学位論文

Stochastic processes on complex networks

(複雑ネットワーク上の確率過程)

平成 25 年 12 月　博士（理学）申請

東京大学大学院理学系研究科

物理学専攻

川本　達郎

Thesis

Stochastic processes on complex networks

Tatsuro Kawamoto
Department of Physics, University of Tokyo

December, 2013

Abstract

Stochastic processes are widely used in the analyses of complex networks, ranging from generation of a network with particular characteristics to modeling of dynamics in a network, and even to a structural analysis of a network. In this thesis, we analyze two kinds of stochastic processes on complex networks.

One is the branching processes which describe the information diffusion on a network, especially an online social network. Our goal is to understand the nature of the diffusion process on a network such as Twitter in which the diffusion of information is quite ubiquitous. In such a case, because the information diffusion occurs very frequently on a huge network, it is expected that there exist some essential differences in its dynamics from the one which had been discussed for a long time. We propose a model macroscopically describing the diffusion which occurs locally in the network, based on a data analysis of the Twitter data. We then show what are theoretically expected from the model.

The other stochastic process that we consider is a random walk. We treat the random walk as a hypothetical process on a network and use it to analyze the community structure of the network. We here focus on one of the state-of-the-art methods called map equation. While a quantity called resolution limit is known as an important quantity of a community detection method, it had only been known empirically for the map equation. In the present thesis, we analytically show its resolution limit and demonstrate its effect in several real networks.

Contents

Chapter 1

Introduction

The target of the present thesis is an arbitrary complex system which can be expressed as a network or a graph, and the aim is to understand their structures and dynamics on them. Examples of networks which we have in mind include neural networks [138], road network in a city [26], relationship network of human being [122], commercial products [108], or scientific disciplines [131], *etc.* The ingredients of a network are nodes and links connecting those nodes; in the case of a social network, for example, a node represents a person and a link indicates that the connected persons are friends.

While there exist many datasets of networks available these days, their structures and dynamics on them have not been fully investigated. Important goals of the researches in this field as physics are to develop theoretical tools which are applicable to various kinds of dataset and to figure out laws hidden in the systems. On the other hand, computer science focuses more on the feasibility, structure, expression, and mechanization of the algorithms [1].

Another important aspect of researches in physics is that they try to understand the networks in reality. One might expect that the real networks are well modeled by the regular lattices or the Erdős-Rényi random graphs [47]. It is known, however, that many real networks have properties which cannot be seen in those simple graphs, such as the so-called *scale-free property* [23] (or simply a fat-tail distribution) and the *small-world property* [150]. In this sense, such real networks are called *complex networks* [46, 53, 103].

The scale-free property means that the degree distribution (the distribution of the number of links connected to a node) of the network exhibits a power law. Let us consider the Erdős-Rényi random network, in which each pair of nodes are connected randomly with a certain probability, for comparison. Its degree distribution is a Poisson distribution, and thus its tail decays exponentially. In contrast to the Poisson distribution, the power-law distribution decays much slower, which means that the existence of the node with a large degree, or a hub, is much more probable than expected in the random network. The Barabási-Albert network [23] is the most famous model generating a scale-free network, in which a node is added at each step of generation according to the process called preferential attachment. In the process of the preferential attachment, a new node with a given degree is connected to the existing nodes with the probability proportional to their degrees. Because the variance of a power-law distribution diverges when its exponent is small, it often causes an essential difference from the result of a distribution with an exponential tail.

The other property, namely the small-world property, means that the network has a small average path length despite of its highly clustered structure. It is the property that a network such as a regular lattice does not have, while many social networks have this feature. The Watts-Strogatz model [150] is known as a fundamental model that generates a small-world network. Although this model does not have a scale-free distribution, it is often used to analyze the property of a small-world network. Other than these fundamental properties of complex networks, there may be more characteristic features in social networks.

Even though the graph theory [45, 156] and the discipline of complex network [12, 46, 103, 112] have existed for a long time, their significance have been raised even more, partly due to the thrive of the online social networks. Ever since the birth of major online social networks, *e.g.* Twitter [2], Facebook [3] *etc.*, enormous amount of people have been involved and the related social data are getting even richer and larger. Therefore, it is a very good time to explorer the structure and the dynamics of things which can be seen from their data, especially the social data.

Another factor which raise the significance and accelerates the research of complex networks is the emergence of *big data*. Big data is special in the following way [104]. A typical dataset had been a sparse (random) sampling of the total existing data. Since it only covers a tiny fraction of the total data, the result is affected by the sampling bias; in response, a number of tools in statistics have been developed in order to eliminate the sampling bias and extract out meaningful results from such data. In contrast, the big data is not just a dataset which is large in size, but the one which contains (almost) all existing data, which is free of the sampling bias. It is expected to allow us the analysis which had not been able to do before.

We can think of many issues and applications of the complex networks with the social data. In the application of marketing for example, we can argue: the relationship between the products and the consumers, to which social groups the promotion should be released, how a viral marketing campaign spreads, *etc.* It is also important to note that despite its obvious significance, quite surprisingly from the modern viewpoint, many companies had overlooked the data analysis of human activities for a long time [104]. That is partially why many results about the collective human behavior are relatively recent. In order to investigate such issues in real world, scientists need to reveal the community structure and the behavior of information spreading on the network; stochastic methods are typical tools for such purposes.

The present thesis consists of two parts. In the first half, chapters 2 and 3, we discuss the information diffusion in a complex network, especially in an online social network. Although the model here is similar to many models which had been proposed in the literature [70, 119, 144, 157] (chapter 2), there are some conceptual and technical differences (Sec. 3.3). While many models assume that the seed node of the diffusion occurs randomly in a network and estimate the spreading behavior under some assumptions on the microscopic processes, we focus on the daily diffusion rooted at a hub node. In this situation, the process occurs frequently in a fairly large scale, which enables us to observe the statistics of its microscopic process clearly. From the analysis of the diffusion data on the Twitter network which we sampled, we found that its microscopic behavior was not the same as often assumed in the literature indeed (Sec. 3.5). Thanks to the big data of Twitter, we are able to observe the statistics of events which are tiny compared to the total data. Based on this finding, we build a phenomenological model (Sec. 3.2) and theoretically analyze what is expected to be happening in reality (Sec. 3.6) and can possibly happen on the model. In particular, we focus on the chance that the daily diffusion goes viral (Sec. 3.7).

In the second half, chapters 4 and 5, we discuss the community detection of a network using a random walker. Roughly speaking, communities are sets of nodes which are densely connected relative to their neighbors. Although it sounds simple, finding communities is a difficult task. One of the difficulties is that one often needs to evaluate all possible partitions of the network to identify the optimal solution and it requires enormous amount of computational cost. Another difficulty is the lack of a unified definition of a community (Sec. 4.1). In many cases, communities are defined algorithmically as the resulting partition which optimizes a quality function. While some definitions qualify too many subgraphs as communities, there are definitions which do not qualify clearly modular subgraphs as communities if they are in a very large network. Many benchmark tests have been done to determine which one is the best to be used.

The aim here is to investigate a theoretical restriction of a method called the map equation, which shows a strong performance in a recent benchmark test (chapter 5). Although it had been said that the map equation should have the resolution limit, namely the lower bound of the detectable community size, its analytic form had not been shown. We derive the estimate of the resolution limit of the map equation analytically and reveal how its performance is restricted (Sec. 5.3). We further show that the hierarchical extension of the map equation [133] is a natural way to raise the resolution. Finally, we confirm our findings with synthetic networks and show what can be seen in real networks. In Conclusion, we will also mention a new problem emerged from our research.

Chapter 2

Diffusion models on complex networks

Before we move on to the diffusion model that we propose, we review several stochastic models which are utilized in the analysis of information diffusion in complex networks, especially focusing on branching processes. We first introduce the Galton-Watson branching process, one of the most fundamental models to describe spreading phenomena, and then explain the Bellman-Harris branching process as well as its extensions. The Bellman-Harris process is a generalization of the Galton-Watson process which takes into account a temporal effect of spreading activity of each node. While the branching processes are microscopic models, the discussion of macroscopic models is also effective. We will introduce a macroscopic model proposed by Wu and Huberman [157], which looks similar to the model that we will consider in the next chapter. Finally, we will list some other treatments of the information diffusion.

2.1 Galton-Watson branching process

The Galton-Watson process [19, 65] is the most fundamental branching process. An example of the resulting tree produced by this branching process is shown in Fig. 2.1. The process starts from a single node, namely the *seed*, and each node generates some new nodes at random. Note that the overlaps of the descendants, *i.e.* the loops, are not considered in any of the branching processes in the following.

The number of new nodes generated from the node i, Y_i, is a stochastic variable. We assume that Y_i is independent and identically distributed (i.i.d.), *i.e.*,

$$P(Y_i = k) = p_k. \tag{2.1}$$

Note that every node is generated as an active node and once it generates new nodes, it is assumed that the node never generates new nodes again, *i.e.* becomes inactive. Therefore, the branching process continues until no node generates a new node. The distance d from the seed node is called *generation*. The number of nodes in the dth generation is denoted

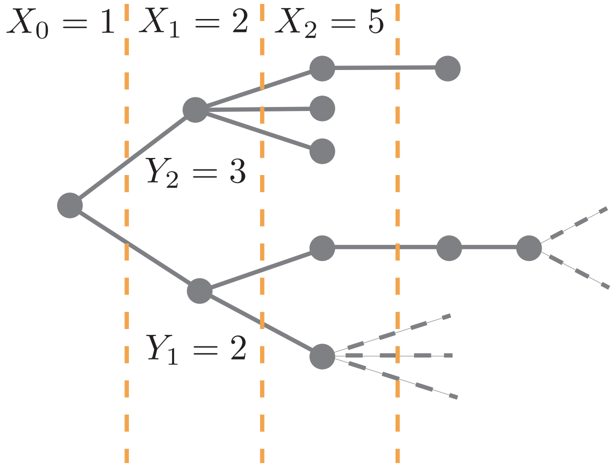


Figure 2.1: An example of the resulting tree of the Galton-Watson branching process.

by X_d. Since it is assumed that the process starts from a single seed, we have $X_0 = 1$. When the number of nodes in the dth generation is i ($X_d = i$), the transition probability that the number of nodes is j in the $(d+1)$th generation ($X_{d+1} = j$) is given by

$$p(j|i) = p\left(X_{d+1} = j | X_d = i\right) = p\left(\sum_{k=1}^{i} Y_k = j\right). \tag{2.2}$$

Major subjects of interest in the Galton-Watson branching process include the *tipping point*, the transition point above which the diffusion grows exponentially on average, and the *extinction probability*, the probability that the process eventually dies out; we review how these quantities are derived in the following.

The method of the generating function is the standard treatment in order to understand the properties of the Galton-Watson process $\{X_n\}$. We define the generating function of p_k as

$$f(s) = \sum_{k=0}^{\infty} p_k s^k \tag{2.3}$$

with

$$f_0(s) = s, \qquad f_1(s) = f(s), \qquad f_{n+1}(s) = f\left(f_n(s)\right). \tag{2.4}$$

We will see below that $f_n(s)$ is the generating function of the n-step transition probability. Using $f(s)$, we can write the generating function of the one-step transition probability $p(k|i)$ as follows:

$$\sum_{k=0}^{\infty} p(k|i) s^k = \sum_{k=0}^{\infty} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \cdots \sum_{k_i=0}^{\infty} \delta_{\sum_{\nu=1}^{i} k_\nu = k} \prod_{\nu=1}^{i} \left(p_{k_\nu} s^{k_\nu}\right) = \left[f(s)\right]^i. \tag{2.5}$$

Similarly, for the n-step transition probability $p_n(k|i)$, we can show that

$$\sum_{k=0}^{\infty} p_n(k|i) s^k = [f_n(s)]^i . \tag{2.6}$$

The proof of (2.6) is given by the mathematical induction as follows. It was shown for $n = 1$ in (2.5). We then assume that (2.6) holds for $n \geq 1$. For the $(n+1)$th step, according to the assumption of independence of X_i, we have

$$\sum_{j=0}^{\infty} p_{n+1}(j|i) s^j = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} p_n(j|k) p(k|i) s^j. \tag{2.7}$$

Since the relation (2.6) holds for the n-step transition probability, we have

$$\sum_{j=0}^{\infty} p_{n+1}(j|i) s^j = \sum_{k=0}^{\infty} p(k|i) [f_n(s)]^k. \tag{2.8}$$

The right-hand-side of (2.8) is the replacement of s in (2.5) with $f_n(s)$, and thus,

$$\sum_{j=0}^{\infty} p_{n+1}(j|i) s^j = [f\left(f_n(s)\right)]^i = [f_{n+1}(s)]^i, \tag{2.9}$$

which proves (2.6). Because we assumed that $X_0 = 1$, (2.6) means that $f_n(s)$ is the generating function of the n-step transition probability:

$$\sum_{k} P(X_n = k | X_0 = 1) s^k = \sum_{k} p_n(k|1) s^k = f_n(s). \tag{2.10}$$

The mean value of X_1 is obtained as the derivative of the generating function $f(s)$, *i.e.*,

$$f'(1) = \left.\frac{df(s)}{ds}\right|_{s=1} = \sum_{k=0}^{\infty} k p_k = \overline{k}, \tag{2.11}$$

where we assume that $\overline{k}$ is finite. For the n-step process, the mean value of X_n reads

$$f_n'(1) = \left.\frac{df_n(s)}{ds}\right|_{s=1} = \sum_{k=0}^{\infty} k p_n(k|1). \tag{2.12}$$

Since $f_n(s) = f_{n-1}(f(s))$ and

$$f_n'(1) = f_{n-1}'(1) f'(1) = \overline{k} f_{n-1}'(1), \tag{2.13}$$

we have

$$f_n'(1) = \overline{k}^n . \tag{2.14}$$

Therefore, the branching process grows exponentially for $\overline{k} > 1$ and decays for $\overline{k} < 1$, *i.e.*, $\overline{k} = 1$ is the tipping point.

The extinction probability q is expressed as

$$q = \lim_{n \to \infty} P(X_n = 0). \tag{2.15}$$

Note that when $X_{n-1} = 0$, we have $X_n = 0$. The probability $P(X_n = 0)$ that the process dies in the nth generation is given by

$$f_n(0) = p_n(0|1) = P(X_n = 0|X_0 = 1) = P(X_n = 0). \tag{2.16}$$

Hence, we have $q = f_n(0) = f_{n-1}(0)$ in the limit where $n \to \infty$. Replacing $f_n(0)$ and $f_{n-1}(0)$ in the relation $f_n(0) = f(f_{n-1}(0))$ with q, we obtain the following equation for the extinction probability:

$$q = f(q). \tag{2.17}$$

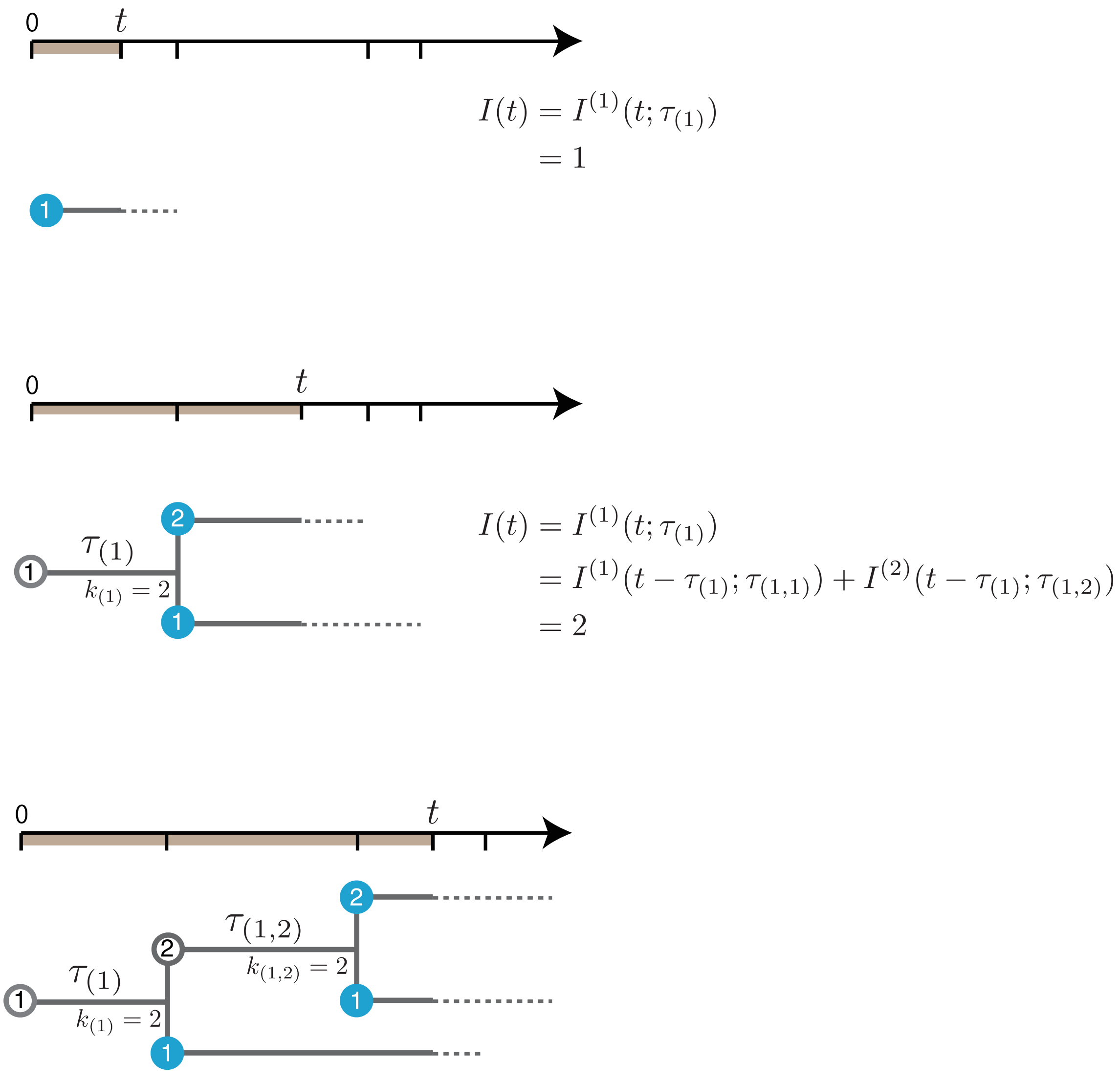
For $\overline{k} \leq 1$, $q = 1$ is the only solution. For $\overline{k} > 1$, there exists a solution $0 < q < 1$ which satisfies (2.17).

The branching process that we reviewed above arose from the statistical analysis of the extinction of family names by Galton in the nineteenth century. A family name is usually inherited from fathers to their sons, while the offsprings are males or females at random; the name gets extinct if the family has no sons. The process is sometimes called the Bienaymé-Galton-Watson branching process [4], since it is said that Bienaymé considered the same stochastic process before Galton.

