathbf{v}_i^T \mathbf{s}_i \right). \tag{6.49}$$

In this paper, we propose to use the iterative reweighted method to solve the above problem. The following theorem states that problem (6.49) is equivalent to another problem which is easier to be solved.

Problem (6.49) can be solved by iteratively minimizing the following problem

$$\min_{\mathbf{s}_i \geq 0, \mathbf{s}_i^T \mathbf{1}=1} \left(\mathbf{g}_i^T \mathbf{s}_i + \gamma Tr\left((\mathbf{s}_i - \mathbf{a}_i)^T U (\mathbf{s}_i - \mathbf{a}_i) \right) + \lambda \mathbf{v}_i^T \mathbf{s} \right), \tag{6.50}$$

where U is a diagonal matrix with $u_{jj} = \frac{1}{2|\widetilde{s_{ij}} - a_{ij}|}$, and $\widetilde{s_{ij}}$ is the current solution.

In the t-th iteration, let

$$\mathbf{s}_i^{t+1} = \arg_{\mathbf{s}_i} \min_{\mathbf{s}_i \geq 0, \mathbf{s}_i^T \mathbf{1}=1} \left[\mathbf{g}_i^T \mathbf{s}_i + \gamma Tr\left((\mathbf{s}_i - \mathbf{a}_i)^T U_t (\mathbf{s}_i - \mathbf{a}_i) \right) + \lambda \mathbf{v}_i^T \mathbf{s} \right], \tag{6.51}$$

which indicates that

$$\begin{aligned} &\mathbf{g}_i^T \mathbf{s}_i^{t+1} + \gamma Tr\left((\mathbf{s}_i^{t+1} - \mathbf{a}_i)^T U_t (\mathbf{s}_i^{t+1} - \mathbf{a}_i) \right) + \lambda \mathbf{v}_i^T \mathbf{s}^{t+1} \\ &\leq \mathbf{g}_i^T \mathbf{s}_i^t + \gamma Tr\left((\mathbf{s}_i^t - \mathbf{a}_i)^T U_t (\mathbf{s}_i^t - \mathbf{a}_i) \right) + \lambda \mathbf{v}_i^T \mathbf{s}^t. \end{aligned} \tag{6.52}$$

The following inequality holds for any positive vector $\mathbf{c} \in \mathcal{R}^{n\times 1}$ and $\mathbf{d} \in \mathcal{R}^{n\times 1}$:

$$\begin{aligned}\sum_{j=1}^{n} \frac{(\mathbf{c}_j - \mathbf{d}_j)^2}{2\mathbf{d}_j} \geq 0 \Longrightarrow \sum_{j=1}^{n} \frac{2\mathbf{c}_j\mathbf{d}_j - \mathbf{c}_j^2}{2\mathbf{d}_j} \leq \frac{\mathbf{d}_j^2}{2\mathbf{d}_j} \\ \Longrightarrow \sum_{j=1}^{n} \mathbf{c}_j - \sum_{j=1}^{n} \frac{\mathbf{c}_j^2}{2\mathbf{d}_j} \leq \sum_{j=1}^{n} \mathbf{d}_j - \sum_{j=1}^{n} \frac{\mathbf{d}_j^2}{2\mathbf{d}_j}.\end{aligned} \tag{6.53}$$

Substitute $\mathbf{c}$ and $\mathbf{d}$ in Equation (6.53) by $|\mathbf{s}_i^{t+1} - \mathbf{a}_i|$ and $|\mathbf{s}_i^t - \mathbf{a}_i|$ respectively, we have

$$\mathbf{s}_i^{t+1} - \mathbf{a}_{i_1} - \sum_{j=1}^{n} \frac{|s_{ij}^{t+1} - a_{ij}|}{2\mathbf{d}_j} \leq \mathbf{s}_i^t - \mathbf{a}_{i_1} - \sum_{j=1}^{n} \frac{|\mathbf{s}_{ij}^t - \mathbf{a}_{ij}|}{2|\mathbf{s}_{ij}^t - \mathbf{a}_{ij}|}. \tag{6.54}$$

Combining Equation (6.52) and Equation (6.54), we arrive at

$$\begin{aligned}&\mathbf{g}_i^T\mathbf{s}_i^{t+1} + \gamma\mathbf{s}_i^{t+1} - \mathbf{a}_{i_1} + \lambda\mathbf{v}_i^T\mathbf{s}^{t+1} \\ &\leq \mathbf{g}_i^T\mathbf{s}_i^t + \gamma\mathbf{s}_i^t - \mathbf{a}_{i_1} + \lambda\mathbf{v}_i^T\mathbf{s}^t.\end{aligned} \tag{6.55}$$

That is to say, minimizing problem in Equation (6.50) also decrease the objective of the problem in Equation (6.49) in each iteration t.

Problem in Equation (6.49) can be rewritten as

$$\min_{\mathbf{s}_i \geq 0, \mathbf{s}_i^T\mathbf{1}=1} \left[\frac{1}{2}\mathbf{s}_i^T\mathbf{U}\mathbf{s}_i - (\mathbf{U}\mathbf{a}_i - \frac{1}{2\gamma}\mathbf{g}_i^T - \frac{\lambda}{2\gamma}\mathbf{v}_i^T)\mathbf{s}_i\right]. \tag{6.56}$$

Let $\mathbf{q}_i = \mathbf{U}\mathbf{a}_i - \frac{1}{2\gamma}\mathbf{g}_i^T - \frac{\lambda}{2\gamma}\mathbf{v}_i^T$, so we need to solve the following problem for each i

$$\min_{\mathbf{s}_i \geq 0, \mathbf{s}_i^T\mathbf{1}=1} \left[\frac{1}{2}\mathbf{s}_i^T\mathbf{U}\mathbf{s}_i - \mathbf{q}_i^T\mathbf{s}_i\right]. \tag{6.57}$$

The Lagrangian function of Equation (6.57) is

$$\mathcal{L}(\mathbf{s}_i, \eta, \alpha_i) = \frac{1}{2}\mathbf{s}_i^T \mathbf{U}\mathbf{s}_i - \mathbf{q}_i^T \mathbf{s}_i - \eta(\mathbf{s}_i^T \mathbf{1} - 1) - \alpha_i^T \mathbf{s}_i, \tag{6.58}$$

where η and $\alpha_i \geq 0$ are the Lagrangian multipliers.

Taking the derivative of Equation (6.58) and setting to zero, we have

$$\mathbf{U}\mathbf{s}_i - \mathbf{q}_i - \eta\mathbf{1} - \alpha_i = 0. \tag{6.59}$$

Then for the j-th element of $\mathbf{s}_i$, we have

$$u_{ii}s_{ij} - q_{ij} - \eta - \alpha_{ij} = 0. \tag{6.60}$$

Note that $\alpha_{ij} = 0$ according to KKT condition [Boyd and Vandenberghe, 2004], then it can be verified that the optimal solution s_{ij} should be

$$s_{ij} = [\frac{1}{u_{ii}}(\eta + q_{ij})]_+. \tag{6.61}$$

We define the following function w.r.t. η

$$\mathbf{h}_i = \sum_{j=1}^{n} [\frac{1}{u_{ii}}(\eta + q_{ij})]_+ - 1. \tag{6.62}$$

According to the constraint $s_i^T \mathbf{1} = 1$, we have

$$\mathbf{h}_i = 0. \tag{6.63}$$

Therefore, the value of η is the root of function $\mathbf{h}_i$. Note that $\mathbf{h}_i$ is a piecewise linear and monotonically increasing function, thus the root can be easily obtained by Newtons method. After computing η, the optimal solution to the problem (6.50) can be obtained by Equation (6.61).

6.3.5.2 CSFG-L1 algorithm

We summarize the detail algorithm in CSFG-L1. In this algorithm, $\mathbf{W}$ and $\mathbf{F}$ are alternately updated until convergence. Finally, the important features are selected according to the learned $\mathbf{W}$.

The algorithm to solve problem (6.39) is as follows:

- Input: Data matrix $X \in \mathcal{R}^{n \times d}$, labels $\mathbf{Y} \in \mathbb{R}^{n \times 1}$, number of projection dimension m, number of select features k, regularization parameter γ.
- Output: k selected features.
- Compute affiliation matrix $\mathbf{A}$ according to (6.32).
- Initialize $\mathbf{W}$ such that $w_{ij} = \frac{1}{m}$, $\mathbf{S}$ such that $s_{ij} = (\frac{-W^T x_i - W^T x_{j_2}{}^2}{\gamma} + \eta)_+$ where η is the root of $\sum_{j=1}^{n} s_{ij} - 1 = 0$.
- Repeat.