As a recent application of a branching process, Liben-Nowell and Kleinberg [97] discussed the propagation of Internet chain letters. Note that there exists an underlying network in which the Internet chain letters propagate although its effect is not explicitly considered. They observed that, in spite of the small width of the branching trees, the depth of the trees can be unexpectedly deep. In order to describe this phenomenon, they constructed a model which contains the effect of the response time of each active node and some backward flow of the letters. It was later shown, however, that the regular Galton-Watson branching process is enough to describe such a narrow diffusion process [59]. It is also important to note that such a narrow diffusion is not a universal behavior of information diffusion; Wang *et al.* [148] found that their dataset of the electric communication can also be modeled as a branching process, in which the trees are typically wide and shallow.

2.2 Bellman-Harris branching process

The Bellman-Harris branching process [19,27,65] is an age-dependent process which takes into account a period of time until a node becomes viral. The viral node is the one which may generate new nodes. As in the Galton-Watson process, each node emerges as an active

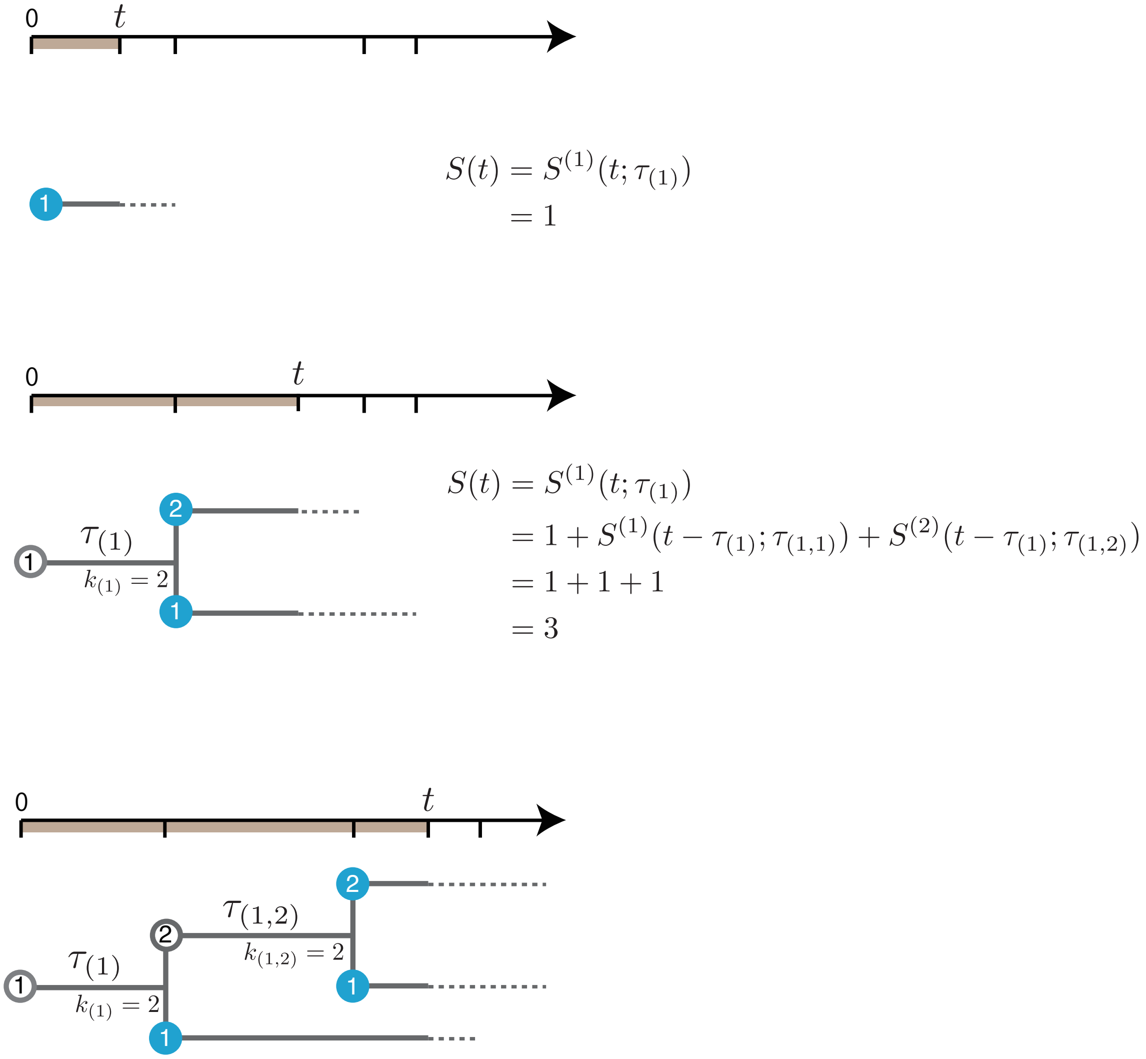


$$\begin{aligned} I(t) &= I^{(1)}(t;\tau_{(1)}) \\ &= 1 \end{aligned}$$

$$\begin{aligned} I(t) &= I^{(1)}(t;\tau_{(1)}) \\ &= I^{(1)}(t-\tau_{(1)};\tau_{(1,1)}) + I^{(2)}(t-\tau_{(1)};\tau_{(1,2)}) \\ &= 2 \end{aligned}$$

$$\begin{aligned} I(t) &= I^{(1)}(t;\tau_{(1)}) \\ &= I^{(1)}(t-\tau_{(1)};\tau_{(1,1)}) + I^{(2)}(t-\tau_{(1)};\tau_{(1,2)}) \\ &= I^{(1)}(t-\tau_{(1)};\tau_{(1,1)}) \\ &\quad + \Big(I^{(1)}(t-\tau_{(1)}-\tau_{(1,2)};\tau_{(1,2,1)}) + I^{(2)}(t-\tau_{(1)}-\tau_{(1,2)};\tau_{(1,2,2)})\Big) \\ &= 3 \end{aligned}$$

Figure 2.2: An example of the Bellman-Harris branching process. The nodes of full circles are active, while the nodes of open circles are inactive.



$$
\begin{aligned}
S(t) &= S^{(1)}(t;\tau_{(1)}) \\
&= 1
\end{aligned}
$$

$$
\begin{aligned}
S(t) &= S^{(1)}(t;\tau_{(1)}) \\
&= 1 + S^{(1)}(t-\tau_{(1)};\tau_{(1,1)}) + S^{(2)}(t-\tau_{(1)};\tau_{(1,2)}) \\
&= 1+1+1 \\
&= 3
\end{aligned}
$$

$$
\begin{aligned}
S(t) &= S^{(1)}(t;\tau_{(1)}) \\
&= 1 + S^{(1)}(t-\tau_{(1)};\tau_{(1,1)}) + S^{(2)}(t-\tau_{(1)};\tau_{(1,2)}) \\
&= 1+1+\Big(1 + S^{(1)}(t-\tau_{(1)}-\tau_{(1,2)};\tau_{(1,2,1)}) + S^{(2)}(t-\tau_{(1)}-\tau_{(1,2)};\tau_{(1,2,2)})\Big) \\
&= 1+1+(1+1+1) \\
&= 5
\end{aligned}
$$

Figure 2.3: The total number of active nodes in the example of Fig. 2.2.

node and becomes inactive after being viral. This is a non-Markovian process and is a generalization of the static and Markovian Galton-Watson branching process. Again, we assume that the branching process starts from one node and the probability that each node produces new nodes is i.i.d. As we mentioned, we introduce, as a stochastic variable, the *response time* at which a node becomes a viral node and denote the cumulative distribution of the response time as $G(\tau)$, *i.e.* the node becomes a viral node by time τ with probability $G(\tau)$. The distribution function of the response time for each node is also assumed to be i.i.d. We denote a node in the dth generation with a vector $\vec{\sigma}_d = (i_0, i_1, \ldots, i_d)$; it means that the node is the i_dth node in the dth generation which is a descendant of the i_{d-1}th node in the $(d-1)$th generation, where the i_{d-1}th node is a descendant of the the i_{d-2}th node in the $(d-2)$th generation, and so forth. Since the process always starts from a seed node, we put $i_0 = 1$. We denote $k_{\vec{\sigma}_d}$ and $\tau_{\vec{\sigma}_d}$ as stochastic variables, which represent the number of descendants and the response time of node $\vec{\sigma}_d$, respectively.

Let us now consider the number of active nodes $I(t)$ at time t. It is counted as follows. (The process is exemplified in Fig. 2.2.) If the time t is before the response time $\tau_{\vec{\sigma}_0}$ of the seed, the seed is active and there are no further branches below, which means $I(t) = 1$. If not, the seed node is no longer active and it must have produced $k_{\vec{\sigma}_0}$ descendants. We then count the number of active nodes which are rooted at each of the $k_{\vec{\sigma}_0}$ descendants, which we denote as $I^{(i_1)}(t - \tau_{\vec{\sigma}_0}; \tau_{\vec{\sigma}_1})$, where i_1 is nothing but the label of a descendant in the first generation, $\vec{\sigma}_1 = (1, i_1)$, and $\tau_{\vec{\sigma}_1}$ is the response time of the i_1th node. We do the same procedure for $I^{(i_1)}(t - \tau_{\vec{\sigma}_0}; \tau_{\vec{\sigma}_1})$ as did for the seed; we repeat it until we count all the active descendants. Hence, the number of the active nodes $I(t)$ at time t is then expressed as the following recursive relation:

$$I^{(i_d)}(t; \tau_{\vec{\sigma}_d}) = \begin{cases} 1 & \text{if } t < \tau_{\vec{\sigma}_d} \\ \sum_{i_{d+1}=1}^{k_{\vec{\sigma}_d}} I^{(i_{d+1})}(t - \tau_{\vec{\sigma}_d}; \tau_{\vec{\sigma}_{d+1}}) & \text{if } t \geq \tau_{\vec{\sigma}_d}, \end{cases} \tag{2.18}$$

where $\tau_{\vec{\sigma}_{d+1}}$ in $I^{(i_{d+1})}(t - \tau_{\vec{\sigma}_d}; \tau_{\vec{\sigma}_{d+1}})$ is a vector which has the same value as that of $\tau_{\vec{\sigma}_d}$ up to the dth element and has i_{d+1} for the $(d+1)$th element. Note that the total number of active nodes reads $I(t) = I^{(i_0)}(t, \tau_{\vec{\sigma}_0})$ using the notation in Eq. (2.18).

We assume that every $k_{\vec{\sigma}_d}$ obeys an identical distribution p_k irrespective of $\vec{\sigma}_d$ (just as in (2.1)), and therefore omit the subscripts of $k_{\vec{\sigma}_d}$ and i in the following. We omit the subscript of $\tau_{\vec{\sigma}_d}$ for the same reason. The probability distribution $P[I(t) = N]$ is then,

$$\begin{aligned} P[I(t) = N] = {} & [1 - G(t)]\delta_{N,1} + \\ & \sum_{k=0}^{\infty} p_k \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_k=0}^{\infty} \delta_{\sum_{i=1}^{k} N_i = N} \int_0^t dG(\tau) \prod_{i=1}^{k} P[I^{(i)}(t - \tau) = N_i], \end{aligned} \tag{2.19}$$

where N_i is the number of active nodes rooted at the node i. Note that $dG(\tau) = g(\tau)d\tau$, where $g(\tau) = dG(\tau)/d\tau$ is the probability distribution function. Using the generating function for $P[I(t) = N]$ and p_k, *i.e.*,

$$F(s, t) = \sum_{N=0}^{\infty} P[I(t) = N] s^N, \tag{2.20}$$

$$f(x) = \sum_k p_k\, x^k, \tag{2.21}$$

we can write Eq. (2.19) in terms of the generating functions as follows:

$$F(s,t) = s[1 - G(t)] + \int_0^t dG(\tau) f\left[F(s, t-\tau)\right]. \tag{2.22}$$

Similar argument holds for the total number of nodes $S(t)$ emerged by time t. For $S(t;\tau_{\vec{\sigma}_d})$, we have

$$S(t;\tau_{\vec{\sigma}_d}) = \begin{cases} 1 & \text{if } t < \tau_{\vec{\sigma}_d} \\ 1 + \sum_{i_d=1}^{k_{\vec{\sigma}_d}} S^{(i_d)}(t - \tau_{\vec{\sigma}_d}; \tau_{\vec{\sigma}_{d+1}}) & \text{if } t \geq \tau_{\vec{\sigma}_d}. \end{cases} \tag{2.23}$$

The count of the total number of active nodes in the process of Fig. 2.2 is exemplified in Fig. 2.3. Again, we omit the subscripts of $k_{\vec{\sigma}_d}$, $\tau_{\vec{\sigma}_d}$, and i_d in the following. The probability $P[S(t) = N]$ that the total number of nodes equals to N by time t reads

$$\begin{aligned} P[S(t) = N] = [1 - G(t)]\delta_{N,1} + & \\ \sum_{k=0}^{\infty} p_k \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_k=0}^{\infty} & \delta_{1+\sum_{i=1}^{k} N_i = N} \int_0^t dG(\tau) \prod_{i=1}^{k} P[S^{(i)}(t-\tau) = N_i]. \end{aligned} \tag{2.24}$$

Defining the generating function $\Phi(t)$ for the total size of the emerged nodes $S(t)$ as

$$\Phi(s,t) = \sum_{N=1}^{\infty} P[S(t) = N] s^N, \tag{2.25}$$

we obtain the analogous recursion relation as follows:

$$\Phi(s,t) = s[1 - G(t)] + s \int_0^t dG(\tau) f\left[\Phi(s, t-\tau)\right]. \tag{2.26}$$

In the second term, we have s in front of the integral because the delta function gives one for $N = 1 + \sum_{i=1}^{k} N_i$. The dynamics of the moments can be calculated from Eqs. (2.22) and (2.26) of the generating functions.

2.2.1 Iribarren-Moro model

Iribarren and Moro [70,71] used the Bellman-Harris branching process to analyze the data of real viral marketing campaigns which run in eleven European markets. Each of them is an online newsletter which is promoted by some invited subscribers and the campaign propagates owing to the successive recommendations. In their work, they distinguish the branching from the seed nodes, which they denote by the subscript zero, and that of the other viral nodes, which they denote by the subscript one. Therefore, instead of

Eqs. (2.22) and (2.26), we have

$$F_0(s,t) = s[1 - G_0(t)] + \int_0^t dG_0(\tau) f_0\left[F_1(s, t-\tau)\right], \tag{2.27}$$

$$F_1(s,t) = s[1 - G_1(t)] + \int_0^t dG_1(\tau) f_1\left[F_1(s, t-\tau)\right], \tag{2.28}$$

and

$$\Phi_0(s,t) = s[1 - G_0(t)] + s\int_0^t dG_0(\tau) f_0\left[\Phi_1(s, t-\tau)\right], \tag{2.29}$$

$$\Phi_1(s,t) = s[1 - G_1(t)] + s\int_0^t dG_1(\tau) f_1\left[\Phi_1(s, t-\tau)\right]. \tag{2.30}$$

The inputs are the distributions of the number of new viral nodes $p_{c,k}$ in $f_c(x)$ $(c = 0, 1)$ and the distributions of the response time $G_c(t)$. Note that the diffusion of information that they consider is a diffusion on the subscribers' network and not all of the receivers (active nodes) become the viral nodes. Hence the number of viral nodes k is determined by the probability λ_c that a node becomes viral and the number of the recommendations k' from the viral node. The average $\overline{k'}$ from a non-seed node is called the *fanout coefficient* and λ_1 is called the *transmissibility*. The probability distribution of the number of new viral nodes $p_{c,k}$ (which is denoted as $\widetilde{p}_{i,r}$ in their paper) is then given by

$$\begin{aligned} p_{c,0} &= 1 - \lambda_c, \\ p_{c,k} &= \lambda_c \, q_{c,k} \qquad \text{for } k > 0, \end{aligned} \tag{2.31}$$

where $q_{c,k}$ is the probability distribution of the number of the recommendations k from a node. We denote its average as $R_c = \sum_k k p_{c,k}$ and the value R_1 is called the *reproductive number*. Equation (2.31) can be interpreted that, although every active node has a finite response time to become viral, it also has a finite probability that the number of resulting recommendations is zero. They showed that the viral marketing data can be well described by setting $q_{c,k}$ equal to the Harris discrete distribution,

$$q_{c,k} = \frac{H_{\alpha_c \beta_c}}{\beta_c + k^{\alpha_c}}, \qquad k = 1, 2, \ldots, \tag{2.32}$$

where $H_{\alpha_c \beta_c}$ is a normalization constant. The distribution (2.32) shows the power-law behavior $p_{c,k} \sim r^{-\alpha_c}$ in its tail.

The other input is the distribution of the response time $G_c(t)$. In order to determine the distribution which describes the real data, they looked at the time dependence of the average number of active nodes $i_1(t)$, which is obtained by

$$i_1(t) = \langle I_1(t) \rangle = \left.\frac{\partial F_1(s,t)}{\partial s}\right|_{s=1}. \tag{2.33}$$

Taking the derivatives of Eqs. (2.27) and (2.28), we have

$$\begin{aligned} i_0(t) &= 1 - G_0(t) + \int_0^t dG_0(\tau) \sum_k k\, p_{0,k}\ (F_1(1,t))^{k-1}\, i_1(t-\tau), \\ &= 1 - G_0(t) + R_0 \int_0^t dG_0(\tau)\, i_1(t-\tau), \qquad (2.34) \\ i_1(t) &= 1 - G_1(t) + \int_0^t dG_1(\tau) \sum_k k\, p_{1,k}\ (F_1(1,t))^{k-1}\, i_1(t-\tau), \\ &= 1 - G_1(t) + R_1 \int_0^t dG_1(\tau)\, i_1(t-\tau), \qquad (2.35) \end{aligned}$$

where we used the normalization condition $F(1,t) = 1$. Observe that in Eqs. (2.34) and (2.35), the contribution of the number of recommendations k only appears as the average values R_0 and R_1, and therefore, the probability distribution $p_{c,k}$ is not explicitly required. Setting the probability distribution function $g_1(t) = dG_1(t)/dt$ to a lognormal distribution,

$$g_1(t) = \frac{1}{\sqrt{2\pi} t \sigma_t} \mathrm{e}^{-(\ln t - \overline{\tau}_1)^2/(2\sigma_t^2)}, \qquad (2.36)$$

they derived that

$$i_1(t) \sim \frac{1}{2(1-R_1)} \mathrm{erfc}\left(\frac{\ln t - \overline{\tau}_1}{\sqrt{2}\sigma_t} \right), \qquad (2.37)$$

which shows a fairly good agreement with the real data. An important contribution of their work is that they observed the lognormal behavior of the response time through the Bellman-Harris branching process. If an exponentially decaying distribution were used, the behavior of $i_1(t)$ would not explain the real data.

2.2.2 Vazquez model

Vazquez [144–146] generalized the Bellman-Harris branching process as follows. As illustrated in Fig. 2.4, while the branching occurs simultaneously from a node at its response time in the Bellman-Harris process, each descendant has its own generation time in Vazquez's model. As Iribarren and Moro did, in Ref. [145], Vazquez also distinguished the distribution $p_{0,k}$ of the number of descendants from the seed nodes and that from the other nodes which we denote as $p_{1,k}$. Furthermore, the maximum depth of the generation d of the process is set to D, while the generation depth of the Bellman-Harris process is unbounded. Then, the probability $P[S^{(0)}(t) = N]$ of the total number of nodes emerged

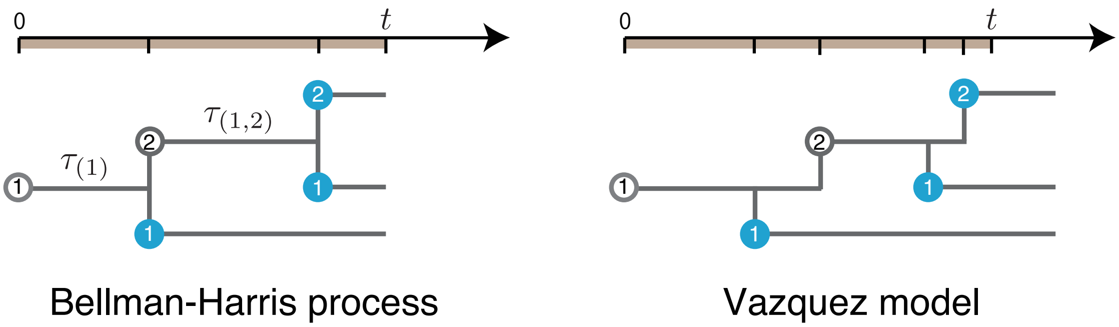


Figure 2.4: Comparison between the Bellman-Harris branching process and the Vazquez model. While the branching occurs simultaneously from a node in the Bellman-Harris branching process, it occurs at descendant's own response time in the Vazquez model.

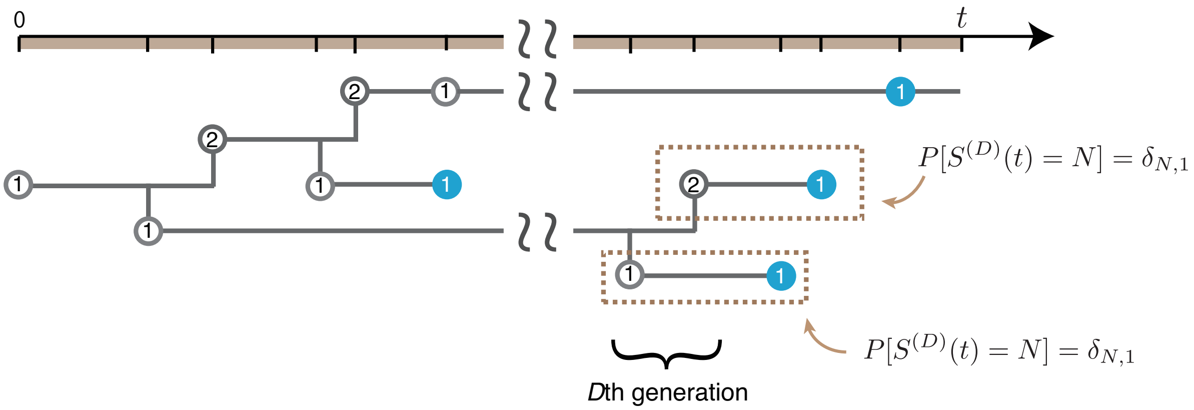

0
t
1
2
1
1
2
1
1
1
2
1
1
1
$P[S^{(D)}(t) = N] = \delta_{N,1}$
$P[S^{(D)}(t) = N] = \delta_{N,1}$
Dth generation

Figure 2.5: The boundary condition of the Vazquez model.

by time t is

$$P[S^{(0)}(t) = N] = \sum_{k=0}^{\infty} p_{0,k} \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_k=0}^{\infty} \delta_{1+\sum_{i=1}^{k} N_i = N}$$

$$\prod_{i=1}^{k} \left[\int_0^t dG_0(\tau) P(S^{(1)}(t-\tau) = N_i) + (1 - G_0(t))\delta_{N_i,0} \right], \qquad (2.38)$$

where $P(S^{(d)}(t) = N_i)$ $(0 < d < D)$ is the probability of the total number of active nodes $S^{(d)}(t)$ which is rooted at the node i in the generation d,

$$P[S^{(d)}(t) = N] = \sum_{k=0}^{\infty} p_{1,k} \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_k=0}^{\infty} \delta_{1+\sum_{i=1}^{k} N_i = N}$$
$$\prod_{i=1}^{k} \left[\int_0^t dG_1(\tau) P(S^{(d+1)}(t-\tau) = N_i) + (1 - G_1(t))\delta_{N_i,0} \right]. \qquad (2.39)$$

That is, a node which generates k descendants possesses k response times for each of the descendants (equivalently, the generation times of the descendants). We again omitted the subscripts of $k_{\vec{\sigma}_d}$, $\tau_{\vec{\sigma}_d}$, and i_d. At the Dth generation, he sets

$$P[S^{(D)}(t) = N] = \delta_{N,1} \qquad (2.40)$$

as the boundary condition, which is illustrated in Fig. 2.5. Note that this boundary condition does not mean that there always exists an active node at the generation D; Eq. (2.40) comes into the calculation when the root nodes of the nodes in the Dth generation were active. As before, in terms of the generating functions,

$$f_c(x) = \sum_{k=0}^{\infty} p_{c,k}\, x^k, \qquad (c = 0, 1) \qquad (2.41)$$

$$\Phi_d(x,t) = \sum_{N=0}^{\infty} P[S^{(d)}(t) = N]\, x^N, \qquad (2.42)$$

we obtain

$$\Phi_d(x,t) = \begin{cases} x\, f_0 \left[\int_0^t dG_0(\tau) \Phi_1(x, t-\tau) + 1 - G_0(t) \right] & \text{for } d = 0, \\ x\, f_1 \left[\int_0^t dG_1(\tau) \Phi_{d+1}(x, t-\tau) + 1 - G_1(t) \right] & \text{for } 0 < d < D, \\ x & \text{for } d = D. \end{cases} \qquad (2.43)$$

From this generating function (2.43), we have the average of the total number of the active nodes $S^{(d)}(t)$, which reads

$$S^{(d)}(t) = \left. \frac{\partial \Phi_d(x,t)}{\partial x} \right|_{x=1} = 1 + R_d \int_0^t dG(\tau) \Phi_{d+1}(x, t-\tau). \qquad (2.44)$$

The average number of new nodes $n(t)$ generated between t and $t + dt$ is obtained by the derivative of $S^{(0)}(t)$. Denoting $g_c(\tau)$ and R_c in the dth generation as $g_c^{(d)}(\tau)$ and $R_c^{(d)}$, respectively, we have

$$
\begin{aligned}
n(t) &= \frac{dS^{(0)}(t)}{dt} \\
&= R_0^{(0)} g_0^{(0)}(t) + R_0^{(0)} \int_0^t d\tau_1 g_0^{(0)}(\tau_1) \frac{S^{(1)}(t-\tau_1)}{dt} \\
&= R_0^{(0)} g_0^{(0)}(t) \\
&\quad + R_0^{(0)} \int_0^t d\tau_1 g_0^{(0)}(\tau_1) \left[R_1^{(1)} g_1^{(1)}(t-\tau_1) + R_1^{(1)} \int_0^{t-\tau_1} d\tau_2 g_1^{(1)}(\tau_2) \frac{dS^{(2)}(t-\tau_1-\tau_2)}{dt} \right] \\
&\vdots \\
&= \sum_{d=0}^{D} z_d \left(g_0^{(0)} * g_1^{(1)} * \cdots * g_1^{(d)}(t) \right),
\end{aligned} \tag{2.45}
$$

where $*$ denotes the convolution (the multiple $*$ denotes the higher-order convolution) and

$$
z_d = R_0^{(0)} \prod_{l=1}^{d} R_1^{(l)} = R_0 R_1^{d-1}, \tag{2.46}
$$

which is the average number of nodes in the generation d. In Ref. [144], he assumed that the descendant distributions and the response time distributions are different in each generation, *i.e.* $p_{c,k}$ and $g_c(\tau)$ $(c = 0, 1, 2, \ldots, D)$; within a generation, every node obeys a common distribution.

In Ref. [144], Vazquez assumed the Poisson distribution for the response time distribution $g_c(t)$ and the power-law distribution for the distribution $p_{c,k}$ of the number of active nodes. It reflects the fact that the diffusion process on a complex network should be affected by the degree distribution of the underlying network which is typically a power-law distribution. He found that $n(t)$ grows exponentially in the period where time t is much less than a characteristic time τ_0, while it becomes a polynomial growth in the period much larger than τ_0.

The exponent γ of the degree distribution $p_{c,k} = k^{-\gamma}$ of the underlying network is crucial in this analysis because he derived that $\tau_0 \to \infty$ for $\gamma \geq 3$ as the network size N_0 grows to infinity and $\tau_0 \to 0$ for $2 < \gamma < 3$ as $N_0 \to \infty$; see Ref. [144] for the detail. This result is important because, in the analysis of the susceptible/infected (SI) model, it was shown that the speed of the spreading becomes infinitely fast in the limit of an infinite network above the infection threshold on a scale-free network when its exponent γ is in the range $2 < \gamma < 3$ [25]. The result of Vazquez revealed that the growth is not actually instantaneous, but polynomial in time.

In Ref. [146], Vazquez *et al.* focused on the behavior of computer viruses which has an extraordinarily long decay time. In many real phenomena, the response time distribution $g_d(t)$ is often not a Poisson distribution, but a fat-tail distribution; indeed, they found

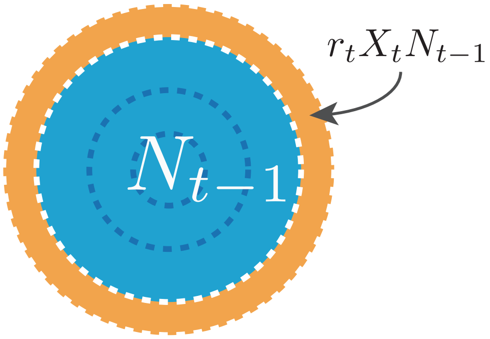


Figure 2.6: A schematic picture of the diffusion process of the Wu-Huberman model.

that the response time of E-mail activity is well fitted by a power-law distribution with an exponential cutoff. They explained why the typical decay time of computer viruses is unexpectedly large by applying the above theory with a fat-tail response time distribution.

2.3 Wu-Huberman model

The branching processes which we reviewed in the previous sections are microscopic models in which statistical laws are specified to each node. In order to explain the information spreading of the Web contents, Wu and Huberman [157] introduced a model which describes the diffusion behavior from a macroscopic point of view. They focused on the statistics of a web service called Digg [5]. Digg is a news aggregator on which users can submit news stories. In Digg, users can vote, or *digg*, for a story; the more a story earns the number of diggs, the more attention it collects. They found that the distribution of the popularity, or the number of diggs, is a lognormal distribution. In order to explain this, instead of building a microscopic model of each user, they modeled the accumulation of the popularity as a random multiplicative process.

Let N_t represent the number of users who have received a story by time t; they equate it to the number of diggs that the story received. Here, time is treated as a discrete variable. Introducing a positive random variable $\widetilde{X}_t$, they assumed that number of the new users who know the story at moment t is expressed as $\widetilde{X}_t N_{t-1}$. Thus, we have the total number of users who know the story N_t as

$$N_t = \left(1 + \widetilde{X}_t\right) N_{t-1}. \tag{2.47}$$

Note that the number of new receivers $\widetilde{X}_t N_{t-1}$ does not solely depend on the number of receivers at the previous time step, but on the total number of receiver N_{t-1} by the previous time step (Fig. 2.6). In order to take account of the fact that the novelty of the story decreases in time, they separate $\widetilde{X}_t$ into the deterministic decaying factor r_t ($r_1 = 1$

and $r_\infty = 0$) and a positive i.i.d. random variables X_t with mean μ and variance σ^2, *i.e.*,

$$N_t = (1 + r_t X_t)\, N_{t-1} = N_0 \prod_{s=1}^{t} (1 + r_s X_s)\,, \tag{2.48}$$

where N_0 is the initial population. When the time step is short so that X_t may be small, Eq. (2.48) can be approximated as

$$N_t \approx N_0 \prod_{s=1}^{t} \mathrm{e}^{r_s X_s} = N_0 \exp\left(\sum_{s=1}^{t} r_s X_s\right). \tag{2.49}$$

Taking the logarithm of both sides, we can express it as

$$\log N_t - \log N_0 \approx \sum_{s=1}^{t} r_s X_s. \tag{2.50}$$

The distribution of N_t therefore approaches to a lognormal distribution because of the central limit theorem.

In order to justify their model, they further measured the ratio of the mean and the variance of $\log(N_t/N_0)$. If their model is correct, the ratio should be

$$\frac{\langle \log N_t - \log N_0 \rangle}{\mathrm{var}\,(\log N_t - \log N_0)} = \frac{\sum_{s=1}^{t} r_s \mu}{\sum_{s=1}^{t} r_s \sigma^2} = \frac{\mu}{\sigma^2}, \tag{2.51}$$

where $\langle \cdots \rangle$ is the average with respect to the stories and $\mathrm{var}(\cdots)$ is their variance. Although they did not obtain a straight line for the plot of $\langle \log N_t - \log N_0 \rangle$ against $\mathrm{var}\,(\log N_t - \log N_0)$, they confirmed that the relationship was almost linear. They also found that the decay factor r_t is a stretched-exponential function of t. In order to see this, they used the following form of r_t which can be obtained from (2.50):

$$r_t = \frac{\log N_t - \log N_{t-1}}{\log N_1 - \log N_0}, \tag{2.52}$$

where the denominator $\log N_t - \log N_0$ is to normalize r_1 to unity.

Their approach is distinct from the other models of the information diffusion because the diffusion occurs owing to the collection of attention. Such a diffusion of popularity was also found in other services such as Wikipedia, Bugzilla, Essembly [155] and the count of HTML views [159].

2.4 Other models of information diffusion

Many models have been proposed to describe the diffusion of information, or the word-of-mouth, especially in the context of viral marketing and the propagation of news contents on the Web. Most of them use the analogy from the models of epidemics, *e.g.*, the percolation models, the susceptible-infected-recovered (SIR)-type models, and the branching processes.

The Daley-Kendal model [43] is a classical SIR-type model for rumor spreading, in which they modified the dependence of the informed (or infected, in terms of epidemiology) people from the original model of Kermark and Mckendrick [79]; there are many related works, *e.g.*, Refs. [10, 56, 106, 165]. In the last decade, the behavior of an epidemic model on complex networks had been studied energetically, mainly by physicists [25, 31, 119] from the mathematical point of view. It is important to note that the model of the (bond) percolation is equivalent to the SIR model when we are interested in the static properties [62, 110].

In the context of viral marketing, Goldenberg *et al.* modeled it as a percolation model [58], while Leskovec *et al.* [90] considered a model which takes account of the saturation of influence from a node. Other than the branching processes which we explained above, a more detailed branching process which takes account of the effect of marketing activities was also considered [143]. For the word-of-mouth in online social networks, people are not only informed by the neighbors of the network, but also by some external sources such as TV and newspapers. In the case of the contents with general interest, it is expected that they collect large attention outside the network as well, and therefore such an effect would be significant. Myers *et al.* [109] modeled such an out-of-network effect in the spreading of information and applied to the diffusion on the Twitter network. There exist many other approaches to the growth of popularity on the Web. Crane and Sornette [42] observed the decay of the time evolution of the views on YouTube videos after endogenous and exogenous bursts and found that the relaxation exponent can be classified into three types. Yang and Leskovec [161] developed a method of data clustering for time series and classified the shape of the popularity evolution in six types in their Twitter data.

Most studies focus on large-scale cascades of information and demonstrate the performance of their models with some real data. While the investigation of the large-scale cascades is important without any doubt, one of the reasons is perhaps because it is often easy to capture the statistical behavior even when the data is collected by a random sparse sampling method. Although the dataset that we use in the next chapter is not very large, because we make use of a dataset which is complete in principle, *i.e.* not sparsely sampled, our analysis is qualitatively different from those by the sparse sampling methods.

Chapter 3

Local model of information diffusion in online social networks

This chapter is one of the main chapters of the present thesis. We propose a stochastic model [76] which describes the information diffusion in online social networks. In our model, we concentrate on the diffusion phenomenon which spreads from a certain hub node. In this sense, our local diffusion model of information is distinct from the other models considered in the literature.

Before we tackle the problem of local diffusion phenomena, since the center of our interest is the diffusion in the network such as Twitter and Facebook, we first review the researches on Twitter and Facebook. Then, after introducing our local diffusion model, we will compare it with the other models which we described in the last chapter. We confirm the plausibility of our model and determine the distribution of the stochastic variable, the retweet rate, in our model from the data analysis of the Twitter data that we collected. The rest of the chapter is devoted to the theoretical analysis on the local diffusion model. We first consider the behavior of the spreading when each diffusion step is uncorrelated to each other. We then consider the case of correlated diffusion process and discuss how the threshold of viral diffusion is altered owing to the strength of the correlation.

3.1 Online social networks and the researches with their data

There exist many online social networks, *e.g.* Twitter, Facebook, Google+, LinkedIn, Instagram, Foursquare, Tumblr, Pinterest, Digg, *etc.* Although there are many differences among the online social networks, they have some common basic structures; each user has his or her own account of a web service and the users can share the information with whom they are connected to, and thus the users form a network. The information they share are blogs, pictures, movies, personal profiles, news stories which they found, *etc.* While the connections such as *friend* in Facebook are undirected, some connections such as

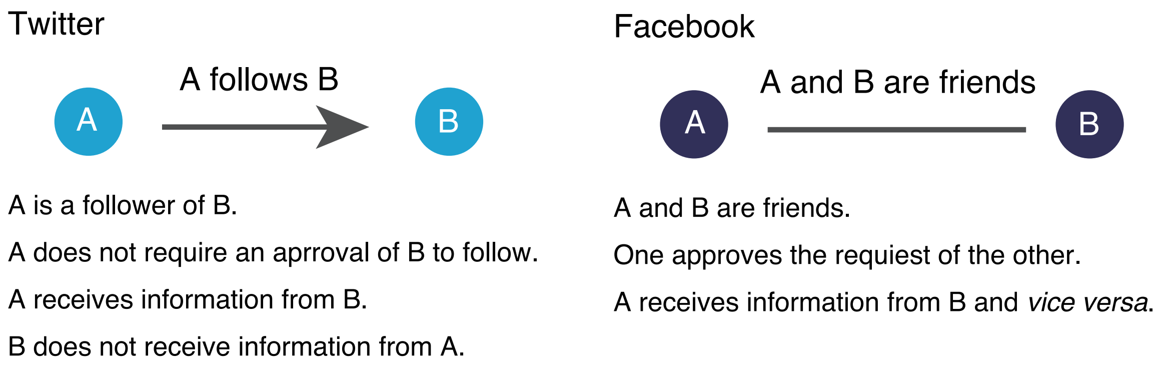

Twitter
A follows B
A
B
A is a follower of B.
A does not require an aprroval of B to follow.
A receives information from B.
B does not receive information from A.
Facebook
A and B are friends
A
B
A and B are friends.
One approves the requiest of the other.
A receives information from B and vice versa.