- Update $\mathbf{F}$ by selecting the first c smallest values from the eigenvector of L_A.
- For each i, update the i-th row of $\mathbf{S}$ by solving the problem in Equation (6.57).
- Update $\mathbf{W}$ by selecting m eigenvectors of $S_t^{-1}X^T L_S X$ corresponding to its m smallest eigenvalues.
- Repeat until convergence.
- Return: Sort $\{w^1{}_2^2, ..., w^d{}_2^2\}$ in descending order, and select top k ranked features as ultimate result.

The further research will focus on the experiments on different data set, especially those with high-dimension data.

6.4 Remarks

In this chapter we use the *ODM* instead of adjacency matrix to develop some new algorithms in some other subjects in the area of graph theory. The Chapter is insisting of two parts: the balanced Min-cut based on *ODM* matrix; the supervised feature selection with constrained structured graph optimization. It has been illustrated that the *ODM* is more effective than the adjacency matrix when dealing with graph data. The further research will be developed on the comparison between *ODM* and adjacency matrix in different kinds of data in various of areas.

Chapter 7

Exchange of majority in opinion evolution

7.1 Introduction

7.1.1 The phenomena of majority exchanges of the opinions

In this chapter, we will develop an application of the opinion dynamics in the area of voting prediction. In nature and society, not only organisations, but also some social events can be represented as graphs or networks, such as the votings and elections. During the long period of a voting from its launch to the result that is revealed, every participant can be described as a node in a network. Once two nodes talk to each other, a connection exists between them. They will influence each other through the connections and fix their own opinions eventually.

Recently, attention has been given to this kind of opinion consistency problems in research of social networks. This is associated with whether and how long it takes for

an individual's opinion to reach a consistent status [Sznajd-Weron and Sznajd, 2000, Castellano et al., 2009, Lambiotte et al., 2009]. Knowledge of opinion dynamics is relevant to the prediction of collective behaviors such as voting and election [Stauffer, 2001, Biswas and Sen, 2009, Hegselmann and Krause, 2000]. Interestingly, the original purpose of studying opinion dynamics was to predict the final voting result in real social networks [Coughlin, 1992, Yildiz et al., 2011, Bernardes et al., 2002, Parhami, 1994, Porter et al., 2005, Halu et al., 2013, Ding and Liu, 2010]. An important feature of voting is that voters can't always reach a consensus before the end of the process. That is why most voting processes present a non-neutral result, for instance, in the political election or the talent show, we do not expect all the participants vote for one candidate, instead, the one gaining the majority of the voters wins.

In the previous studies of the opinion dynamics, we have observed that the majority of opinions change until the consensus is achieved. This is similar to what happens in the real voting. In this Chapter, we will investigate the reasons and features of this phenomena. We will discuss how the network topology impacts it.

7.1.2 Research structure

In Section 7.1, we give a background introduction of the study. In Section 7.2, we simulate the opinion process on five typical networks and observe the exchanges of majority. In Section 7.3, we observe that the phenomena happen in a real-life 'face-to-face' network. In Section 7.4, we discuss the reasons and features of the exchanges. We investigate how the result is impacted by the different topology of the networks. In

Section 7.5, we conclude the study and look forward to the further study in this area.

7.2 Simulations of exchanges of the majority

The opinion networked dynamical models studied here are based on the opinion model of Curtis and Smith [Curtis and Smith, 2008]. The five networks are were defined in Chapter 2. In this study, all the networks discussed are assumed to be connected networks. There is always a path between any two nodes i and j. The networks are all unweighted-undirected networks. Evolution and competition in the 5 networks are shown in the Fig 7.2-4. We use $ODE45$ in Matlab to solve the Equation 7.1.

$$\dot{x_i} = \sum_{j=1}^{N} b_j A_{ij}(x_j - x_i), \quad i = 1,2\ldots N, \quad t \geq 0 \tag{7.1}$$

In Figure 7.1-7.3, t denotes in the $ODE45$. If the time is long enough, all the opinions will converge at a value close to 0 [Pecora and Carroll, 1990, Grabow et al., 2012, Chavez et al., 2005]. Since the consensus can't be exact 0, we restrict the opinion values in 8 digits after decimal points, the opinions can converge at 0. In a real voting, we divide those who hold the positive opinions and the negative ones into two opposing camps. The opinions with positive values and negative values are considered as two opposing parties in a voting. Given enough time, the opinions will all become zero as illustrated in the next section. We mark them as "consensus" in the graph since they have achieved the consensus and will not change any more. Meanwhile, the numbers of people holding positive and negative opinions at time t are recorded in the simulation.

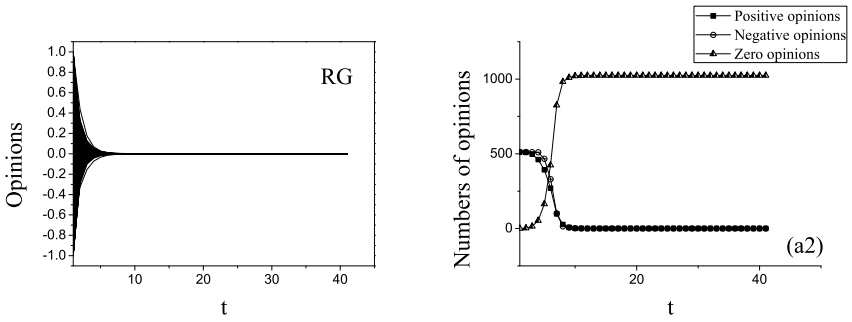


Figure 7.1: The opinion evolution process and the exchanges in ER .

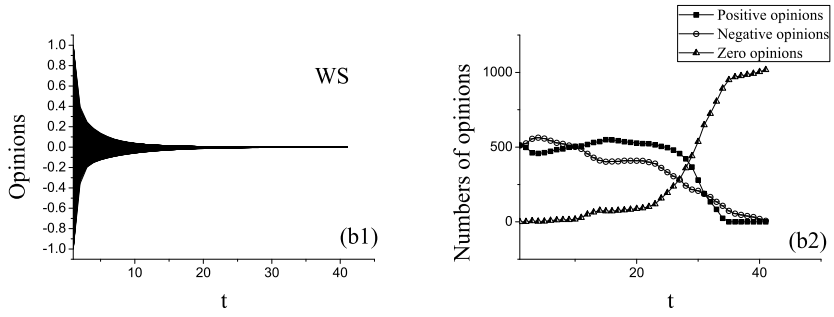


Figure 7.2: Figures (b1)and (b2) show the convergence and the competition of the opinions in WS. The convergence on WS takes the longest time among all, without the most frequent exchanges.

The party with more people is the leading party while the other is the opposing party. The *RG* supports the fastest opinion convergence among all the networks, which leaves little time for the exchange to happen. In Figure 7.2(a2) no obvious exchange can be observed.

As can be observed in Figures 7.3(a1),(b1),(c1),(d1) and (e1), with the same initial opinions $X^0 = x_1^0, x_2^0, \cdots, x_N^0$, different topologies lead to the same consensus $x_s = x_1^t = x_2^t = \cdots = x_N^t$ at time t, which is the average of the initial opinions. We have proved it in Chapter 2.

The identity consensus on all the networks may mislead the prediction of the result.

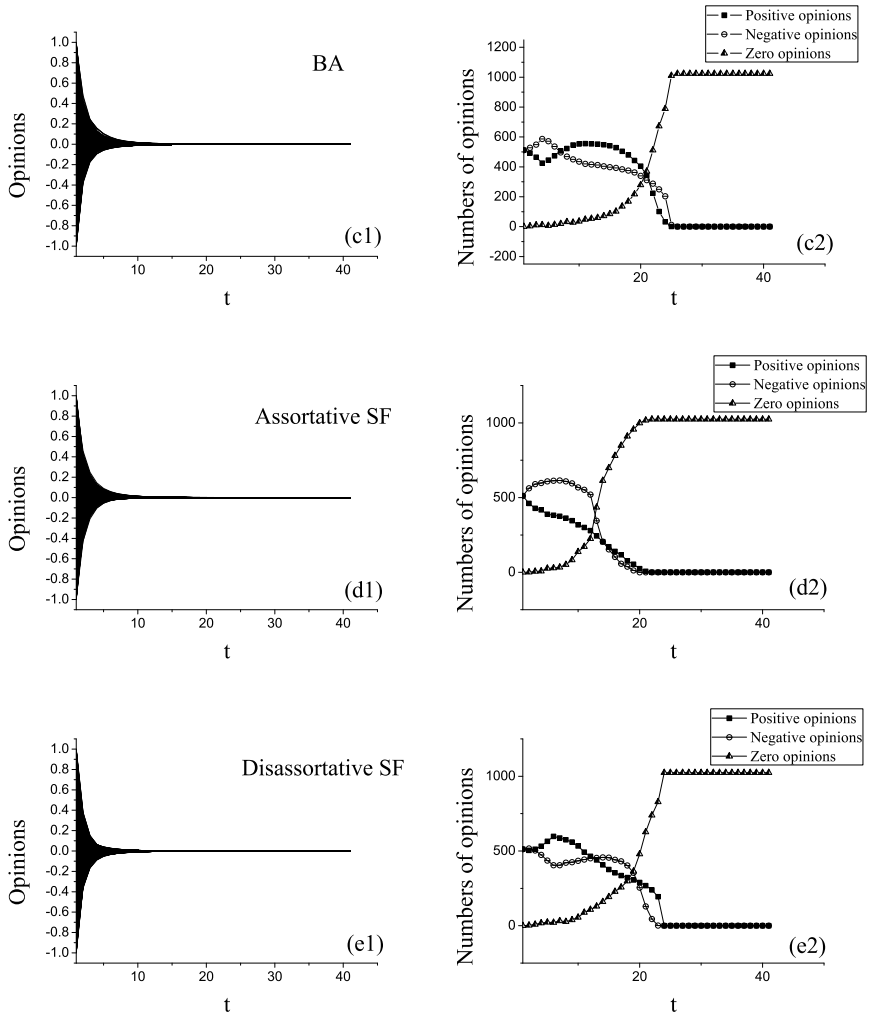


Figure 7.3: In Figures(c1),(d1) and (e1) are the opinion evolution in *BA*, *AssortativeSF* and *DisassortativeSF*. Figure(c2),(d2) and (e2) show the competition of opinions. The difference between the convergence speed is too small to observe. However, as recorded, it's the fastest in *ASSF* among the three and the slowest in *DSSF*.

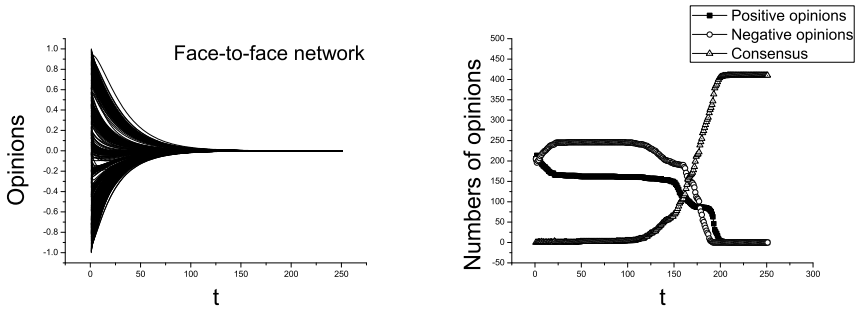


Figure 7.4: The majority change on the face-to-face network.

In fact, most social collative behaviors end before the consensus comes. Suppose the opinions of voters are affected by their friends in an election between two parties (negative and positive). The results will be totally different if the election ends at different time in the five networks.

7.3 Application of the exchange in majority

In [Isella et al., 2011], a behavioral network of face-to-face contacts in a long running museum exhibition was tracked. The network consists of 251 nodes and 5530 links. We give each node a random initial opinion between $[-1, 1]$ and simulate the process of their talks. The degree correlation of the network is 0.755, which means it has strong assortativity. Some obvious exchanges of majority have been observed in this opinion process. See Figure.7.4. If a decision or a voting is going to be made based on the face-to-face communications, the time to collect the opinions and put an end to the event will significantly impact the final result.

7.4 Discussion

The opinion evolutions approach the same value of consensus with a varying speed on the five networks. During the process, a significant phenomenon is the exchange of the majority between positive and negative opinions, which is caused by the local consensus before the global one [Arruda et al., 2013, Arenas et al., 2006, Torok et al., 2013]. From Equation 7.1 we can get

$$x_i^t = \sum_{j=1}^{N} P_{ij} e^{\Lambda_j t} x_j^0 \tag{7.2}$$

So the difference of opinions between any two nodes i and m is

$$|x_i^t - x_m^t| \leq \sum_{j=1,k=1}^{N} |P_{ij} - P_{mj}| e^{\Lambda_j t} P_{jk}^{\dagger} x_k^0 \tag{7.3}$$

$$\leq \sum_{j=1}^{N} |P_{ij} - P_{mj}| e^{\Lambda_j t} \tag{7.4}$$

The Equations 7.2 and 7.3 illustrate how the network topology impacts the process of opinion convergence. When time is long enough, all exponentials are zero and all the opinions go to identity. During the process, the small eigenvalues ensure those nodes with similar projections on the eigenvectors to get synchronized eventually. In other words, the small communities in the network will achieve a local consensus before they arrive the global consensus together. When the opinions of a community move together from positive to negative, or in contrary, the exchange of majority may happen.

In this study, we test how the network topology impacts the exchange frequency F

of the leading opinions and the ratio of the longest leading time R in the synchronization process. Two statistical parameters are chosen to represent the network topology: the clustering coefficient C and the average path length L. Both of them will be adjusted by changing the average degree of the nodes K. In Figure 7.5, we illustrate how F and R change by increase K, C and L. When K goes from 3 to 5, the exchange frequency F are all dropping in the five networks, while the longest leading time is increasing. The clustering coefficient C which has a positive correlation with K affects the F and R in the same way while the average path length L in the contrary way.

Since we don't observe clear exchanges in ER, we will focus on the other four networks. Because of the small world feature, the community structure is unclear in WS. There are not too many chances for the small groups of opinions to cross the zero together. For the same reason, it's harder to replace the majority in WS than in any other networks. So the WS holds relatively lower exchange frequency F and longer leading time R.

For the classes of SF networks, the community structures are clearer. Although $ASSF$ and $DSSF$ are generated from BA, the varying of assortativity causes different average path lengths L, clustering coefficient C in the three networks. The $DSSF$, with longer L and higher C, provides the most frequent exchanges with very short leading times. There are many small communities with similar sizes in the $DSSF$. Once an exchange happens, it's easy for the next one to replace it. The $ASSF$ with small numbers of huge communities is in contrary.

In real-life voting, the leading party may want to maintain the superiority while

the opposition party may hope the next exchange to come soon. As can be observed in Figure.7.5, for the leading party, any behavior to prevent C from dropping or L from increasing may help, for instance, the establishment of small clusterings, the communications between large degree people and the isolated ones.

7.5 Remarks

In this study, we have investigated the exchanges of the advantage between two parties in a voting. We consider all the people participating in as a network. Five typical networks are selected to describe the most possible structures of real-life networks. The opinion evolution during a voting is simulated on the five networks. We have investigated the reasons and the features of the exchanges. We have found that the structure of the networks will significantly impact the frequency of the exchanges and the time length between every two exchanges. A new method to predict and manipulate a voting is suggested. It is difficult to build linear relations between the exchanges and the structural characteristics of the network. When adding or deleting any of the connections, the average path length(L), the clustering coefficient(C) and other characteristics will all change, which causes unpredicted impact to the exchanges. In the future, we intend to build clearer relations between exchanges and topology. Then it is possible to develop a method to control the result of an voting by adjusting the connections.