Figure 3.1: The ways of connection between users in Twitter and Facebook.

follower in Twitter are directed. That is, when user A follows user B but B does not follow A, then A receives the information of user B although the user B does not receive information from A (See Fig. 3.1). In many online social networks, there exists a function to spread the received information to the neighboring users, *e.g. retweet* in Twitter and *share* in Facebook; this is basically what causes the information diffusion in online social networks. The other common function that many online social networks has is the one to rate the received information, *e.g. favorite*, *like*, or *digg*, which may result in raising the popularity of the content; note, however, that there is no explicit effect caused by it in the case of Twitter, while it helps to collect popularity in Digg explicitly. The function of like in Facebook is more complicated.

Among the researches focusing on the major online social networks, most of them use the Twitter data owing to the easiness of accessibility to its data, while there are not so many papers published using the Facebook data because most of its data is not publicly available. There exist numbers of data analysis on Twitter, especially at the early stage, *e.g.*, semantic analysis [152], statistics of some basic variables such as the number of follow/followers, the number of posts, and the number of retweets [69,72,82,83,140,158], characteristics of the posts and communication patterns [39, 67, 141, 160], influence of users [21,35], *etc.* The researches on Facebook have been done mainly by the people who are allowed to access to its data. A striking one is an experiment done by Bakshy *et al.* [22]; they altered a function of Facebook itself for a short period of time in order to investigate the behavior of the information diffusion on its network.

There are already many application with the Twitter data, *e.g.*, the analysis in politics [40], elections [33], stock market [32], spammer detection [28], and the prediction of the spread of diseases [16,96,120]. For the studies of the information spreading, some authors proposed the inference methods for the volume of the spreading [55,60,68]. How links are formed is also an important problem and studied energetically [105,129,162,163].

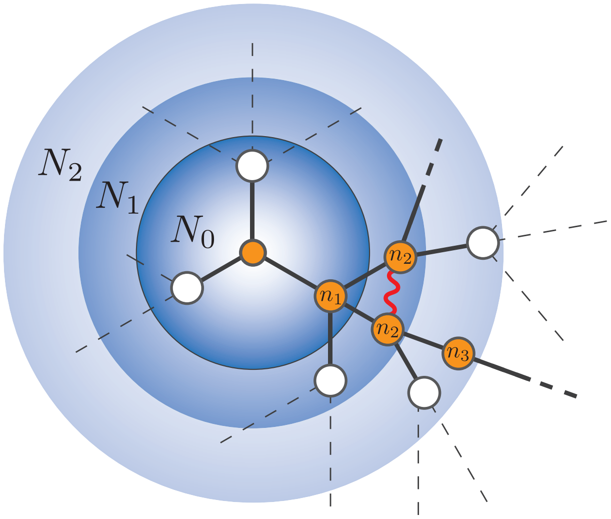


Figure 3.2: Information diffusion on an online social network. The node at the center represents the seed and the linked nodes can receive the information. A solid line represents that the information has diffused through the link. While we take account of the over-counting of nodes such as the one illustrated by the wavy line in the data analysis, we ignore it in the sections of theoretical analyses, *i.e.* we assume a tree structure.

3.2 The local diffusion model

We now explain the local diffusion model [76] in the context of the Twitter network. The information which diffuses in the Twitter network is a post of a user, which is called a *tweet*. As we mentioned above, the tweet diffuses in the network owing to the function called *retweet*. Figure 3.2 shows a schematic picture of the tweet diffusion process. Whenever a user generates a tweet, it will be sent to N_0 followers of the tweet owner, whom we call users in the zeroth generation. Next, when n_1 users out of N_0 followers retweet, the original tweet will be sent to the followers of the n_1 retweeters; we call them users in the first generation. We label the number of the receivers in the first generation as N_1. Such a chain of diffusion of a tweet continues until people stop retweeting or all the followers in the last generation are the users who already received the tweet. We will refer to the total number of receivers as $N_{\text{tot}} = \sum_{g=0}^{\infty} N_g$ and the total retweet count as $n_{\text{RT}} = \sum_{g=1}^{\infty} n_g$, where g stands for the label of the generation. While N_0 is simply the number of the followers of the seed account, N_g for $g \geq 1$ reads

$$N_g = \sum_{f=1}^{n_g} k_f - c_g, \tag{3.1}$$

where f stands for the label of each retweeter and k_f stands for the number of his or her followers. The factor c_g is the number of over-counting of the followers (*e.g.*, the wavy line in Fig. 3.2). In the case where the network is close to the tree structure and the distribution of the number of followers is homogeneous, *i.e.* there is no strong local structure such as communities, we can employ the approximation

$$N_g \simeq n_g \sum_{k=0}^{\infty} k\, p_g(k) =: n_g \overline{k}_g, \tag{3.2}$$

where k and $p_g(k)$ are the number of the followers of the retweeters in the $(g-1)$th generation and its distribution, respectively. A more precise argument is done later in Sec. 3.4.

Let us next estimate the number of the retweeters, n_g. Since there are N_{g-1} candidates to generate the retweeters in the gth generation, we assume

$$n_g = \beta_g N_{g-1}, \tag{3.3}$$

where β_g is a variable which we call the retweet rate. Although β_g is a discrete variable because n_g and N_{g-1} are integers, we treat it as if it were a continuous variable. In Sec. 3.5, we will observe that the retweet rate has a distribution over many incidents of tweet diffusion. We therefore regard β_g as a continuous stochastic variable hereafter.

Combining Eqs. (3.2) and (3.3), we have

$$N_m = J_m N_{m-1} = \cdots = \prod_{g=1}^{m} J_g N_0, \tag{3.4}$$

$$n_m = \beta_m N_{m-1} = \cdots = \beta_m \prod_{g=1}^{m-1} J_g N_0, \tag{3.5}$$

where

$$J_g = \beta_g \overline{k}_g \qquad (g \geq 1), \tag{3.6}$$

which is a stochastic variable because β_g is a stochastic variable. In the framework of the branching process, β_g, $\overline{k}_g$, and J_g for $g \geq 1$ correspond to the transmissibility of a generation (not of a single user), the fanout coefficient, and the reproductive number of a generation, respectively. Although the probability distribution of J_1 may strongly depend on the characteristics of the seed account, J_g for $g \geq 2$ are expected to obey a common probability distribution; indeed, the first generation and the rest of the generations are distinguished in the models of Refs. [71,145] as well. Therefore, the number of viewers of the tweet in each generation, N_g, is expressed as a random multiplicative process because of the hierarchical structure of the followers. We do not consider the time dependence of the retweet rates for simplicity. It is a plausible assumption for the daily tweet diffusion since most of the tweets finish diffusing very quickly [83]. In spite of the highly clustered structure of the online social network, the validity of the tree approximation for a diffusion path is discussed and empirically proven in Refs. [71,110,146] and the references therein.

We can regard our model as a macroscopic modeling of the Galton-Watson process, which is even simpler. What is distinct from other models is that, we can directly observe

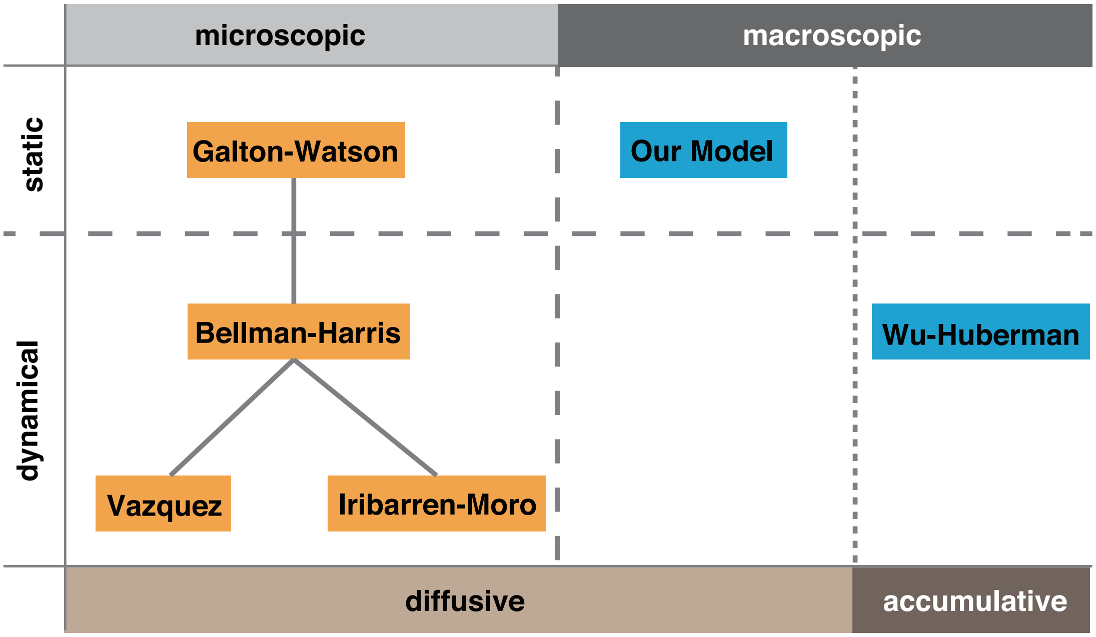


Figure 3.3: Classification of the diffusion models. The model that we consider is static, macroscopic, and diffusive.

the statistics of the retweet rates β_g thanks to the complete data of Twitter; the accessibility is a characteristic of Twitter data and measuring such a stochastic variable would not be possible with a sparse sampling data. In section 3.5, we will directly observe the statistics of the retweet rates β_g and confirm that our modeling is indeed plausible.

We give a couple of comments on the details of functions in Twitter. First, the word retweet is sometimes used for two different meanings in the literature. The retweet button was first introduced at the end of 2009. Retweeting used to be simply a name of custom on Twitter to transfer the tweet of another user; it is called *informal retweet* nowadays, while the retweet by clicking the retweet button is called *formal retweet*. Galuba *et al.* [55] also introduced a model of tweet diffusion in a very different manner from ours, but they limited themselves to the URL-embedded tweets and counted the informal retweets, whereas we analyze tweets in general and count the formal retweets in our model. The other comment is that the chain of retweets among the followers is not the only way in which tweets diffuse; as long as the tweet owner is a public user, anyone can read the tweet and any user has the right to retweet. Although, we do not treat such other processes in our model. We only count the formal retweets by non-private users because we believe that it gives the major contribution to the daily tweet diffusion.

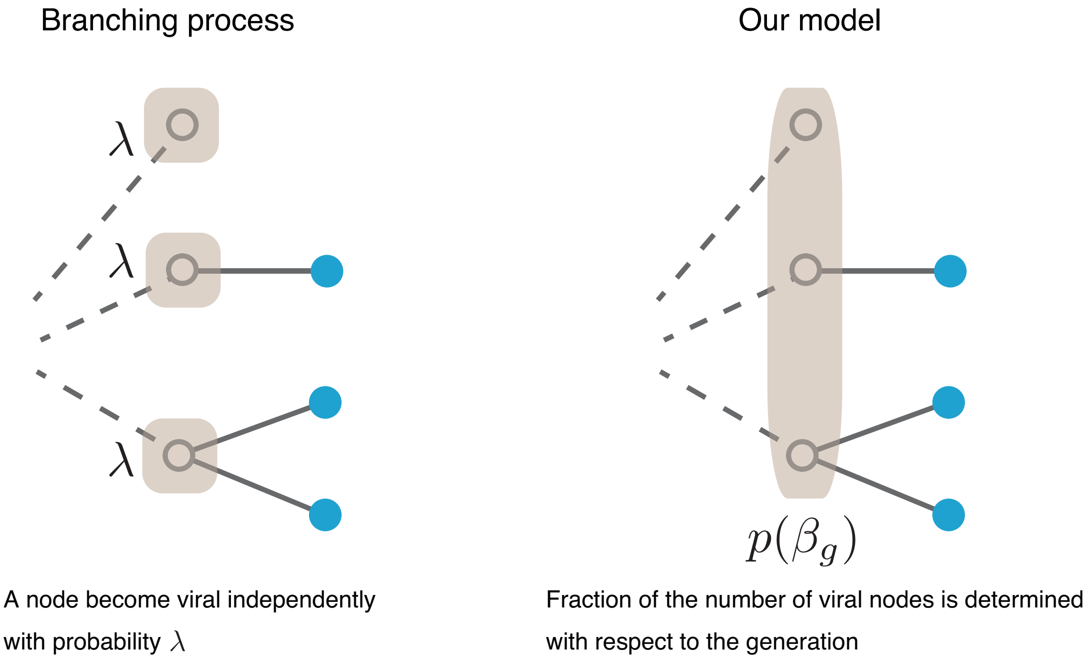

Branching process
λ
λ
λ
A node become viral independently with probability λ
Our model
$p(\beta_g)$
Fraction of the number of viral nodes is determined with respect to the generation

Figure 3.4: The difference between a branching process and our model.

3.3 Comparison with other diffusion models

Now we compare the model that we proposed with the other models of information diffusion which we listed in the previous chapter. The classification diagram of the models is shown in Fig. 3.3. As we stated above, our model is a local model in which the diffusion is always rooted at a predetermined hub, while other models select a seed node at random. Other than that, the differences are whether the model is microscopic or macroscopic, static or dynamical, and diffusive or accumulative.

The branching processes are microscopic models in which a stochastic variables is assigned to each node, while our model and the Wu-Huberman model are macroscopic in which a stochastic variable is assigned to each generation. The Galton-Watson process, the most fundamental model of the branching process, is a static model, *i.e.* the response time is not taken into account. The Bellman-Harris process is an age-dependent dynamical model which consider the response time of each node and the model of Iribarren and Moro considered the response time with a fat-tail distribution. The model of Vazquez is also the Bellman-Harris-type process in which each node becomes a viral node at its own time. In both model of Iribarren and Moro and that of Vazquez, the degree distribution of a viral node is assumed to be a power-law distribution.

In the model of Iribarren and Moro, each node has a fixed probability λ_c, *i.e.* the transmissibility, to become a viral node. Note that, even though the transmissibility is not explicitly considered as in the Galton-Watson process, it can be implemented easily by defining the distribution p_k as in Eq. (2.31). Since the reproducing process of each node is assumed to be independent in any branching process, it implies that the number of viral nodes among a generation obeys the following binomial distribution:

$$p(n_g; N_{g-1}) = \binom{N_{g-1}}{n_g} \lambda_c^{n_g} (1 - \lambda_c)^{N_{g-1} - n_g}. \tag{3.7}$$

On the other hand, the ratio of the number of viral nodes compared to the number of receivers within a generation, *i.e.* the retweet rate β_g, is allowed to obey an arbitrary distribution in our model (see Fig. 3.4); as we will show below, the distribution of the retweet rate of a generation is very different from the binomial distribution indeed. Moreover, we will consider the case in which the retweet rates are correlated later.

The model by Wu and Huberman is a macroscopic model with the decaying factor r_t of the diffusion rate. Although their model looks similar to ours, their model is for a physically different process. In Digg that they focused on, when a news story collects some popularity, it earns a higher probability to be peered, which results in collecting more popularity; thus, their model is an accumulative process rather than a diffusive process. In contrast, the information diffusion in Twitter occurs owing to the cooperation of the retweeting users and the total amount of the spreading cannot be regarded as the source of higher popularity.

Although our model is distinct from other information diffusion models because it assigns a stochastic variable for each generation, the idea of dividing the nodes in the network into the generations exists in other topics in the literature [24, 41, 52, 111] as well; it is often called *ring*. For example, Baronchelli *et al.* [24] analyzed the mean first passage time of a random walker using the ring structure. The loop effect, or the back flow from the point of view of the ring, is essential in their model, which is distinct from the situation that we consider.

3.4 Justification of the formulation with generating functions

In Sec. 3.2, we expressed the number of receivers of the tweet as Eq. (3.2). In the following sections, we will calculate the statistical averages of the stochastic variables n_g and N_g for $g \geq 1$. In this section, we show that our estimation in Eq. (3.2) is justified when we calculate their mean values. As above, we neglect the existence of the loops and every node has a common degree distribution p_k.