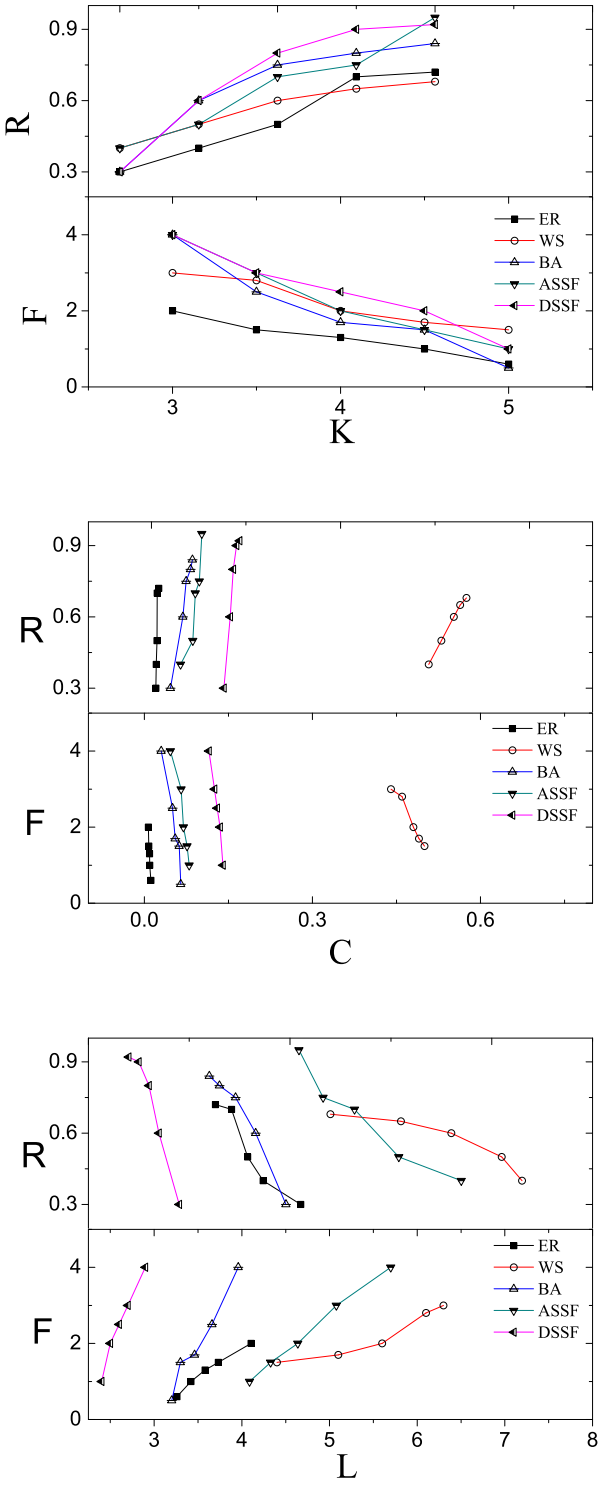


Figure 7.5: When increasing the clustering coefficient, the exchange frequency drops and the longest majority grows. The increasing of the average path length decreases the exchange frequency while increase the longest majority.

Chapter 8

The Suppression Effect on complex networks built by Achlioptas Process

8.1 Introduction

8.1.1 Opinion evolution on growth networks

In the previous chapters we studied continuous opinion models in complex heterogeneous networks to investigate how nodes holding different opinions come up with a consensus. We focused on the situation with a social outcast in the network, a node who has strong and negative influence to others connected to it. The structure and evolving mechanism are crucial factors to determine whether there will be a stable consensus after the network stops evolving and the network efficiency to achieve a stable status if there might be one.

In this chapter, we will discuss a group of growth network models, where network grows and opinions evolve at the same time. Some special formation of network evolu-

tion may undergo an explosive phase transition, which means, a small number of new connections adding in network causes immense change in network dynamic. Consensus may appear after the transition point [Shao et al., 2009]. We discuss the finite-size scaling in the model. Then simulations are taken on Scale-Free network and random network to compare the efficiency of network dynamic in producing consensus with the same number of connections. The surprising results come up that the network with explosive transition does not show apparent advantage when there is no outcast in the network. However, it strongly suppresses the influence of the outcast and promotes the probability of a stable consensus when an outcast exists. This mechanism for network growing may solve a typical problem of complex network, that the scale-free network is robust under random attack but fragile under aimed attack.

8.1.2 Research structure

This chapter is structured as follows: In Section 8.1, we give a brief introduction of the research. In Section 8.2, we use the Achlioptas Process to build growth ER random graph and SF network. The finite-size scaling(FSS) is discussed. In Section 8.3, we simulate the opinion evolution on ER and SF built by Achlioptas Process. The emergence of opinion convergence has been observed in both cases. In Section 8.4, we present the suppression effect on these networks. In Section 8.5, we give the conclusions.

8.2 The growth networks by Achlioptas Process

Achlioptas Process is the network evolving process from Dimitris Achlioptas

For SF:

1. Since the SF network obeys a power-law $P(k) = k^{-\lambda}$ with $2 \leq \lambda \leq 4$ approximately, a degree sequence $\{k_1, k_2, \cdots, k_N\}$ by power law with $\lambda = 2.5$ is set. Give the node i of the N nodes k_i stubs(half link).

2. Choose four stubs randomly by the probability $p_i = k_i / \sum^{N_s k_s}$, and connect them with two links randomly.

3. Choose the link merging a smaller cluster. There might be three possible conditions as follows (Figure 8.1 is from reference

   For ER, the procedure is not so complicated. We only need to add two potential links each time and pick one of them using the Steps 3 and 4 in the SF procedure and give up the other link. The degree distribution will merge obeying poisson distribution as any normal ER.

   For both ER and SF models, the growth proceeds until one reaches the desired density of links d. We define d as the number of links of the graph divided by the total number of links present in the graph when it has been completed. The total number of connections $d = p_n N$ are added in the system with a threshold p_c where a single giant component emerges in the network. If we set the size of the largest cluster in the system as G_s, a sudden change can be observed and many other unexpected behaviors emerge as well. Here we need to discuss the

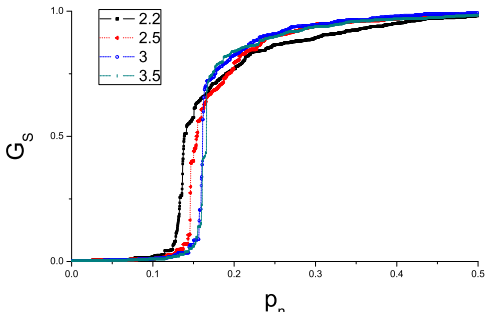


Figure 8.1: The merge of giant component on scale-free network with $\lambda = 2.5, 3, 3.5, 4$

finite-size scaling [Fortunato and Radicchi, 2011, Cho et al., 2010, 2009] of the networks numerically to observe if the exponent λ will impact the emergence of dynamical behaviors(see Figure 8.2). We have discovered that there exists a critical point λ_c. When $2 > \lambda_c > \lambda$, the transition is in the second-order as in conventional *SF* networks. When $\lambda > \lambda_c$, the p_c is finite and the transition is first-order, which means there is a jump in the size of the giant component as shown in Figure 8.2. We have discovered that $2 > \lambda_c > 2.4$. We define the discontinuity of G_s as δG, which is the distance between two tangent lines, one from the rapidly increasing transition region and the other from the smoothly increasing curve after the jump. In the finite-size network when $\lambda > 2.4$, the G_s shows the first-order transition. However, when $\lambda \to 2$, the transition point p_c and the δG decrease. This can be observed in networks of different sizes.

The mechanism by which a giant component forms in conventional *SF* networks is different from in *ER*. In *ER*, due to the lack of hubs, the multiple isolated small components are created and merged together. The threshold p_c occurs when a

sudden connection of those small components, which is around 0.5 as observed numerically. In *SF* the the giant component grows from the hubs with high degree and aggregates small-size components eventually. During the process, if two nodes get selected from the same component, the component size will not be changed by adding a link between them. Thus, the existence of a giant component implies that even under *AP*, the probability of growing the giant component is very high.

The size of G_s can also be written as

$$G_s = 1 - \sum_s s n_s(t), \tag{8.1}$$

where n_s is the number of inner links of the s-size small cluster at time t. All clusters in the network will be calculated except for the largest one. From Figure 8.2 we can see the transitions of G_s for different *SF*s intersect at approximately one point. We consider the t-intercept of the tangent of G_s at t_x, denoted as $t_d(N)$. Then the time is

$$t_d(N) = t_x - G_s(t_x)(\frac{dG_s(t)}{t})|^{-1}_{t=t_x}. \tag{8.2}$$

When N is large enough, the derivative of G_s diverges as

$$\frac{dG_s(t)}{t})|_{t=t_x} \sim N^{\theta}. \tag{8.3}$$

As simulated in previous studies, $\theta \approx 0.5$. The transition is discontinuous when $N \to \infty$.

8.3 The emergence of opinion convergence by Achlioptas Process

The sudden change of the largest cluster size is only one of the phenomena coming after the explosive transition by *AP*. Phase transition will merge in relevant statistical physical features such as the shortest path length and the coupled capacity. The synchronization ability and robustness will be significantly enhanced. However, whether these are phase transitions remains unproved. The opinion model from Chapter 2 is a typical application of the network synchronization.

$$\frac{\partial x_i}{\partial t} = \sum_{j=1}^{N} b_j a_{ij}(x_j - x_i), i = 1, 2 \ldots N. \tag{8.4}$$

Give opinions to N isolated nodes, $\{x_1, x_2, \cdots, x_N\}, x_i \in [0,1]$. Adding a link into the network and nodes on each edge can start talking to each other continuously ever since. Compare the standard deviation of opinions in *ER* by *AP* and the normal *ER*, we may know the efficiency of the two networks to achieve a consensus(See Figure 8.3). The *ER* and *SF* are both made of 128 nodes and 1000 links.

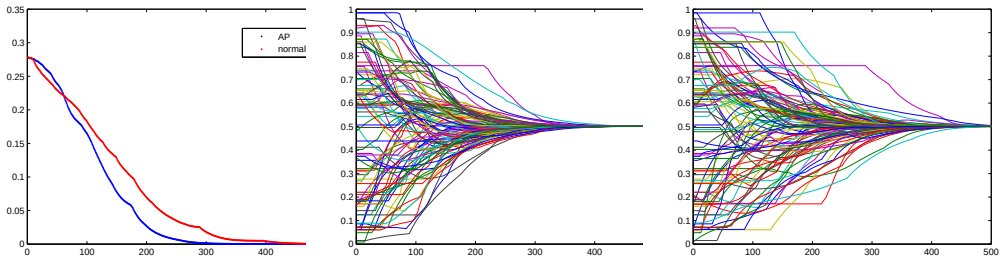


Figure 8.2: The comparison between opinion process on ER by Achlioptas Process and the normal ER.

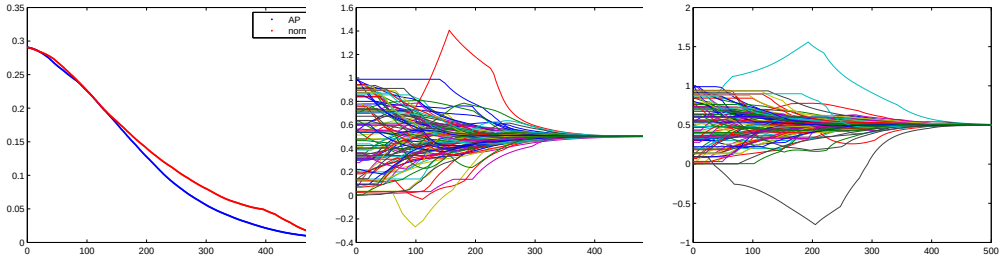


Figure 8.3: The comparison between opinion process on SF by Achlioptas Process and the normal SF.

The same experiment has been taken on the SF group and the results are shown in Figure 8.4.

In both groups, we observe that the AP rule accelerates the opinion convergence.

8.4 Suppression Effect by Achlioptas Process

The simplest linear opinion model is a communication between two people x_1^0 and x_2^0 with their initial opinions a and b respectively. After talking, both opinions become $x_1^1 = x_1^2 = (a+b)/2$. The network structure and nonlinear dynamic give more possibility of results from the same initial conditions. Network evo-

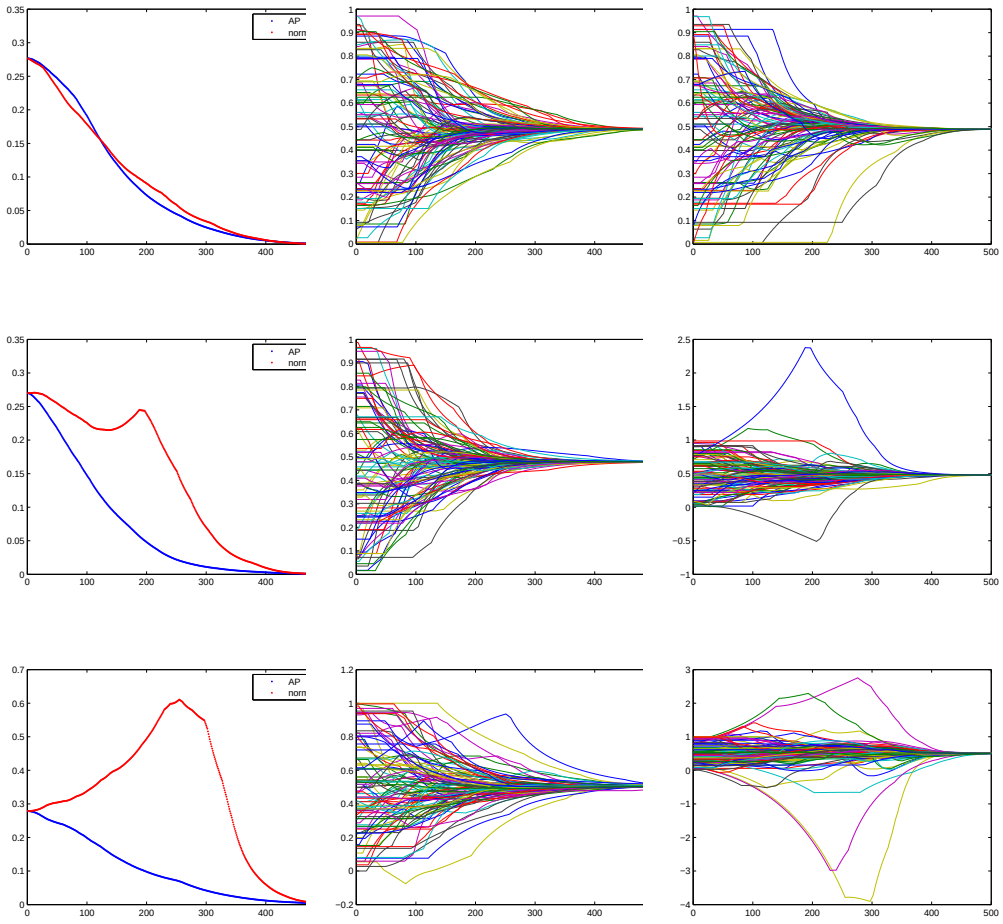


Figure 8.4: Small outcast with lowest, medium and highest degree on *ER*

lution like *AP* process can create emergence of the ability of a single node or cluster, as well as restrict it. We put an outcast into the system, who has the negative influence ability to others. It has been demonstrated that SF is robust to random attack while fragile to intentional attack. So we take the experiments respectively setting the outcast with degree from low to high.

The broad distributions of *SF* indicate that there is a whole hierarchy of node roles based on their degrees, going from a large majority of nodes with low de-