First of all, the probability $p(n_g)$ of the number of retweeters n_g in the gth generation can be written as

$$p(n_g) = \sum_{n_{g-1}=0}^{\infty} \cdots \sum_{n_1=0}^{\infty} p(n_g|n_{g-1}) p(n_{g-1}|n_{g-2}) \cdots p(n_2|n_1) p(n_1; N_0), \tag{3.8}$$

where

$$p(n_1; N_0) = \int d\beta_1 \, p(\beta_1) \delta_{n_1 = \lfloor \beta_1 N_0 \rfloor}. \tag{3.9}$$

The symbol $\lfloor x \rfloor$ ($x \in \mathbb{R}$) represents the largest integer less than x. Defining a generating function $f_\beta(s)$ as

$$f_\beta(s) = \sum_{k=0}^{\infty} p_k s^{\beta k}, \tag{3.10}$$

we have

$$\begin{aligned}
\sum_{n_g=0}^{\infty} p(n_g|n_{g-1}) s^{n_g} &= \sum_{n_g=0}^{\infty} \int_0^\infty d\beta_g \sum_{k_1=0}^{\infty} \cdots \sum_{k_{n_{g-1}}=0}^{\infty} \\
&\qquad \times \delta_{n_g = \lfloor \beta_g \sum_\nu k_\nu \rfloor} \left(p(\beta_g) \prod_{\nu=1}^{n_{g-1}} p_{k_\nu} \right) s^{\beta_g \sum_\nu k_\nu} \\
&= \int_0^\infty d\beta_g \, p(\beta_g) \prod_{\nu=1}^{n_{g-1}} \left[\sum_{k_\nu=0}^{\infty} p_{k_\nu} s^{\beta_g k_\nu} \right] \\
&= \int_0^\infty d\beta_g \, p(\beta_g) \left[f_{\beta_g}(s) \right]^{n_{g-1}}. \qquad (3.11)
\end{aligned}$$

In the above calculation, we neglected the fact that n_g is an integer and treated its sum as if it were an integral; we use this trick in the following repeatedly. Using (3.11), we can write down the generating function of n_g as follows:

$$\begin{aligned}
F(s) &\equiv \sum_{n_g=0}^{\infty} p(n_g) s^{n_g} \\
&= \sum_{n_g=0}^{\infty} \sum_{n_{g-1}=0}^{\infty} \cdots \sum_{n_1=0}^{\infty} p(n_g|n_{g-1}) s^{n_g} p(n_{g-1}|n_{g-2}) \cdots p(n_2|n_1) p(n_1; N_0) \\
&= \int_0^\infty d\beta_g \, p(\beta_g) \sum_{n_{g-1}=0}^{\infty} \cdots \sum_{n_1=0}^{\infty} \left[f_{\beta_g}(s) \right]^{n_{g-1}} p(n_{g-1}|n_{g-2}) \cdots p(n_2|n_1) p(n_1; N_0) \\
&= \int_0^\infty d\beta_g \int_0^\infty d\beta_{g-1} \, p(\beta_g) p(\beta_{g-1}) \\
&\qquad \sum_{n_{g-2}=0}^{\infty} \cdots \sum_{n_1=0}^{\infty} \left(f_{\beta_{g-1}} \left[f_{\beta_g}(s) \right] \right)^{n_{g-2}} p(n_{g-2}|n_{g-3}) \cdots p(n_2|n_1) p(n_1; N_0) \\
&\;\;\vdots \\
&= \int_0^\infty d\beta_g \cdots d\beta_2 \, p(\beta_g) \cdots p(\beta_2) \sum_{n_1=0}^{\infty} \left[f_{\beta_2,\ldots,\beta_g}(s) \right]^{n_1} p(n_1; N_0) \\
&= \int_0^\infty d\beta_g \cdots d\beta_1 \left[f_{\beta_2,\ldots,\beta_g}(s) \right]^{\lfloor N_0 \beta_1 \rfloor}, \qquad (3.12)
\end{aligned}$$

where we denoted

$$f_{\beta_x,\beta_{x+1},\ldots,\beta_g}(s) := f_{\beta_x}\left[f_{\beta_{x+1},\ldots,\beta_g}(s)\right] \qquad \text{for } x = 2, 3, \ldots, g-1. \tag{3.13}$$

Note that when $\beta_d = 1$ for any d, our model reduces to the Galton-Watson branching process. Using Eq. (3.12), we can calculate the mean value of n_g as

$$\begin{aligned}
\langle n_g \rangle = \left.\frac{dF(s)}{ds}\right|_{s=1} &= \int_0^\infty d\beta_g \cdots d\beta_1 N_0 \beta_1 \left[f_{\beta_2,\ldots,\beta_g}(1)\right]^{\lfloor N_0\beta_1 \rfloor - 1} \left.\frac{d}{ds} f_{\beta_2,\ldots,\beta_g}(s)\right|_{s=1} \\
&= \int_0^\infty d\beta_g \cdots d\beta_1 N_0 \beta_1 (\beta_2 \overline{k}) \cdots (\beta_g \overline{k}) \\
&= N_0 \overline{k}^{g-1} \langle \beta_1 \cdots \beta_g \rangle,
\end{aligned} \tag{3.14}$$

where $\langle \cdots \rangle$ stands for the statistical average with respect to the stochastic variables. In the above calculation, we used $df_\beta(1)/ds = \beta\overline{k}$ and the normalization condition $f_\beta(1) = 1$.

We can calculate the mean value of the number of receivers N_g similarly. The generating function of N_g reads

$$\begin{aligned}
G(s) \equiv \sum_{N_g=0}^\infty p(N_g) s^{N_g} &= \sum_{N_g=0}^\infty \sum_{n_g=0}^\infty p(N_g|n_g) p(n_g) s^{N_g} \\
&= \sum_{N_g=0}^\infty \sum_{n_g=0}^\infty \sum_{k_1=0}^\infty \cdots \sum_{k_{n_g}=0}^\infty \delta_{N_g=\sum_\nu k_\nu} \prod_{\nu=1}^{n_g} \left(p(k_\nu) s^{k_\nu}\right) p(n_g) \\
&= \sum_{n_g=0}^\infty \left[f_{\beta=1}(s)\right]^{n_g} p(n_g).
\end{aligned} \tag{3.15}$$

Then, we have the mean value $\langle N_g \rangle$ as follows:

$$\begin{aligned}
\langle N_g \rangle &= \left.\frac{dG(s)}{ds}\right|_{s=1} = \sum_{n_g=0}^\infty p(n_g) n_g \left[f_{\beta=1}(1)\right]^{n_g-1} \left.\frac{df_{\beta=1}(s)}{ds}\right|_{s=1} \\
&= \langle n_g \rangle \overline{k} \\
&= N_0 \overline{k}^g \langle \beta_1 \cdots \beta_g \rangle.
\end{aligned} \tag{3.16}$$

Now, we see that Eqs. (3.14) and (3.16) are consistent with Eqs. (3.4) and (3.5). We also see that the treatment in Eqs. (3.4) and (3.5) are not precise when we calculate higher moments of n_g and N_g.

3.5 Data analysis for the retweet rate β_g

Using the data sampled by the tool Twitter API [6], we directly observed the behaviors of β_1 and β_2. We chose The New York Times (@nytimes) and Reuters Top News (@Reuters) for the seed accounts and sampled the diffusion data with $n_2 > 0$. The data are summarized in Table 5.1.

3.5.1 Possible errors, selection of the seed accounts, and restrictions

There are some inevitable errors in our data. We cannot sample the data of private users and there might be some miscounts in n_g because the follow-followed relation might have changed by the time we sampled the data. In order to sample the data as accurately as possible, we need to select the seed accounts carefully; we chose the seed accounts which tweet frequently and the number of whose followers are not changing rapidly so that we can expect the network around the seed account is almost static during the period of sampling. In order to see the statistical behavior clearly, it is good to choose an account with a large number of followers and high retweet rates. (We omitted the data with more than 800 retweets because Twitter API seems to fail to count the retweets correctly in such cases.) In the data analysis of the retweet rate β_g, we take into account the factor of over-counting c_g in Eq. (3.1), and thus we do not assume a tree structure nor the homogeneity of the distribution of the followers.

3.5.2 Result

Figures 3.5(a) and 3.5(b) show the histograms of β_1 and their normal Q-Q plots [153]. They show that the retweet rate β_1 seems to obey lognormal distributions with slight additive shifts, *i.e.*

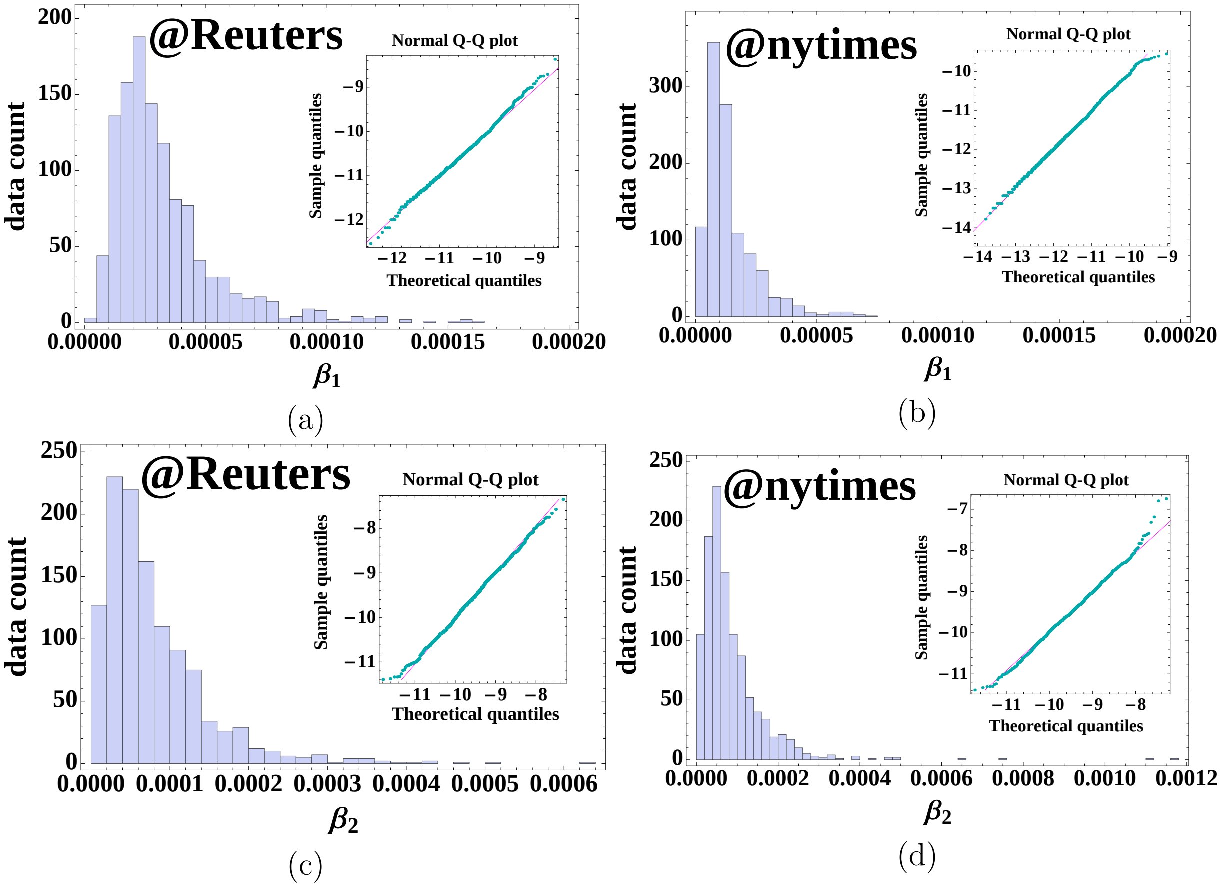
$$\beta_1 = \mathrm{e}^{\omega_1} + \delta_1, \tag{3.17}$$

where ω_1 obeys Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$ with μ_1 being the mean and σ_1^2 being the variance of $g = 1$. For β_1, the mean μ_1 and the variance σ_1^2 seem to depend strongly on the character of the seed account. The slight additive shift might be due to the systematic activities by Twitter bots.

We expect that the retweet rate β_2 also obeys lognormal distributions with slight additive shifts. Figures 3.5(c) and 3.5(d) show the histograms of β_2 and their normal Q-Q plots; they indeed indicate the lognormal behaviors. For β_2, the mean μ_2 and the variance σ_2^2 are very close for both of the seed accounts; it seems to be plausible to model in such a way that the retweet rate β_g obeys a common probability distribution for $g \geq 2$.

In Table 5.1, we listed the averages of the over-counting of the users in the first generation, *i.e.* $c_1/\sum_{f=1}^{n_1} k_f$ in Eq. (3.1). The over-counting of users are less than 5% on average, and thus the networks around the seed accounts have almost the tree structures. Although it is still doubtful whether the tree-structure approximation is appropriate in all generations, it is hard to imagine a drastic qualitative change to the diffusion phenomenon due to the loop correction since there is no back flow.

Since we are fixing the seed account, N_0 is a constant and the distribution of β_1 is proportional to that of n_1. The number of followers in the first generation, N_1, and the number of retweeters among them, n_2, can take different values for each sample. As are shown in Figs. 3.6(c) and 3.6(d), both of them obey lognormal distributions and they are



(a)

(b)

(c)

(d)

Figure 3.5: The histograms of the retweet rates and the normal Q-Q plots of the logarithms of the retweet rates. (a) and (c) are of β_1 and β_2 for @Reuters. (b) and (d) are of β_1 and β_2 for @nytimes.

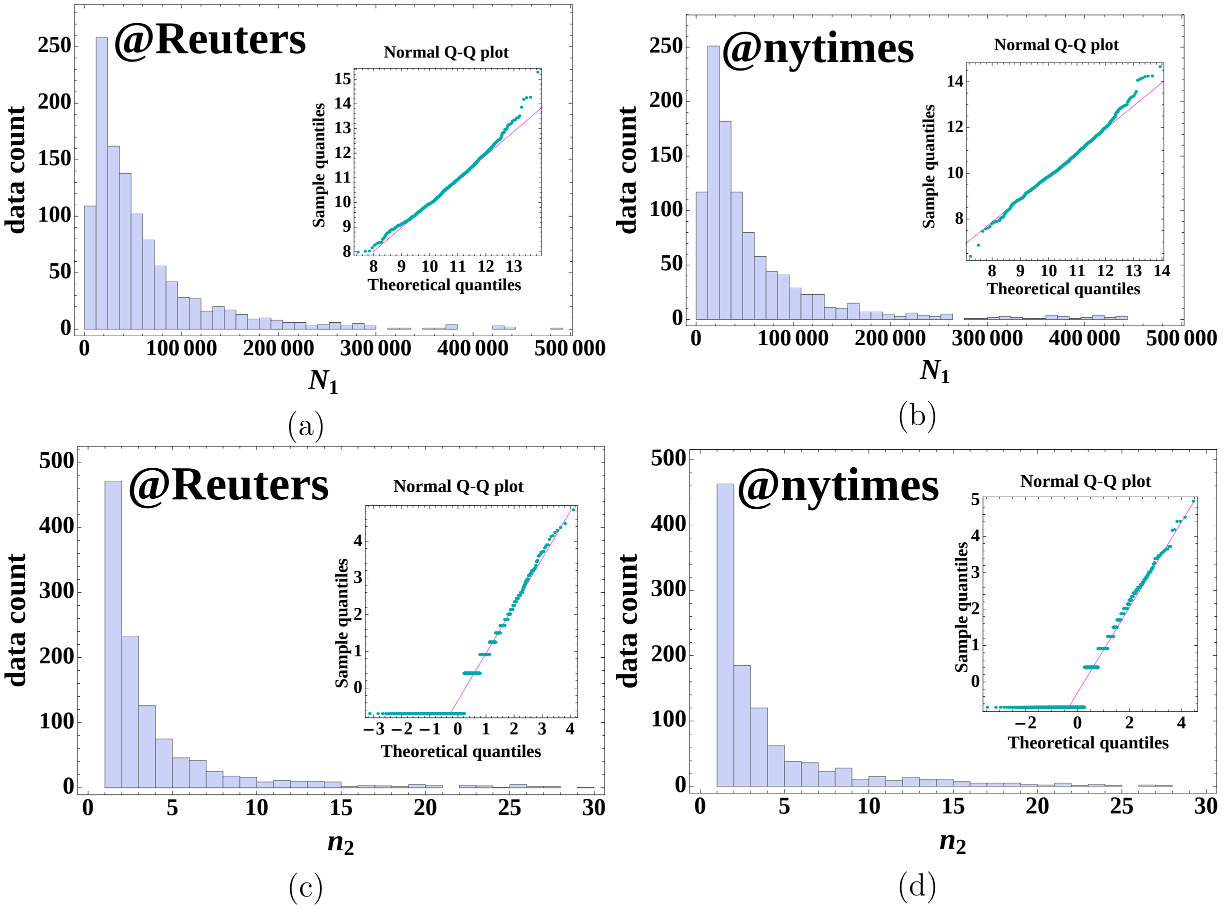
not independent of each other. The correlation coefficients of n_2 and N_1, $\rho(n_2, N_1)$, have large positive values (see Table 5.1); the correlation coefficient which is calculated by

$$\rho(n_2, N_1) = \frac{\langle n_2 N_1 \rangle - \langle n_2 \rangle \langle N_1 \rangle}{\sqrt{\langle n_2^2 \rangle - \langle n_2 \rangle^2}\sqrt{\langle N_1^2 \rangle - \langle N_1 \rangle^2}} \tag{3.18}$$

varies from -1 to 1. The fact that n_2 and N_1 are correlated supports our fundamental assumption that the diffusion is actually occurring along the network. If they were not correlated, it would imply that the retweeters might have been triggered by something other than the retweeters of the previous generation such as external sources of information.

Our result that the retweet rate β_2 obeys a lognormal distribution is plausible because independent lognormal distributions have the reproductive property; *i.e.* for the two stochastic variables β_2 and N_1, which obey lognormal distributions, we have

$$p(\ln n_2) = p(\ln \beta_2) * p(\ln N_1) = \mathcal{N}(\mu_1, \sigma_1^2) * \mathcal{N}(\mu_2, \sigma_2^2) = \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2), \tag{3.19}$$



(a)

(b)

(c)

(d)

Figure 3.6: The histograms of the retweet rates and the normal Q-Q plots of the logarithms of the number of followers and the retweet count in the second generation. (a) and (c) are of N_1 and n_2 for @Reuters. (b) and (d) are of N_1 and n_2 for @nytimes.

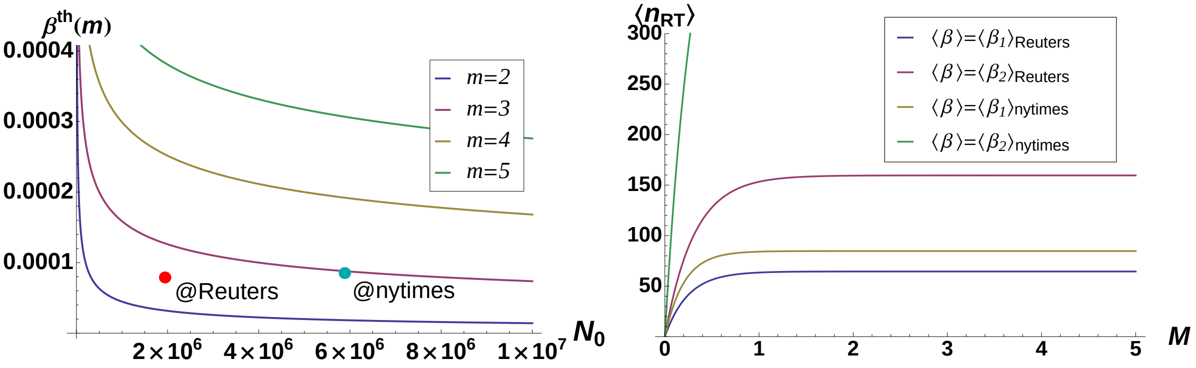
where $*$ stands for the convolution. We here assumed that β_2 and N_1 are independent of each other.

3.6 Estimation of the diffusion range (uncorrelated retweet rate)

In the following sections, we will analyze what can be estimated and expected from the model which we introduced above [76]. For simplicity, we will assume that the distribution of the retweet rate β_g is the same for each generation. From the model, we can estimate how much of the retweet rate $\beta^{\text{th}}(m)$ is required to reach the mth generation on average and the average of the total number of retweets, $\langle n_{\text{RT}} \rangle$, for given parameters. In this section, we restrict ourselves to the case where the retweet rates β_g are independent of each other and their averages have a common value $\langle \beta \rangle$.