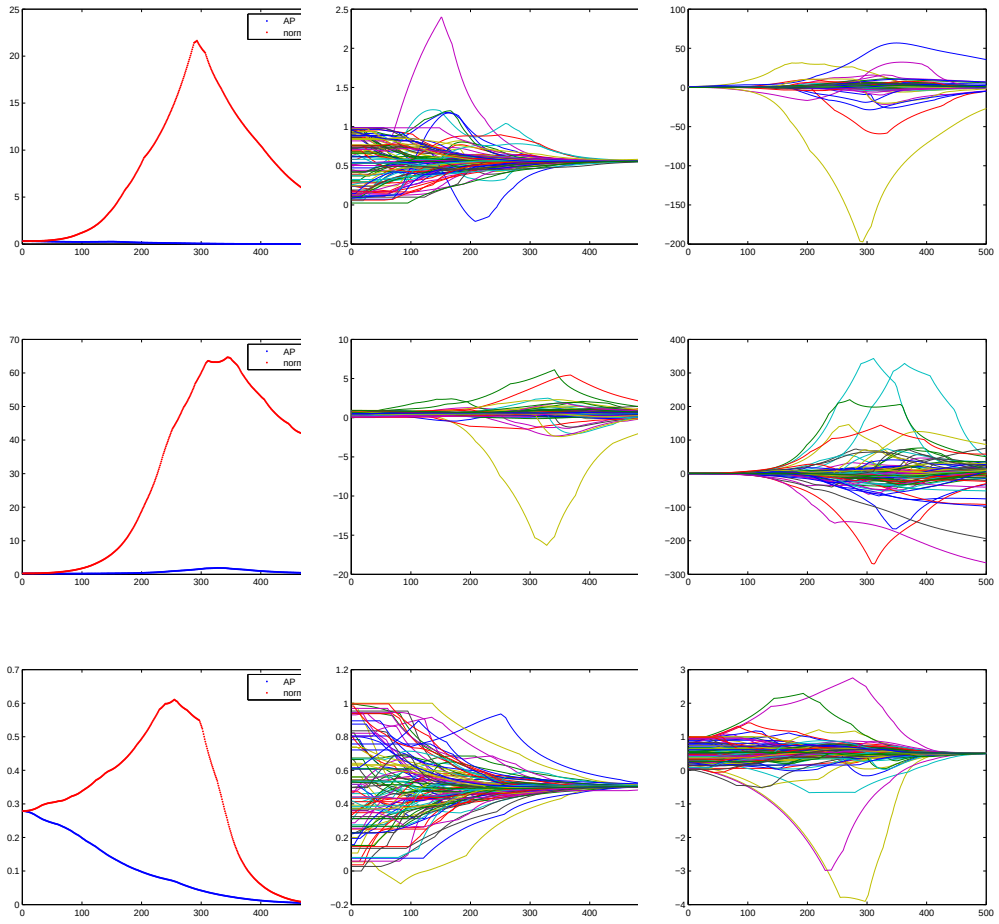


Figure 8.5: Small outcast with lowest, medium and highest degree on *SF*

gree to a small subset of nodes with high degree, or hubs. The hubs have a fundamental role for the structure and dynamics of networks. Normal *SF* networks have so many hubs that a very small fraction of links is enough to keep a macroscopic fraction of nodes of the graph in the same connected component, which can be equivalently stated by saying that the percolation threshold is zero.

8.5 Remarks

In this chapter, we have the conclusions as follows:

- *AP* rule causes percolation transition of the size of the largest cluster in the growing network. The first-order transition occurs on *SF* with the power-law exponent $\lambda > \lambda_c$. The second-order transition is restrained in the region $\lambda_c > \lambda > 2$. The critical point dividing these two regions is $2.4 > \lambda_c > 2$.
- When there is no social outcast on the networks, *AP* rule accelerates the opinion convergence, which is more obvious in *ER* than in *SF*.
- When there is a social outcast, *AP* rule emerges with a stronger suppression ability against the outcast in *SF* than in *ER*. The ability differences in two networks are widen while the outcast gets stronger.

Chapter 9

Conclusion

In this thesis, we have investigated the opinion dynamics on complex networks and applications in social networks. The opinion dynamics are considered as consensus problem in the linear system and the synchronization in the nonlinear system. We have discussed several substrates with the methods of graph theory, asymptotic methods, master stability function, complex network tools and some other tools from the mathematics and physics.

In Chapter 2, we have put five kinds of networks into consideration. The opinion dynamics are simulated on five networks. We have discussed some disagreements from the previous studies and investigated the relations between network topology and opinion convergence time. The results are as follows:

(a) For most complex networks, a shorter average path length L indicates a faster opinion convergence. However, for the group of regular network, WS and RG, when the L drops from $p = 0$ to $p = 1$, the convergence time

t decreases rapidly in the beginning then becomes stable far before $p = 1$.

(b) The clustering coefficient C has no monotonic relation with t nor with L.

(c) The Pearson coefficient Pr as the measure of degree correlation describes how the communities in the network impact the convergence.

The analysis from the graph theory is provided to solve the raw problem of opinion convergence time. However, it is still unsolved how to control the network behaviors by adding or deleting the nodes and connections. The further study on this topic will focus on the network control and the applications to real life networks.

In Chapter 3, we have discussed weighted-directed complex networks and investigated the algebraic connectivity λ_{ac} of these networks. We simulate the opinion dynamics on networks with an outcast. Then we analyze how the opinion system evolves with an outcast which is powerful enough. We divide all the nodes into three types and conclude that:

(a) The outcast can be impacted by more than one neighbour with initial opinions larger or smaller than its own opinion. The outcast's opinion will increase or decrease constantly.

(b) The neighbours of the outcast, like the node 2 in Figure.3.7, are the key nodes to decide whether the system will converge or not. If the second item of Equation (3.21) contributes more to the ith opinion, the outcast will be attracted to the main group. Otherwise, the neighbour's opinion will

move against the outcast's opinion. The difference between the opinions will become larger.

(c) Those who don't connect to the outcast directly, like the nodes $3-5$ in Figure 3.7, are only influenced by node 2. If node 2 tends to converge with the outcast, all of the nodes will converge. Otherwise, nodes like nodes $3-5$ will tend to approach node 2. It will look like the outcast pushes all the other nodes away. However, it's not guaranteed that node 2 and nodes $3-5$ will have a local consensus.

In Chapter 4, we have observed the nonlinear opinion dynamics on the weighted-directed complex networks. We have reviewed the history of synchronization in dynamical systems. We have discussed the Lyapunov exponent and the master stability function. Several algorithms are presented to calculate the Lyapunov exponent. We test the finite-size scaling of the WS network and BA model to determine the network parameters. We simulate the system evolution on the RG, WS and BA under perturbation. The stability of synchronization in these networks is discussed.

In Chapter 5, we have used the linear opinion process to develop an algorithm of network partition. During the opinion evolution towards consensus, some nodes achieve local consensus much earlier than the global consensus. We conclude that these kinds of nodes have closer relations even there is no direct link between them. In this way, we partition the network using the opinion dynamical

matrix(ODM) instead of adjacency matrix. The algorithm is tested in benchmark networks and a real network *RALIC* from UCL. The partition results are better than what we have obtained from the previous methods based on adjacency matrix.

In Chapter 6, we have used the ODM instead of adjacency matrix to develop some new algorithms in some other subjects in the area of graph theory. The chapter contains two parts: the balanced Min-cut based on ODM matrix; the supervised feature selection with constrained structured graph optimization. It has been illustrated that the ODM is more effective than the adjacency matrix when dealing with graph data. The further research will be developed on the comparison between ODM and adjacency matrix in different kinds of data in different areas.

In Chapter 7, we have investigated the exchanges of the advantage between two parties in a voting or an election. We consider all the people participating in a voting as a social network. Five typical networks are selected to describe the most possible structures of real-life networks. The opinion evolution during a voting is simulated on the five networks. We have found that the structure of the networks significantly impact the frequency of the exchanges and the time length between every two exchanges. A new method to predict and manipulate a voting is suggested.

In Chapter 8, we have developed growth networks with opinions evolving at

the same time. The Dimitris Achlioptas process (AP) is used to generate the networks. In this study, we have the conclusions as follows:

- *AP* rule causes the percolation transition on the size of the largest cluster in the growing network. The first-order transition occurs on *SF* with the power-law exponent $\lambda > \lambda_c$. The second-order transition is restrained to the region $\lambda_c > \lambda > 2$. The critical points dividing these two regions is $2.4 > \lambda_c > 2$.
- When there is no social outcast on the network, *AP* rule accelerates the opinion convergence, which is more obvious on *RG* than on *SF*.
- When there is a social outcast, *AP* rule emerges with a stronger suppression ability against the outcast on *SF* than on *RG*. The ability differences in two networks are widen while the outcast gets stronger.

In the future work, we will focus on investigating any possible statistical physical statistics of network topology. W e can build relations between these characteristics and any dynamical behaviors on the networks. The methodology of network control in both network models and real-life networks will be developed by adjusting these characteristics.

Bibliography

R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47, 2001. 14, 41, 49, 62, 124

W.N. Anderson and T.D. Morley. Eigenvalues of the laplacian of a graph. *Linear Multilinear Algebra.*, 18:141–145, 1985. 71

A. Arenas, A. Diaz-Guilerab, and C.J. Prez-Vicente. Synchronization processes in complex networks. *Phys.D.*, 224(1):27–34, 2006. 16, 130, 131, 136, 202

A. Arenas, A. Diaz-Guilera, Moreno Y. Kurths, J., and C. Zhou. Synchronization in complex networks. *Phys. Rep.*, 469(3):93–153, 2008. 34, 70

G.F.de. Arruda, Peron, T.K. Dalmaso, M.G. and. Achcar J.A. de Andrade, and F.A. Rodrigues. The influence of network properties on the synchronization of kuramoto oscillators quantified by a bayesian regression analysis. *J.Stat.Phys.*, 152:519–533, 2013. 63, 202

A.L. Barabasi, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Phy.A.*, 272(1):173–187, 1999. 130

M. Barahona and L. M Pecora. Synchronization in small-world systems. *Physical review letters*, 89(5):054101, 2002. 63

E. R. Barnes. An algorithm for partitioning the nodes of a graph. In *20th IEEE Conf.*, 1981. 118, 123

A.T. Bernardes, D. Stauffer, and J. Kertesz. Election results and the sznajd model on barabasi network. *Eur. Phys. J. B.*, 25:123–127, 2002. 197

S. Biswas and P. Sen. Model of binary opinion dynamics: Coarsening and effect of disorder. *Phys. Rev. E*, 80(2):027101, 2009. 197

B. Blasius, A. Huppert, and L. Stone. Complex dynamics and phase synchronization in spatially extended ecological systems. *Nature*, 399(6734):354–359, 1999. 130

S. Boccaletti. *The synchronized dynamics of complex systems.* Amsterdam: Elsevier., 2008. 103

S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D-U. Hwang. Complex networks: Structure and dynamics. *Phys.Rep.*, 424(4):175–308, 2006. 103

W. E. Boyce. Probabilistic methods in applied mathematics. *Academic Press.*, page 0173, 1968. 66

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 187, 193

T. L. Carroll and L. M. Pecora. Synchronizing chaotic circuits. *Circuits and Systems, IEEE Transactions on*, 38(4):453–456, 1991. 103

C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81(2):591646, 2009. 30, 197

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. 177

Yin-Wen Chang, NTU EDU, and Chih-Jen Lin. Feature ranking using linear svm. *Causation and Prediction Challenge Challenges in Machine Learning, Volume 2*, page 47, 2008. 178

M. Chavez, D-U. Hwang, A. Amann, H. G. E. Hentschel, and S. Boccaletti. Synchronization is enhanced in weighted complex networks. *Phys. Rev. Lett.*, 94(21):218701, 2005. 40, 53, 198

Y. Chen, G. Rangarajan, and M. Ding. Stability of synchronized dynamics and pattern formation in coupled systems: Review of some recent results. *Communications in Nonlinear Science and Numerical Simulation*, 11(8):934–960, 2006. 103

J. Cho, Y.and Kim, J. Park, B. Kahng, and D. Kim. Percolation transitions in scale-free networks under the achlioptas process. *Physical review letters*, 103 (13):135702, 2009. 209

Y. Cho, S. Kim, J. Noh, B. Kahng, and D. Kim. Finite-size scaling theory for explosive percolation transitions. *Phys.Rev.E.*, 82(4):042102, 2010. 209

P. Clifford and A. Sudbury. A sample path proof of the duality for stochastically monotone markov processes. *Ann. Probab.*, pages 558–565, 1985. 31

R. Cohen and S. Havlin. Scale-free networks are ultrasmall. *Phys. Rev. Lett.*, 90(5):058701, 2003. 63

P.J. Coughlin. *Probabilistic Voting Theory.* Cambridge University Press., 1992. 197

D. Cumin and C.P. Unsworth. Generalising the kuramoto model for the study of neuronal synchronisation in the brain. *Phy.D.*, 226(2):181–196, 2007. 131

J. P. Curtis and F. T. Smith. The dynamics of persuasion. *Math. Models. Methods. Appl. Sci.*, 2:115–122, 2008. 20, 22, 35, 52, 53, 74, 130, 198

G. Deffuant, D. Neau, F. Amblard, and G Weisbuch. Mixing beliefs among interacting agents. *Adv. Compl. Sys.*, 2000. 14, 32, 33

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik.*, 1:269–271, 1959. 62

F. Ding and Y. Liu. Modeling opinion interactions in a bbs community. *Eur.Phys.J.B.*, 78:245–252, 2010. 197

P Erdos and A Ranyi. On random graphs. *Publ. Math. Debrecen*, 1959. 42

L. Euler. The solution of a problem relating to the geometry of position. *Commentarii Academiae Scientiarum Imperialis Petropolitanae.*, 8:128–140, 1736. 123

I. J. Farkas, Imre. Derenyi, A. Barabasi, and Tamas. Vicsek. Spectra of "real-world" graphs: Beyond the semicircle law. *Phys. Rev. E.*, 64:026704, 2001. 38

M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23:298–305, 1973. 71, 78

G.W. Flake, S. Lawrence, C. L. Giles, and F.M. Coetzee. Self-organization and identification of web communities. *IEEE Computer.*, 35:66–71, 2002. 127

S. Fortunato. Community detection in graphs. *Phys. Reps.*, 486(3):75174, 2010. 123, 126, 137, 139

S. Fortunato and F. Radicchi. Explosive percolation in graphs. In *Journal of Physics: Conference Series*, volume 297, page 012009. IOP Publishing, 2011. 209

M. Girvan and M.E. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826, 2002. 124, 127, 142

J. Gomez-Gardenes and Y. Moreno. Synchronization in networks with variable local properties. *Internat. J. Bifurc. Chaos*, 17:2501–2507, 2007. 69

C. Grabow, S. Grosskinsky, and M. Timme. Do small worlds synchronize fastest? *Eur.Phys.J.B.*, 90:48002, 2012. 198

R. Guimera and L.A.N. Amaral. Functional cartography of complex metabolic networks. *Nature.*, 433(7028):895–900, 2005. 126

L. Gulyas, G. Horvath, T. Cseri, and G. Kampis. An estimation of the shortest and alrgest average path lenghth in graphs of given density. *ECCS11.*, pages 37–43, 2011. 63

A. Halu, K. Zhao, A. Baronchelli, and G. Bianconi. Connect and win: The role of social networks in political elections. *Eur.Phys.Lett.*, 102:16002, 2013. 197

F. Harary. *Graph Theory.* Addison-Wesley., 1969. 39, 62

C. Hayashi. *Nonlinear oscillations in physical systems.* New York: McGraw-Hill., 1964. 103

R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence: models, analysis and simulation. *JASSS*, 5, 2000. 32, 197

R Hegselmann and U Krause. Opinion dynamics and bounded confidence: models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 2002. 32

R. Hegselmann and U. Krause. Opinion dynamics driven by various ways of averaging. *Computational Economics.*, 2004. 33

P. W. Holland and S. Leinhardt. *Transitivity in structural models of small groups.* Comparative Group Studies. 68

H. Hong, B. Kim, M. Choi, and H. Park. Factors that predict better synchronizability on complex networks. *Phys. Rev. E.*, 69(6):067105, 2004. 34, 38

Samuel H Huang. Supervised feature selection: A tutorial. *Artificial Intelligence Research*, 4(2):p22, 2015. 177

C. Huygens. *Horologium Oscillatorium.* Apud F. Muguet, Parisiis, France., 1673. 102

Iñaki Inza, Pedro Larrañaga, Ramón Etxeberria, and Basilio Sierra. Feature subset selection by bayesian network-based optimization. *Artificial intelligence*, 123(1):157–184, 2000. 178

L. Isella, J. Stehle, A. Barrat, C. Cattuto, J. Pinton, and W.V.den. Broeck. What is in a crowd? analysis of face-to-face behavioral networks. *J. Theo. Bio.*, 271:266–280, 2011. 201

E A. Jackson. Controls of dynamic flows with attractors. *Physical Review A*, 44 (8):4839, 1991. 103

J. Jost and M. P. Joy. Spectral properties and synchronization in coupled map lattices. *Phys. Rev. E.*, 65(1):016201, 2001. 38

B. H. Junker and Falk. Schreiber. *Analysis of Biological Networks.* Wiley-Interscience, New York, USA., 2008. 118

T. Kapitaniak, L. O. Chua, and G. Zhong. Experimental synchronization of chaos using continuous control. *International Journal of Bifurcation and Chaos*, 4(02):483–488, 1994. 103

J. Kelner. Lecture notes of linear al. *MIT*, 2009. 39, 53, 54, 55, 79

R. Lambiotte, J. Saramäki, and V. D. Blondel. Dynamics of latent voters. *Phys. Rev. E*, 79:046107, Apr 2009. doi: 10.1103/PhysRevE.79.046107. 20, 30, 31, 197

A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Sci.Rep.*, 2:101038, 2012. 118, 120

A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E.*, 78(4):046110, 2008. 137