	@Reuters	@nytimes
Number of seed tweets	1352	1140
Period	from Jun. 26, 2012 to Aug. 9, 2012	from Jun. 19, 2012 to Aug. 18, 2012
N_0	1 940 477	5 882 680
$\langle n_{\mathrm{RT}} \rangle$	74.0	97.3
$(\langle \beta_1 \rangle,\, \langle \beta_2 \rangle)$	$(3.27 \times 10^{-5},\, 7.90 \times 10^{-5})$	$(1.43 \times 10^{-5},\, 8.52 \times 10^{-5})$
$(\langle N_1 \rangle,\, \langle n_2 \rangle)$	$(75\,276,\, 4.46)$	$(76\,533,\, 4.54)$
$\rho(n_2,\, N_1)$	0.468	0.481
$\langle c_1 / \sum_{f=1}^{n_1} k_f \rangle$	0.0471	0.0470
Fitting parameters	$\mu_1 = -10.51$	$\mu_1 = -11.51$
for $\beta_1 = \mathrm{e}^{\omega_1} + \delta_1$	$\sigma_1^2 = 0.6$	$\sigma_1^2 = 0.77$
$p(\omega_1) = \mathcal{N}(\mu_1, \sigma_1^2)$	$\delta_1 = 0$	$\delta_1 = 1.0 \times 10^{-6}$
Fitting parameters	$\mu_2 = -9.65$	$\mu_2 = -9.51$
for $\beta_2 = \mathrm{e}^{\omega_2} + \delta_2$	$\sigma_2^2 = 0.68$	$\sigma_2^2 = 0.69$
$p(\omega_2) = \mathcal{N}(\mu_2, \sigma_2^2)$	$\delta_2 = -1.0 \times 10^{-5}$	$\delta_2 = -1.0 \times 10^{-5}$
Fitting parameters	$\mu = 10.64$	$\mu = 10.58$
for $N_1 = \mathrm{e}^{\omega} + \delta$	$\sigma^2 = 0.99$	$\sigma^2 = 1.04$
$p(\omega) = \mathcal{N}(\mu, \sigma^2)$	$\delta = 0$	$\delta = 2000$
Fitting parameters	$\mu = 0.48$	$\mu = 0.49$
for $n_2 = \mathrm{e}^{\omega} + \delta$	$\sigma^2 = 1.12$	$\sigma^2 = 1.23$
$p(\omega) = \mathcal{N}(\mu, \sigma^2)$	$\delta = 0.5$	$\delta = 0.5$

Table 3.1: Data of tweet diffusion from @Reuters and @nytimes. The angular brackets $\langle \cdots \rangle$ stands for the sample average.



(a) (b)

Figure 3.7: (a) Threshold where the diffusion reaches the mth generation on average. We set $\overline{k} = 500$. The points are for @Reuters and @nytimes in the case where we assumed $\langle\beta\rangle = \langle\beta_2\rangle$. (b) The average of the total number of retweets $\langle n_{\mathrm{RT}}\rangle$ as a function of the diffusion range M for various values of the average retweet rate $\langle\beta\rangle$. We set $\overline{k} = 500$ and plotted the cases where $\langle\beta\rangle$ equals $\langle\beta_g\rangle$ of @Reuters and @nytimes. We set the values of them for N_0, respectively.

According to Eq. (3.5), the average of the number of retweets in the mth generation reads $\langle n_m\rangle = N_0\overline{k}^{m-1}\langle\beta\rangle^m$. Then we have the threshold for the retweet rate where the diffusion reaches the mth generation on average, *i.e.* $\langle n_m\rangle \geq 1$:

$$\beta^{\mathrm{th}}(m) = \left(N_0\overline{k}^{m-1}\right)^{-\frac{1}{m}} = N_0^{-\frac{1}{m}}\overline{k}^{\frac{1}{m}-1}. \tag{3.20}$$

The behavior of Eq. (3.20) is exemplified in Fig. 3.7(a); in the case of the seed accounts which we investigated, the tweets diffuse up to the second or the third generation (see $\langle\beta_1\rangle$ and $\langle\beta_2\rangle$ in Table 5.1). While we employed the mean value of n_m in the definition of the threshold, it is also plausible to consider the median of n_m instead.

For a given range M of the diffusion, it is straightforward to calculate the average of the total number of retweets,

$$\langle n_{\mathrm{RT}}\rangle = \sum_{g=1}^{M}\langle n_g\rangle = N_0\langle\beta\rangle\sum_{g=0}^{M-1}\left(\langle\beta\rangle\overline{k}\right)^g = N_0\langle\beta\rangle\frac{1-\left(\langle\beta\rangle\overline{k}\right)^M}{1-\langle\beta\rangle\overline{k}}. \tag{3.21}$$

The behavior of Eq. (3.21) is exemplified in Fig. 3.7(b); it shows that $\langle n_{\mathrm{RT}}\rangle$ is not very sensitive to the diffusion range M in the case where $\langle\beta\rangle\overline{k}$ is small.

3.7 Viral diffusion

In this section, we focus on the situation where the diffusion goes viral, *i.e.*, the information which spreads to users who are extraordinarily far from a seed user [77]. The situation that we imagine for a viral diffusion is the diffusions of information of general interest, *e.g.*, such as postings with funny jokes, poetic writings, important news which are not broadcasted on other mass media, *etc.* The higher the retweet rate is, the wider the range of the diffusion is, which also results in a large number of retweets. A naive description of a tweet which enjoys many retweets would be the retweeting by a single user with a large number of the followers. Although it might be an important factor, even the accounts with millions of followers do not receive thousands of retweets for their daily tweets. Therefore, such a naive description does not explain the whole mechanism of the viral diffusion. The cooperation by many users is presumably crucial to the spread of the tweet.

Let us define a viral diffusion more precisely. As before, we assume a tree structure with a homogeneous degree distribution for the underlying network. We also assume an infinite path length from a seed user. Mathematically speaking, we define a viral diffusion as the diffusion which never stops on such a network; there exists a transition point for the retweet rate at which the diffusion goes viral. Our goal here is not to reproduce the statistical behavior of the data precisely, but to explore mathematical properties of the semi-microscopic diffusion law; even though diffusions always die out in reality because of the loop structure and the finite path length, as well as the decay of the retweet rate due to the temporal effect and the distance from the seed user, the analysis of such a transition point on the present toy model seems a plausible guideline for a viral diffusion.

Whenever the diffusion goes viral, however, we can easily imagine that the effect of correlation and fluctuation plays an important role. As we have seen in Sec. 3.5, the distribution of the retweet rate obeys a lognormal distribution which has a fat tail and it is interesting to see how it affects the dynamics. Although we consider a weak correlation of a certain form, we will show that it can largely enhance the chance of the viral diffusion indeed. We will first discuss the transition point of the viral diffusion in the case of independent retweet rates, and then we will show that the transition point is shifted owing to the correlation between the retweet rates.

In the following, we assume that every retweet rate β_g obeys a common lognormal distribution for simplicity, although its average and variance depend on the character of the seed node at $g = 1$ in reality. Then we set

$$p(J_g) = \frac{1}{J_g\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(\ln J_g - \mu)^2\right], \tag{3.22}$$

where $J_g = \beta_g \overline{k}$ as before and express J_g as

$$J_g = \mathrm{e}^{\mu+\xi_g}, \tag{3.23}$$

where μ and σ^2 are constant and ξ is a stochastic variable which obeys a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

3.7.1 The case of independent diffusion rate

In the following, we will consider the average number of the informed nodes N_{tot}, normalized by N_0. In the case where the stochastic variables J_g are independent of each other and all their averages are the same, *i.e.* $\langle J_g\rangle = \langle J\rangle$, we have

$$\begin{aligned}\frac{\langle N_{\mathrm{tot}}\rangle}{N_0} &= 1 + \langle J\rangle + \langle J\rangle^2 + \langle J\rangle^3 + \cdots \\ &= \frac{1}{1-\langle J\rangle}\end{aligned} \tag{3.24}$$

for $\langle J\rangle < 1$. In the case of the lognormal distribution (3.22), we have $\langle J\rangle = \exp\left(\mu + \sigma^2/2\right)$.

Since $J_g = \beta_g \overline{k}$, and hence $\langle J\rangle = \langle\beta\rangle\overline{k}$, Eq. (3.24) gives the transition point

$$\beta_{\mathrm{ex}} = \overline{k}^{-1} \tag{3.25}$$

for the viral diffusion. In the case of the Twitter network, $\overline{k} \sim \mathcal{O}(10^2)$ and hence the transition point is $\beta_{\mathrm{ex}} \sim \mathcal{O}(10^{-2})$. On the other hand, in the case of some major news accounts such as The New York Times (@nytimes) and Reuters Top News (@Reuters), $\langle\beta\rangle \sim \mathcal{O}(10^{-5})$, which is much lower than the transition point. Because of the restriction of Twitter API [6], we cannot measure the value of the retweet rate β_g of the viral diffusion explicitly. Although the possibility of reaching the transition point $\beta_{\mathrm{ex}} = \overline{k}^{-1}$ depends on the average and the variance of the retweet rate, the threshold appears to be too high to reach in reality if we assume that J_g are independent of each other.

3.7.2 The case of correlated diffusion rates

In order to make a better estimate of the transition point, let us now consider the quantity $\langle N_{\mathrm{tot}}\rangle/N_0$ in the case where the stochastic variables J_g are not independent of each other. Instead of setting ξ_g in Eq. (3.23) as an independent Gaussian variable, we now set

$$p(\{\xi_g\}) = \frac{1}{Z}\exp\left[-\frac{1}{2}\sum_{ij}\xi_i\Sigma^{-1}_{ij}\xi_j\right], \qquad Z = \sqrt{\frac{(2\pi)^N}{\det\Sigma^{-1}}}, \tag{3.26}$$

where Z is the normalization factor and Σ^{-1} is the inverse matrix of the covariance matrix $\Sigma_{ij} = \langle\xi_i\xi_j\rangle$. The matrix Σ^{-1} is an infinite-dimensional matrix; we first treat it as an $N \times N$ matrix and take the limit $N \to \infty$ in the end. We assume the following matrix for Σ^{-1}:

$$\Sigma^{-1} = \begin{bmatrix}\sigma^{-2} & -\eta & 0 & \cdots \\ -\eta & \sigma^{-2} & -\eta & \cdots \\ 0 & -\eta & \sigma^{-2} & \cdots \\ \vdots & \vdots & \vdots & \ddots\end{bmatrix}. \tag{3.27}$$

The statistical average of the number of the informed nodes is now given by

$$\frac{\langle N_{\rm tot}\rangle}{N_0} = 1 + \sum_{m=1}^{\infty}\langle\prod_{g=1}^{m} J_g\rangle, \tag{3.28}$$

where the average $\langle\cdots\rangle$ is now taken with respect to the correlated distribution (3.26). In order to calculate the average, we diagonalize the matrix Σ^{-1} with a unitary matrix U to obtain

$$P(\vec{x}) = \frac{1}{Z}\exp\left[-\frac{1}{2}\sum_{i=1}^{N}(\sigma^{-2} - \eta\lambda_i)x_i^2\right], \tag{3.29}$$

where

$$\vec{x} = U\vec{\xi}, \qquad U_{mn} = \frac{1}{L}\sin(mk_n), \tag{3.30}$$

$$\lambda_\alpha = 2\cos k_\alpha, \qquad k_\alpha = \frac{\pi\alpha}{(N+1)}, \tag{3.31}$$

$$L^2 = \frac{1}{2}(N+1). \tag{3.32}$$

See Appendix A for detail. After this diagonalization, we have

$$\begin{aligned}\langle\prod_{g=1}^{m} J_g\rangle &= \mathrm{e}^{m\mu}\int d\vec{\xi}\, P(\vec{\xi})\exp\left(\sum_{g=1}^{m}\xi_g\right)\\ &= \mathrm{e}^{m\mu}\int\frac{d^N x}{Z}\exp\left[-\frac{1}{2}\sum_{i=1}^{N}a_i x_i^2\right]\exp\left(\sum_{j=1}^{N}b_j x_j\right)\\ &= \mathrm{e}^{m\mu}\exp\left(\sum_{j=1}^{N}\frac{b_j^2}{2a_j}\right),\end{aligned} \tag{3.33}$$

where

$$\begin{aligned}a_i &= \sigma^{-2} - \eta\lambda_i = \sigma^{-2} - 2\eta\cos k_i,\\ b_j &= \frac{1}{L}\sum_{g=1}^{m}\sin(gk_j),\end{aligned} \tag{3.34}$$

and we used the relation

$$\sum_{g=1}^{m}\xi_g = \frac{1}{L}\sum_{g=1}^{m}\sum_{j=1}^{N}\sin(gk_j)x_j = \sum_{j=1}^{N}b_j x_j. \tag{3.35}$$

Substituting these values into Eq. (3.33), we obtain

$$\begin{aligned}\langle\prod_{g=1}^{m} J_g\rangle &= \mathrm{e}^{m\mu}\exp\left[\sum_{j=1}^{N}\sum_{g,g'=1}^{m}\frac{\sin gk_j\sin g'k_j}{a_j(N+1)}\right]\\ &= \mathrm{e}^{m\mu}\exp\Bigg[\frac{1}{2(N+1)}\sum_{j=1}^{N}\sum_{g,g'=1}^{m}a_j^{-1}\\ &\qquad\Big(\cos k_j(g-g')-\cos k_j(g+g')\Big)\Bigg].\end{aligned} \tag{3.36}$$

Let us now consider the case where $\epsilon \equiv \eta/\sigma^{-2} \ll 1$ and analyze the expansion of a_j^{-1} with respect to ϵ:

$$a_j^{-1} = \sigma^2\left(1 + 2\epsilon\cos k_j + o(\epsilon)\right). \tag{3.37}$$

From the zeroth-order expansion, we simply obtain $\langle\prod_{g=1}^{m} J_g\rangle = \langle J\rangle^m$, which reduces to the non-correlated case (3.24). Including the first-order correction of ϵ, we have

$$\langle\prod_{g=1}^{m} J_g\rangle = \langle J\rangle^m\exp\left[\frac{2\epsilon\sigma^2}{2(N+1)}\sum_{j=1}^{N}\sum_{g,g'=1}^{m}\cos k_j\left(\cos k_j(g-g')-\cos k_j(g+g')\right)\right]. \tag{3.38}$$

After some algebra, we obtain

$$\langle\prod_{g=1}^{m} J_g\rangle = \langle J\rangle^m\mathrm{e}^{\epsilon\sigma^2(m-1)}. \tag{3.39}$$

Hence, the total number of the informed nodes normalized by N_0 reads

$$\frac{\langle N_{\text{tot}}\rangle}{N_0} = 1 + \sum_{m=1}^{\infty}\langle\prod_{g=1}^{m} J_g\rangle = 1 + \frac{\langle J\rangle}{1-\langle J\rangle\mathrm{e}^{\epsilon\sigma^2}} \tag{3.40}$$

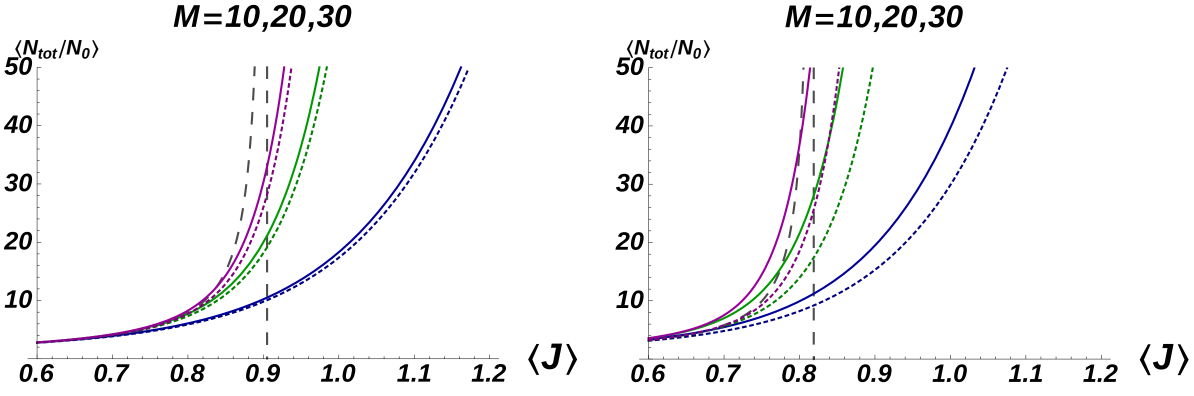
for $\langle J\rangle\mathrm{e}^{\epsilon\sigma^2} < 1$. Since $\langle J\rangle = \langle\beta\rangle\overline{k}$ again, the transition point for the viral diffusion β_{ex} now reads

$$\beta_{\text{ex}} = \overline{k}^{-1}\mathrm{e}^{-\epsilon\sigma^2} \tag{3.41}$$

instead of Eq. (3.25). The correlation between the retweet rates thus shifts the transition point to a lower retweet rate.

We expect that the perturbative estimate (3.39) of the transition point gives an upper bound of the true transition point. In Fig. 3.8, we can confirm it by comparing (i) (solid lines) numerical estimates of Eq. (3.36) substituted into

$$\frac{\langle N_{\text{tot}}\rangle_M}{N_0} = 1 + \sum_{m=1}^{M}\langle\prod_{g=1}^{m} J_g\rangle, \tag{3.42}$$



(a) (b)

Figure 3.8: Numerically calculated results of $\langle N_{\text{tot}}\rangle/N_0$ in Eq. (3.42), where the sum is taken up to $M = 10$ (blue), 20 (green), 30 (red). The dotted lines indicate the approximated results with the perturbative estimate (3.39) and the solid lines indicate the results with numerical estimates of Eq. (3.36). The parameters are set to $\sigma^2 = 2$, $\epsilon = 0.05$ for (a) and $\sigma^2 = 2$, $\epsilon = 0.1$ for (b). We set $N = 30$ for the calculation of Eq. (3.36); the result is the same as long as $N \geq M$. The broken line shows the behavior of Eq. (3.40), which is the case of $M = \infty$ with the perturbative estimate (3.39).

and (ii) (dotted lines) perturbative estimates Eq. (3.39) substituted into Eq. (3.42). The former is always greater than the latter as far as we checked. We hence expect that it is also true in the limit $M \to \infty$. Then the true curve of $\langle N_{\text{tot}}\rangle/N_0$ in the limit $M \to \infty$ should be greater than its perturbative estimate (3.40) (dashed lines in Fig. 3.8). It implies that the true transition point of the viral diffusion is equal to or lower than the perturbative estimate (3.41).

Let us next write down the transition point in terms of the correlation coefficient of the retweet rates instead of the off-diagonal element $\epsilon = \eta/\sigma^{-2}$ of the matrix Σ^{-1}. The matrix Σ^{-1} which contains the off-diagonal element ϵ is the inverse matrix of the covariance matrix Σ of ξ_g, which is related to that of J_g by Eq. (3.23). Expressing the inverse of the covariance matrix as $\Sigma^{-1} = \sigma^{-2}F_N$, the covariance matrix Σ reads

$$\Sigma_{ik} = \Sigma_{ki} = \frac{\sigma^2}{\det F_N} \det F_{i-1} \det F_{N-k}\, \epsilon^{k-i} \tag{3.43}$$

for $i \leq k$, where the subscript of the matrix F_N denotes the number of dimensions and we defined $\det F_0 = 1$. The determinant of F_g has the following recursion relation

$$\det F_g = \det F_{g-1} - \epsilon^2 \det F_{g-2}. \tag{3.44}$$

In the limit $N \to \infty$, it reduces to

$$\frac{1}{r} = 1 - \epsilon^2 r, \tag{3.45}$$

where

$$r = \lim_{N \to \infty} \frac{\det F_{N-1}}{\det F_N}. \tag{3.46}$$

Considering the fact that r needs to satisfy $r^n < \infty (n \to \infty)$, we have

$$r = \frac{1 - \sqrt{1 - 4\epsilon^2}}{2\epsilon^2}. \tag{3.47}$$

Hereafter, we will work in the limit $N \to \infty$. Noting that $\det F_{g-1}$ and r are both $1 + \mathcal{O}(\epsilon^2)$, we have the matrix elements of Eq. (3.43) as

$$\begin{aligned} \langle \xi_g \xi_{g+1} \rangle &= \epsilon \sigma^2 \det F_{g-1} \frac{\det F_{N-g-1}}{\det F_N} = \epsilon \sigma^2 \det F_{g-1} r^{g+1} \\ &= \epsilon \sigma^2 + \mathcal{O}(\epsilon^2). \end{aligned} \tag{3.48}$$

Hence, up to the accuracy of $\mathcal{O}(\epsilon)$, the off-diagonal element ϵ is written in terms of the covariance of $\langle \xi_g \xi_{g+1} \rangle$ as

$$\epsilon = \sigma^{-2} \langle \xi_g \xi_{g+1} \rangle. \tag{3.49}$$

The covariance of ξ_g is written in terms of the covariance of J_g according to Eq. (3.23) using Wick's theorem:

$$\begin{aligned} &\langle J_i J_j \rangle - \langle J_i \rangle \langle J_j \rangle \\ &= \mathrm{e}^{\mu_i + \mu_j} \left(\langle \mathrm{e}^{\xi_i + \xi_j} \rangle - \langle \mathrm{e}^{\xi_i} \rangle \langle \mathrm{e}^{\xi_j} \rangle \right) \\ &= \mathrm{e}^{\mu_i + \mu_j} \left(\sum_{w=0}^{\infty} \frac{1}{w!} \langle \xi_i + \xi_j \rangle^w - \sum_{u,v=0}^{\infty} \frac{1}{u!v!} \langle \xi_i \rangle^u \langle \xi_j \rangle^v \right) \\ &= \mathrm{e}^{\mu_i + \mu_j} \left(\sum_{l=1}^{\infty} \sum_{m,n=0}^{\infty} \frac{1}{(l+2m)!(l+2n)!} l! \frac{(l+2m)!}{2^m m! l!} \frac{(l+2n)!}{2^n n! l!} \langle \xi_i \xi_j \rangle^l \langle \xi_i^2 \rangle^m \langle \xi_i^2 \rangle^n \right) \\ &= \mathrm{e}^{\mu_i + \mu_j} \mathrm{e}^{\frac{1}{2}(\sigma_i^2 + \sigma_j^2)} (\mathrm{e}^{\langle \xi_i \xi_j \rangle} - 1) \\ &= \langle J_i \rangle \langle J_j \rangle (\mathrm{e}^{\langle \xi_i \xi_j \rangle} - 1). \end{aligned} \tag{3.50}$$

Therefore, Eq. (3.49) now reads

$$\epsilon = \sigma^{-2} \ln \frac{\langle J_g J_{g+1} \rangle}{\langle J_g \rangle \langle J_{g+1} \rangle}. \tag{3.51}$$

Substituting Eq. (3.51) into Eq. (3.41), we have the shift of the threshold of the transition point β_{ex} in the form

$$\begin{aligned} \beta_{\mathrm{ex}} &= \overline{k}^{-1} \left(\frac{\langle \beta_g \rangle \langle \beta_{g+1} \rangle}{\langle \beta_g \beta_{g+1} \rangle} \right) \\ &= \overline{k}^{-1} \left[1 + \rho(\beta_g, \beta_{g+1}) \frac{V(\beta_g)}{\langle \beta_g \rangle^2} \right]^{-1}, \end{aligned} \tag{3.52}$$

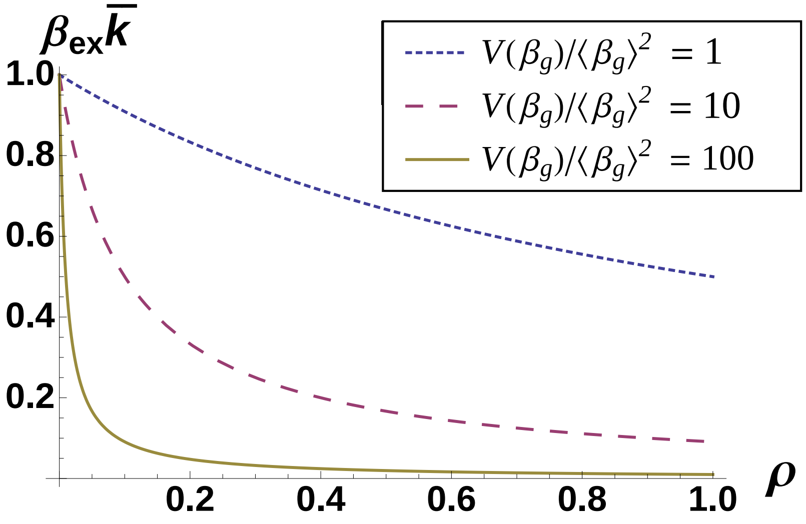


Figure 3.9: The dependence of the transition point of the viral diffusion, Eq. (3.52), as a function of the correlation coefficient $\rho(\beta_g, \beta_{g+1})$. The result of Eq. (3.25) corresponds to the case where $\rho = 0$.

where $\rho(\beta_g, \beta_{g+1})$ is the correlation coefficient which varies from -1 to 1 and $V(\beta_g)$ is the variance of β_g.

We exemplify the behavior of Eq. (3.52) in Fig. 3.9. If $V(\beta_g)/\langle\beta_g\rangle^2 \sim \mathcal{O}(1)$, the transition point would be lowered only up to a half of the case of the independent process, while it is lowered significantly in the case where $V(\beta_g)/\langle\beta_g\rangle^2 \gtrsim \mathcal{O}(10^2)$; even when $\rho(\beta_g, \beta_{g+1}) = 0.2$, the diffusion is about twenty times more likely to go viral than the uncorrelated case.

3.7.3 Remarks

When we discuss the viral diffusion, the average of the retweet rate is not the only significant factor, but its fluctuation and the correlation may also play important roles. Equation (3.52) means that the transition point where the diffusion goes viral is shifted owing to the correlation $\rho(\beta_g, \beta_{g+1})$ of the retweet rates between the generations. The larger the variance $V(\beta_g)$ of the retweet rate is compared to the square of its average $\langle\beta_g\rangle$, the easier it is to make the diffusion go viral. On the other hand, it is hopeless to expect the information diffusion with very narrow variance of the retweet rate to go viral, unless it is constantly very close to the transition point of the uncorrelated case, $\beta_{\text{ex}} = \overline{k}^{-1}$.

We defined the transition point of the viral diffusion as a theoretical guideline of the information diffusion on an online social network such that the information reaches the nodes which are extraordinarily far from the seed node. We showed how the correlation between the nodes enhance the chance of the viral diffusion. Although we used a perturbation expansion with respect to the off-diagonal matrix element ϵ in Eq. (3.37), its higher-order expansion is straightforward. Note that ϵ cannot be too large, in other words, $\rho(\beta_g, \beta_{g+1})$ cannot be close to one, in order to retain the positivity of the covariance matrix Σ, which also validates the perturbation expansion. We numerically showed that the true transition point may be even lower than the current result of the perturbative approach.

For Twitter, the transition point would be unrealistically far to reach without the correlation between the generations. The significant change of the transition point due to the correlation seems to be essential in understanding the reason why such postings sometimes diffuse extraordinary far from the seed user.

The transition point (3.52) may be still far to reach even after taking into account the correlation effect. The assumptions which we made on the underlying network such as the homogeneity of distribution and the infinite path length may cause the change of the estimation of the transition point. In order to analyze the diffusion more precisely, removing these assumptions is an interesting future problem. The heterogeneity would describe the effect of complex diffusion paths. Although the average path lengths are usually very short for many networks in real world [46], the path length of the diffusion can be much longer than the average path length of the underlying network, because the diffusions do not always occur along the shortest paths [21, 59, 97, 118].

Chapter 4

Methods of community detection in complex networks

Community structure is an important property of a complex network. One way to detect communities in a network is to consider a random walk on a network as a hypothetical stochastic process. The aim of this chapter is to review several methods of community detection. We first explain how communities are defined and go over a brief history of the community detection methods. Then, short tutorials of the spectral clustering (Secs. 4.4) and the method of the modularity (Sec. 4.5) follow, which are both very famous and extensively studied. The random walk interpretation of the spectral clustering and the method of the modularity are also explained, although they are originally formulated in terms of graph quantities.

4.1 How communities are defined

Intuitively speaking, Fig. 4.1 is perhaps the picture that everyone imagines from the word "community structure" in a network. We can regard each set of nodes enclosed by a dashed line as a community. Nodes are densely connected within a community compared to the nodes outside the community. Community structure is an important property of a network because the fact that nodes are densely connected often has a significant meaning; in a protein network, it means the nodes with a similar function [37] and in a friendship network, it implies a social group such as people in the same school, the people with a common hobby, and so on. It may seem that defining a community itself is a simple matter and the problem is how to detect it. There is, however, no unified definition of a community; indeed, there exist many [53]. While they are conceptually close, if not equivalent, to each other, each of them is defined in different plausible ways. The following are the ideas (not the precise definitions) of a community in some detection methods.

weak definition [123] A community is a subgraph whose sum of its internal degrees of the community is greater than the sum of their external degrees.

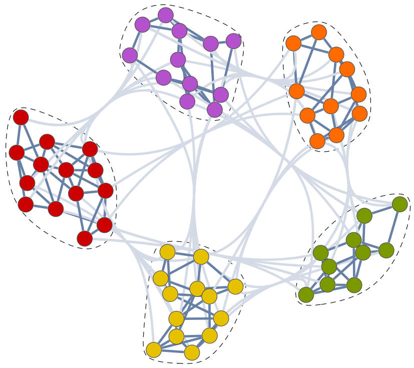


Figure 4.1: An example of communities in a network. Each community is enclosed by a dashed line.

modularity [114] A community is a subgraph which has more internal links than expected in a null model which does not have a community structure in principle.

map equation [131] Consider a random walker on the network. A community is a subgraph where the random walker seldom escapes from there.

clique percolation [117] A community is a subgraph which is close to a complete graph.

Girvan-Newman method [57] A community is a subgraph which is disconnected from the rest of the network by removing some significant links.

There are a lot more definitions than the ones which we listed above. While some of them are defined by the local structure of a subgraph, in many methods, communities are algorithmically defined. That is, communities are defined as the resulting products after a certain optimization process. Hence, a definition of a community is often embedded in a method. There exist many equivalent names for a community; it is called a module or a cluster as well. In this thesis, we call it a community in a conceptual argument and call it a module when we do a mathematical argument.

Which definition and method are the best is not determined yet. For some definitions, we cannot determine which is better because, depending on the problem, one definition may be more appropriate than the other definitions. Some methods, however, can be ranked according to benchmark tests. Some of them may have higher accuracy and/or numerical efficiency compared to others. The accuracy that we mean here is the ability to detect fuzzy communities, *i.e.* the communities with many links between other communities compared to the internal links. (It is called the detectability.) Moreover, as long as the methods share the intuitive definition of a community, perhaps they are not all independent, but are mathematically related closely to each other.

Other than the above conceptual issue, a community has yet more degrees of freedom in its definition. A community is conventionally defined as a set of nodes and the links which connect those nodes, but we do not consider the overlap of communities. It is, however, possible to define a community as a set of links and allow communities to overlap. It is quite natural, for example in the case of a social network, because everyone usually belongs to many different social groups and it is appropriate to choose one community. Considering the hierarchical structure is also essential. A community often consists of some communities of a smaller scale and those small communities may also consist of communities of an even smaller scale. Therefore, it is not natural to choose one scale for the communities, but we should consider the whole hierarchy of the community structure.

A network can be weighted and directed. For a network of companies in which nodes are connected according to the business connection between them, their links have different weights depending on the amount of the transaction. In the case of a collaboration network among the researchers in a certain field, some researchers collaborate for many times and thus have multiple links between them; those multiple links can be regarded as a weighted single link. The direction is also an important property and there are numbers of examples. In a citation network of academic papers, the links indicate the citations among articles. Such links are always in one direction because one can cite a published paper while the paper in the past can never cite papers which will be published in future. While the extension of a community detection method for unweighted networks to weighted networks is usually an easy task, the extension to directed networks often cause a conceptual problem since we do not have a common view of the role of the direction in communities.

4.2 A brief history of the community detection

A number of community detection methods have been invented, developed, and applied to a variety of networks [53]. The history of the community detection goes back to the work by Rice [127] who studied the clusters of people in small political bodies based on the voting similarity and to the method considered by Weiss and Jacobson [151] who studied work groups within a government agency. The method of Weiss and Jacobson was to remove the nodes which belong to different groups; it can be regarded as the origin of a modern divisive method [57].

Methods of spectral clustering [51,101,116,137] are traditional methods of community detection and had been studied mainly by the researchers who are not in the disciplines of physics. Those methods make use of the eigenvectors of the unnormalized or normalized graph Laplacians which we will introduce later. Finding the exact solution in a spectral clustering method is an optimization problem of a quality function in a discrete space. It is, however, computationally infeasible in many cases. Therefore, it is usual to replace the discrete space with a continuous one and solve a relaxed problem, although the accuracy of the resulting partition is often not very high [101]. We have to specify the number of modules as an input for the spectral clustering. In some cases, we can estimate a plausible input from a gap in eigenvalues of the graph Laplacian.

The divisive method which was proposed by Girvan and Newman [57] in 2002 is one of the most famous methods and is historically important; many physicists started to work on the problem of community detection stimulated by their work. In their method, they consider a quantity called *edge betweenness* for each link. When we consider the shortest paths of every node pair, the edge betweenness of a link is the number of the shortest paths that contain it. Higher the edge betweenness of a link is, more likely the link connects two modules; therefore, the links with high edge betweenness tend to disconnect the modules. We remove the links with high betweenness successively, until a set of nodes, *i.e.* a module, is isolated. A problem of the method of Girvan and Newman is that it is not numerically efficient because one needs to calculate the shortest paths of every node pair. Another problem is that the method does not tell us where to stop the algorithm. If we do not stop the algorithm, we divide the network until each node becomes a single module, which is expected to take a very long time in the case of a large network.

In order to determine where to stop the algorithm with the edge betweenness [57], Newman and Girvan later introduced a quality function called *modularity* [114] in 2004. Ever since, the method of maximizing the modularity itself has become the most popular method in this field. As we briefly mentioned in Sec. 4.1, the modularity quantifies how nodes inside a module are densely connected than they are expected in a random graph without a community structure; we will explain the modularity in detail in Sec. 4.5. There have been many extensions of the method of the modularity. One of the reasons why the modularity collected huge attention is because the method automatically determines the number of modules, while many of classical methods required the number of the modules to be an input. Numerical efficiency is another reason of the popularity. Although the exact optimization of the modularity is an **NP**-complete problem because it requires the trial of all possible partitions, the greedy optimization algorithm often shows a decent performance at a reasonable computational cost. The algorithm proposed by Blondel *et al.* [30], or the *Louvain method*, is known as a numerically efficient algorithm for the modularity optimization.

Although many detection methods treat undirected networks for simplicity, needless to say, it is important to develop methods for directed networks. While there is an extension of the modularity to the directed network introduced by some authors [81, 89], the method which naturally takes account of the flow was invented by Rosvall and Bergstrom [131] in 2008. It is based on the information theory and the correct partition is obtained by minimizing the quality function which is called *map equation*. This is the method which we discuss mainly in the second half of this thesis and we will explain it in great detail later in the next chapter. The method of the map equation is not only useful for the directed networks, but also has good features that the modularity has; the method automatically determines the number of the modules during the optimization process and the optimization is done at a reasonable computational cost. Its extraordinary performance is proved in recent benchmark tests [15, 85]. Before the method of the map equation, Rosvall and Bergstrom invented another method which is also based on the information theory [130], although it did not show a decent performance in the benchmark [85] compared to the map equation.

There are many other methods such as the one using synchronization [17], a quantity called *communicability* [49], *Surprise* [13], order statistics local optimization method (OSLOM) [88], and the one in a framework of statistical inference, which is called *stochastic block model* [75, 115]. The quality function called the *conductance* is also popular in computer science [53], which is close to the quality functions used in the spectral custering. The method of *clique percolation* [117], which was introduced by Palla *et al.* is also famous because it enables us to detect overlapping communities. The definition of a community in the clique percolation is a little bit different from other methods; instead of defining a community as a result of partitioning, it defines a community as a subgraph which is close to a complete graph, or a clique. A community is detected locally and is not affected by the existence of other communities, and thus we are able to make it overlapped with others. There are some other methods which allow communities to be overlapped, such as the one with the communicability [49] and the ones which define communities with links [11, 50, 80] as we mentioned above. Reference [80] shows the link-community version of the map equation.

Invention of new methods and algorithms are not the only goal. There are also some researches which discuss the robustness [74], significance [87, 88], and the properties [20, 38, 63, 93] of the communities. It was shown that the community-size distribution of a real network often obeys a power law [38, 63]. As we will show in the next chapter, such a distribution is observed in some networks by the method of the map equation as well, and thus it is expected to be a property independent of the detection method. The detectability, which measures the ability to detect a fuzzy module, is studied as well [121, 126]. There are analytical results shown recently, while it used to be discussed numerically in some benchmark tests.