J. Li, D. Chen, X. Ma, and H. Li. Unfold synchronization community structure using markov and special signature. *Int. J. Mod. Phys. B.*, 26(30):1250171, 2012. 130

X. Li and G. Chen. Synchronization and desynchronization of complex dynamical networks: an engineering viewpoint. *IEEE. Trans. Cir. Sys.*, 50:10577122, 2003. 40, 53

T.M. Liggett. Interacting partical systems-an introduction. *ICTP Lecture Notest.*, 4:17001, 2004. 31

S.L. Lim. *Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation.* PhD thesis, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia., 2010. 148, 149

S.L. Lim and P.J. Bentley. How to be a successful app developer: lessons from the simulation of an app ecosystem. In *ACM Genetic and Evolutionary Computation Conference.*, 2012. 148

S.L. Lim and A. Finkelstein. Stakerare: using social networks and collaborative filtering for large-scale requirements elicitation. In *IEEE Transactions on Software Engineering.*, 2012. 148

S.L. Lim, D. Quercia, and A. Finkelstein. Stakenet: using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32nd IEEE International Conference on Software Engineering.*, 2010. 148

S.L. Lim, D. Damian, and A. Finkelstein. Stakesource2.0: using social networks of stakeholders to identify and prioritise requirements. In *Proceedings of the 33rd IEEE International Conference on Software Engineering.*, 2011. 148

Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4):491–502, 2005. 177

J. Lorenz. A stabilization theorem for dynamics of continuous opinions. *Phys. A.*, 355(1):217–223, 2005. 20, 30, 31

J. Lorenz and D. Urbig. About the power to enforce and prevent consensus by manipulating communication rules. *Advances in Complex Systems.*, 2007. 33

W. Lu, B. Liu, and T. Chen. Cluster synchronization in networks of distinct groups of map. *Eur.Phys.J.B.*, 77(2):257–264, 2010. 130

J. MacQueen. some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, 1:281–297, 1967. 118, 123, 139

G. Mao and N. Zhang. Analysis of average shortest-path length of scale-free network. *J. App. Maths.*, page 865643, 2013. 63

P.N. McGraw and M. Menzinger. Clustering and the synchronization of oscillator networks. *Phys. Rev. E.*, 72:015101, 2005. 69

R. Merris. Laplacian matrices of graphs: a survey. linear algebra and its applications. *Line. Alg. App.*, 197:143–176, 1994. 39, 53

R. H. Mohring, H. Schilling, B. Schutz, D. Wagner, and T. Willhalm. Partitioning graphs to speed up dijkstra's algorithm. *J. Expe. Alg.*, 11:2–8, 2007. 62

S. Motoki, T. Ichigaku, and M. Hiroshi. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *Proceedings*

of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM press., 2007. 123

M. E. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701, 2002. 52

M. E. Newman. The structure and function of complex networks. *SIAM. Rev.*, 45(2):167–256, 2003. 41, 124

M. E. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E.*, 69(6):066133, 2004. 126

M. E. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23):8577–8582, 2006a. 144

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E.*, 74:036104, 2006b. 118, 123

M.E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E.*, 69(2):026133, 2004. 127, 142

M.W. Newman. *The Laplacian Spectrum of Graphs.* PhD thesis, Thesis of The Universityof Manitoba., 2000. 120, 123

Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD in-*

ternational conference on Knowledge discovery and data mining, pages 977–986. ACM, 2014. 182, 185

R. Olfati-Saber. Ultrafast consensus in small-world networks. *P.IEEE*, pages 2371–2378, 2005. 23, 34, 41

B. Parhami. Voting algorithms. *IEEE.T.Reliab.*, 43:617–629, 1994. 197

L. M. Pecora and T. L. Carroll. Synchronization in chaotic systems. *Phys. Rev. Lett.*, 64(8):821, 1990. 103, 107, 198

L. M. Pecora, T. L. Carroll, G.A. Johnson, and D.J. Mar. Fundamentals of synchronization in chaotic systems, concepts,and applications. *Chaos.*, 7(4):520–543, 1997. 103

A. Pikovsky, M. Rosenblum, and J. Kurths. *Synchronization: a universal concept in nonlinear science.* Cambridge: Cambridge University Press., 2001. 102

A. Pluchino, V. Latora, and A. Rapisarda. Changing opinions in a changing world: a new perspective in sociophysics. *Int.J.Mod.Phys.C.*, 16(04):515, 2005. 131

M.A. Porter, P.J. Mucha, M.E. Newman, C.M. Warmbrand, and A. Affiliations. A network analysis of committees in the us house of representatives. *Proc. Nat. Acad. Sci.*, 102:7057–7062, 2005. 197

A. Pothen. *Graph Partitioning Algorithms with Applications to Scientific Computing.* Kluwer Academic Press., 1997. 118

K. Pyragas. Continuous control of chaos by self-controlling feedback. *Physics Letters A*, 170(6):421–428, 1992. 103

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986. 181

G. Ren. *The geometric meaning of linear algebra.* Xi' an University of Tech., 2015. 79

M. G. Rosenblum, A. S. Pikovsky, and J. Kurths. Phase synchronization of chaotic oscillators. *Phys. Rev. Lett.*, 76(11):1804, 1996. 103

R. Saber, J. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE.*, 95(1):215–233, 2007. 19, 29, 34

Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007. 177

T.Z. Sen, A. Kloczkowski, and R L. Jernigan. Functional clustering of yeast proteins from the protein-protein interaction network. *BMC Bioinf.*, 7:355, 2006. 118, 119

J. Shao, S. Havlin, and H.E. Stanley. Dynamic opinion model and invasion percolation. *Physical review letters*, 103(1):018701, 2009. 207

R. Sinabra. *Evolutionary games on graphs:The ultimatum game.* 2008. 37

D. Stauffer. Monte carlo simulations of sznajd models. *JASSS.*, 5, 2001. 20, 30, 31, 197

D. Stauffer, A. Sousa, and C. Schulze. Discretized opinion dynamics of the deffuant model on scale-free networks. *Journal of Artificial Societies and Social Simulation.*, 2004. 33

J. J. Stocker. *Nonlinear vibrations.* New York: Interscience Publishers., 1950. 103

K. Sznajd-Weron. Sznajd model and its applications. *Acta. Phys. Pol. B.*, 5: 03239, 2005. 30

K. Sznajd-Weron and J. Sznajd. Opinion evolution in closed community. *Int. J. Mod. Phys. C.*, 11(6):1157–1165, 2000. 20, 30, 197

J. Torok, T. Iniguez, G.and Yasseri, Y.M.S. Miguel, K. Kaski, and J. Kertesz. Opinions, conflicts and consensus: Modeling social dynamics in a collaborative environment. *Phys.Rev.L.*, 110:088701, 2013. 202

D. Urbig. Attitude dynamics with limited verbalisation capabilities. *Journal of Artificial Societies and Social Simulation.*, 2003. 33

B. Van der Pol. Forced oscillations in a circuit with resistance. *Philos. Mag.*, 3: 64–80, 1927. 103

U. Von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, 2007. 120, 138

D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. 23, 43, 48, 68

C. W. Wu. *Synchronization in complex networks of nonlinear dynamical systems.* Singapore: World Scientific., 2007. 79

L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Syst. Control. Lett.*, 53(1):65–78, 2004. 23

R. Yamapi, F.M. Moukam Kakmeni, and J.B. Chabi Orou. Nonlinear dynamics and synchronization of coupled electromechanical systems with multiple functions. *Communications in Nonlinear Science and Numerical Simulation*, 12(4):543–567, 2007. 103

B. Yang, Liu D., Liu J., D. Jin, and H. Ma. Complex network clustering algorithms. *J. Softw.*, 20:54–66, 2009. 16, 123

T. Yang and L.O. Chua. Generalized synchronization of chaos via linear transformations. *International Journal of Bifurcation and Chaos*, 9(01):215–219, 1999. 103

E. Yildiz, D. Acemoglu, A. E. Ozdaglar, A. Saberi, and A. Scaglione. *Discrete opinion dynamics with stubborn agents.* SSRN.elibrary, 2011. 197

D. Yu, M. Righero, and L. Kocarev. Estimating topology of networks. *Phys.Rev.L.*, 97(18):188701, 2006. 120

T. Zhao, M.and Zhou, B. Wang, G. Yan, H. Yang, and W. Bai. Relations between average distance, heterogeneity and network synchronizability. *Phys. A.*, 371: 773–780, 2006. 69