4.3 Benchmarks and applications

One of the classical examples of the real-world networks that is used as a benchmark of the community detection is Zachary's karate club network in the United States [164] as shown in Fig. 4.2. Each node represents a member of the karate club and a pair of nodes are connected if the interaction exists between them outside of the club activity. There were 34 members in total including the instructor and the president of the club. There was a conflict between the instructor and the president in the club and the members were separated into the people who supported the instructor and the people who supported the president. The problem is whether a detection method is able to detect the communities correctly. The important feature of this network is that the answer, *i.e.* who supported whom, is known by interview. Lusseau's dolphins' network [18, 100, 130] is another example in which the correct communities are known. Other than these networks, the methods of community detection have been applied to networks of many disciplines. For biochemical netowrks, there have been researches for protein networks [37, 128, 139], a metabolic network [66], and a gene network [154]. In Ref. [154], the authors analyzed the abstracts of articles in the Medline database in order to build a network of genes; a pair of genes are connected if they are mentioned together in an article. As a result of a community

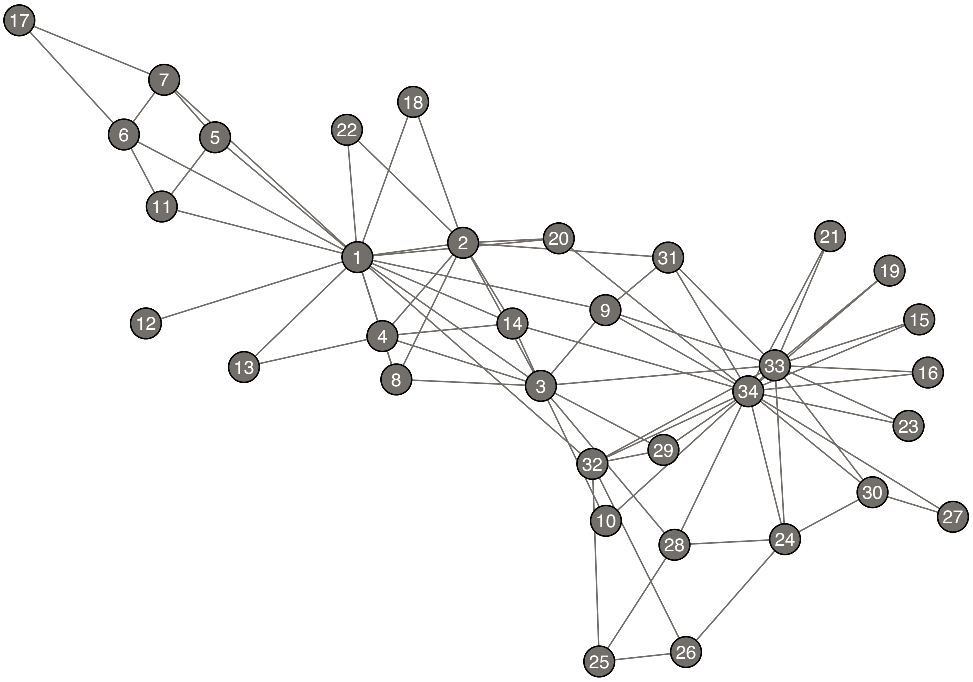


Figure 4.2: Zachary's karate club network. Node 1 represents the instructor and node 34 represents the president.

detection, one obtains sets of genes which are functionally related within each module. There are also works on the Web contents [125, 142] and politics [107, 122].

Other than the benchmarks of the real world networks, there are some benchmark tests with synthetic networks. One of the most basic synthetic benchmark networks is of Girvan and Newman [57]. The network consists of 128 nodes and is divided into equal size modules (i.e., 32 nodes in every module). A pair of nodes are connected with probability P_{in} if they belong to the same module and if not, they are connected with probability P_{out} ($P_{\text{in}} > P_{\text{out}}$), while keeping the average degree of a node to 16. Higher the probability P_{out} is, more difficult an algorithm to detect the correct modules.

The Girvan-Newman benchmark network is, however, not difficult enough to test the performances of the algorithms and the setting of the equal size modules is not realistic since many real networks consist of communities of many scales. For the benchmark test which is more difficult and more realistic, Lancichinetti, Fortunato and Radicchi proposed another type of synthetic networks, which is called LFR-benchmarks [85, 86]. In the LFR-benchmark network, both the degree distribution of the nodes and the module size distribution are chosen to follow power laws. Each node has a fraction $(1 - \mu)$ of its links which are connected to the nodes in the same module and the rest of its links are connected to the nodes outside of the module at random; the parameter μ is called mixing parameter. Although the LFR-benchmarks have skew module-size distributions, most of the modules have similar size.

A benchmark network based on Relaxed Caveman structure, which is constructed by rewiring the links of the isolated modules of the complete graphs, is also considered [14, 15, 149]. By setting the distribution of the module sizes highly skewed, one can construct a benchmark network which is even more difficult than the LFR-benchmark network.

In many benchmark tests, the network loses its initial community structure as one raises the mixing parameter (or the rate of rewiring) and one discusses whether the initial community structure is detectable or not. Aldecoa *et al.* call them *open* benchmarks [14, 15]. The open benchmarks have the following weakness; when an algorithm fails to detect the designed structure at a certain value of the mixing parameter, one cannot distinguish whether it is due to the poor performance of the algorithm or because the community structure of the network does not correspond to that of the initial network anymore. In order to overcome this limitation, Aldecoa *et al.* considered a type of benchmark called *closed* benchmarks [14, 15], in which the initial and the final networks have designed community structures and the initial network is converted to the final network by rewiring. One observes the evolution of the result of the detected structure in the conversion process. Note that any networks for the initial and the final networks can be chosen, including the ones used in the above open benchmark tests.

4.4 Spectral clustering

In this section, we basically follow Ref. [101], although we use a different convention for the normalized graph Laplacian and a slightly different notation. The methods of spectral clustering make use of the eigenvectors of unnormalized and normalized graph Laplacians. As we will explain in Sec. 5.1, they appear as the matrix which is responsible for the time evolution of a random walker. Unnormalized and normalized graph Laplacians, L, L_{rw}, and L_{sym} are defined as

$$\begin{cases} L = D - A & & \text{unnormalized graph Laplacian,} \\ L_{\mathrm{rw}} = LD^{-1} & L_{\mathrm{sym}} = D^{-1/2} L D^{-1/2} & \text{normalized graph Laplacian,} \end{cases} \tag{4.1}$$

where A is the adjacency matrix and $D = \mathrm{diag}(k_\alpha)$ (k_α is the degree of the node α). Both L_{rw} and L_{sym} are called the normalized graph Laplacians.

4.4.1 RatioCut, Ncut, and conductance

In the following, we denote a network, or a graph, as V. Before we analyze the properties of the graph Laplacians, we first show the goal of the spectral clustering. We define a quantity called *cut size* (or simply, *cut*); it is the number of links which connect the modules. This quantity plays an important role in the analysis of the next chapter as well. Mathematically, it can be expressed as follows. Let the number of links which connect the module A and the module B be denoted by

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}, \tag{4.2}$$

where $w_{ij} = 1$ if there exists a link between the node i and the node j. The cut size for the partition $\{A_1, A_2, \dots, A_m\}$ is then

$$\mathrm{cut}(A_1, A_2, \dots, A_m) = \frac{1}{2} \sum_{i=1}^{m} W(A_i, \overline{A}_i), \tag{4.3}$$

where $\overline{A}_i$ is the complement of A_i in V. The factor 1/2 is to take account of the double counting of the links between the modules. At a glance, it may seem that the one which minimizes the cut size gives the correct partition. It is, however, not correct because we would obtain the network without any partitions, which gives the cut size equal to zero. Therefore, we need to balance the cut size with the size of the modules. The quality functions that we consider are *RatioCut* [64,116] and *Ncut* (or the *Normalized cuts*) [137]. Their definitions are as follows:

$$\mathrm{RatioCut}(A_1, A_2, \dots, A_m) = \sum_{i=1}^{m} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}, \tag{4.4}$$

$$\mathrm{Ncut}(A_1, A_2, \dots, A_m) = \sum_{i=1}^{m} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{\mathrm{vol}(A_i)}. \tag{4.5}$$

In this section, $|X|$ denotes the number of vertices and $\mathrm{vol}(X)$ the sum of degrees in a (sub)graph X. In the above definitions, the total number of vertices $|A_i|$ and the total number of links $\mathrm{vol}(A_i)$ in the module i are the factors which regulate the size of the modules. Hence, the problem is to minimize the RatioCut or the Ncut. Indeed, those quantities can be expressed in terms of the graph Laplacians.

The conductance $\Phi(S)$ of a subgraph S, which is often used in computer science, is quite similar to the RatioCut. It is defined as

$$\Phi(S) = \frac{2\,\mathrm{cut}(S, \overline{S})}{\min(|S|, |\overline{S}|)}. \tag{4.6}$$

Note that the conductance is a global quantity because it has $|\overline{S}|$ in the denominator, even though its argument is a subgraph.

4.4.2 Partitioning as an eigenvector problem

For simplicity, we consider partitioning a network V into two clusters, A and $\overline{A}$. The partitioning into more than two clusters is discussed later. The RatioCut can be expressed with the unnormalized graph Laplacian L as follows. We consider a $|V|$-dimensional vector

$|f\rangle$, whose element f_i takes one of two values depending on whether the node v_i belongs to A or $\overline{A}$ as follows:

$$f_i = \begin{cases} \sqrt{|\overline{A}|/|A|} & v_i \in A, \\ -\sqrt{|A|/|\overline{A}|} & v_i \in \overline{A}. \end{cases} \tag{4.7}$$

Let $\langle f|$ be the transpose of $|f\rangle$. The inner product $\langle f|L|f\rangle$ is then

$$\begin{aligned}
\langle f|L|f\rangle = \langle f|D|f\rangle - \langle f|A|f\rangle &= \sum_i k_i f_i^2 - \sum_{ij} f_i f_j w_{ij} \\
&= \frac{1}{2}\left(\sum_i k_i f_i^2 - 2\sum_{ij} f_i f_j w_{ij} + \sum_j k_j f_j^2\right) \\
&= \frac{1}{2}\sum_{ij} w_{ij}(f_i - f_j)^2 \qquad (4.8) \\
&= \frac{1}{2}\sum_{i\in A, j\in\overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \frac{1}{2}\sum_{i\in\overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\
&= \mathrm{cut}(A, \overline{A})\left(\frac{\overline{A}}{A} + \frac{A}{\overline{A}} + 2\right) \\
&= |V|\left(\frac{\mathrm{cut}(A, \overline{A})}{|A|} + \frac{\mathrm{cut}(\overline{A}, A)}{|\overline{A}|}\right) \\
&= |V| \cdot \mathrm{RatioCut}(A, \overline{A}), \qquad (4.9)
\end{aligned}$$

where $|V| = |A| + |\overline{A}|$. Equation (4.8) holds for an arbitrary vector. Therefore, minimizing the RatioCut is equivalent to minimizing the inner product $\langle f|L|f\rangle$ by arranging the partition A and $\overline{A}$. Note that the norm of the vector $|f\rangle$ is

$$\langle f|f\rangle = \sum_i f_i^2 = |A|\frac{|\overline{A}|}{|A|} + |\overline{A}|\frac{|A|}{|\overline{A}|} = |V|, \tag{4.10}$$

which is independent of the partition. The vector $|f\rangle$ also has a property

$$\begin{aligned}
\sum_i f_i &= \sum_{i\in A}\sqrt{|\overline{A}|/|A|} - \sum_{i\in\overline{A}}\sqrt{|A|/|\overline{A}|} \\
&= |A|\sqrt{|\overline{A}|/|A|} - |\overline{A}|\sqrt{|A|/|\overline{A}|} = 0, \qquad (4.11)
\end{aligned}$$

which means that $|f\rangle$ is perpendicular to $|1\rangle$, the vector whose elements are all equal to unity.

In practice, the minimum of the RatioCut is computationally difficult to achieve because of the discreteness of the vector space of $|f\rangle$; one needs to test the values of $\langle f|L|f\rangle$

according to (4.7) for all possible partition. It is then usual to relax the condition and treat the vector $|f\rangle$ as a continuous vector. Therefore, in the relaxed condition, the problem to be considered is

$$\min_{f\in\mathbb{R}^{|V|}} \langle f|L|f\rangle \qquad \text{subject to} \qquad |f\rangle \perp |1\rangle,\ \langle f|f\rangle = |V|. \tag{4.12}$$

The resulting vector gives the best estimate for the minimum value of the RatioCut in the case of bisection. Finding such a vector can be expressed as an eigenvalue problem. The unnormalized graph Laplacian L is positive semi-definite (which is obvious from Eq. (4.8)) and the smallest eigenvalue is zero with the eigenvector $|1\rangle$ as long as L is irreducible, *i.e.* the network consists of one connected component; it means that L has eigenvalues $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_{|V|}$. Thus, ignoring the trivial eigenvector with the smallest eigenvalue, *i.e.* setting $|f\rangle \perp |1\rangle$, the eigenvector with the second smallest eigenvalue is the one which satisfies the condition (4.12). In order to obtain the partition from the obtained eigenvector, we need to digitalize it. By assigning each node v_i to A and $\overline{A}$ as

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0, \end{cases} \tag{4.13}$$

we obtain the bisection of the network.

The partitioning with respect to the Ncut can be discussed similarly. Defining a vector $|\hat{f}\rangle$ as

$$\hat{f}_i = \begin{cases} \sqrt{\mathrm{vol}(\overline{A})/\mathrm{vol}(A)} & v_i \in A, \\ -\sqrt{\mathrm{vol}(A)/\mathrm{vol}(\overline{A})} & v_i \in \overline{A}, \end{cases} \tag{4.14}$$

we obtain

$$\langle \hat{f}|L|\hat{f}\rangle = \mathrm{vol}(V)\cdot \mathrm{Ncut}(A,\overline{A}). \tag{4.15}$$

In this case, however, the norm of $|\hat{f}\rangle$ reads

$$\langle \hat{f}|\hat{f}\rangle = \sum_i \hat{f}_i^2 = |A|\,\frac{\mathrm{vol}(\overline{A})}{\mathrm{vol}(A)} + |\overline{A}|\,\frac{\mathrm{vol}(A)}{\mathrm{vol}(\overline{A})}, \tag{4.16}$$

which obviously depends on the partition. Hence, the minimum of the Ncut cannot be obtained as the second eigenvector of the unnormalized graph Laplacian.

Instead, let us consider the following similarity transformation:

$$\langle \hat{f}|L|\hat{f}\rangle = \langle g|D^{-1/2}LD^{-1/2}|g\rangle =: \langle g|L_{\mathrm{sym}}|g\rangle, \tag{4.17}$$

where

$$\begin{aligned} &|g\rangle = D^{1/2}|\hat{f}\rangle, \\ &\langle g|g\rangle = \langle \hat{f}|D|\hat{f}\rangle = \mathrm{vol}(A)\,\frac{\mathrm{vol}(\overline{A})}{\mathrm{vol}(A)} + \mathrm{vol}(\overline{A})\,\frac{\mathrm{vol}(A)}{\mathrm{vol}(\overline{A})} = \mathrm{vol}(V). \end{aligned} \tag{4.18}$$

The norm of the vector $|g\rangle$ is invariant under the different choice of the partitions and is perpendicular to $D^{1/2}|1\rangle$:

$$\langle 1|D^{1/2}|g\rangle = \langle 1|D|f\rangle = \mathrm{vol}(A)\sqrt{\frac{\mathrm{vol}(\overline{A})}{\mathrm{vol}(A)}} - \mathrm{vol}(\overline{A})\sqrt{\frac{\mathrm{vol}(A)}{\mathrm{vol}(\overline{A})}} = 0. \tag{4.19}$$

That is, under the relaxed condition where we replace the discrete vector space of $|g\rangle$ with the continuous space, the minimum value of the Ncut is given by

$$\min_{g\in\mathbb{R}^{|V|}} \langle g|L_{\mathrm{sym}}|g\rangle \qquad \text{subject to} \qquad |g\rangle \perp D^{1/2}|1\rangle,\ \langle g|g\rangle = \mathrm{vol}(V). \tag{4.20}$$

Analogously to the case of the unnormalized graph Laplacian L, the normalized graph Laplacian L_{sym} is also positive semi-definite and has a zero eigenvalue with the eigenvector $D^{1/2}|1\rangle$. Therefore, the eigenvector of L_{sym} with the second smallest eigenvalue gives the best estimate for the minimum value of the Ncut. In order to obtain the partition from the eigenvector, we follow the same procedure as (4.13). Note that the eigenvectors $\langle\hat{f}|$ can be regarded as the left-eigenvectors of L_{rw}, *i.e.* $\langle\hat{f}|L_{\mathrm{rw}} = \langle\hat{f}|\lambda$, or in terms of the unnormalized graph Laplacian L, $|\hat{f}\rangle$ is the generalized eigenvectors of L, *i.e.* $L|\hat{f}\rangle = \lambda D|\hat{f}\rangle$.

The spectral clustering can partition a network into more than two modules. For a given network, we consider a trial partition $\{A_j\}$ $(j = 1, 2, \ldots, m)$; note again that the number of modules m is an input. We then construct the following $|V| \times m$ indicator matrix H based on the trial partition:

$$H_{ij} = (a_1|1_1\rangle, \cdots, a_j|1_j\rangle, \cdots, a_m|1_m\rangle) = \begin{cases} a_j & i \in A_j, \\ 0 & \text{otherwise,} \end{cases} \tag{4.21}$$

where $|1_j\rangle$ $(j = 1, 2, \ldots, m)$ is an indicator vector whose ith element is unity if node i is in the module j and zero otherwise, and $a_j \neq 0$ is a constant factor.

We first explain the case of the RatioCut with the unnormalized graph Laplacian L. If we set $a_j = 1/\sqrt{|A_j|}$ in (4.21), the indicator matrix H has the following relation to the RatioCut:

$$\mathrm{Tr}\left(H^T L H\right) = \sum_{i=1}^{k} a_j^2 \langle 1_j|L|1_j\rangle = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_j, \overline{A}_j)}{|A_j|} = \mathrm{RatioCut}(A_1, \ldots, A_m), \tag{4.22}$$

where H^T is the transpose of H. For the calculation of $\langle 1_j|L|1_j\rangle$, we used Eq. (4.8). The equality (4.22) means that, if we adjust the trial partition $\{A_j\}$ so that $\mathrm{Tr}\left(H^T L H\right)$ may be minimized, it turns out to be the correct partition in the sense of the RatioCut. Note also that $H^T H = I$ holds. Relaxing the discrete-space problem to the continuous-space problem again, we end up with the following:

$$\min_{H\in\mathbb{R}^{|V|\times m}} \mathrm{Tr}\left(H^T L H\right) \qquad \text{subject to} \qquad H^T H = I. \tag{4.23}$$

The solution of (4.23) is given by H which contains the first m eigenvectors of L with the smallest m eigenvalues as columns. Finally, from the matrix H which is obtained as the result of the eigenvector problem, we can obtain the partition with the following procedure (we will explain why it works below). We regard each row of H as an m-dimensional vector, where the ith row vector corresponds to node i, and apply the widely-known k-means method [102] to these vectors. For the k-means method, there exists a convenient heuristic algorithm called the Loyd algorithm [99].

In order to understand the last step of the procedure, let us consider the ideal case where the network consists of m pieces of mutually disjoint components A_j $(j = 1, 2, \ldots, m)$. In such a case, L is expressed as a block-diagonal matrix with m blocks and there exists m degenerated eigenvectors with the zero eigenvalue (the smallest eigenvalue). Each of them is an indicator vector up to the degrees of freedom of a constant factor a_j. If we align those indicator vectors to form the indicator matrix H, each row of it becomes the m-dimensional vector which indicates the module which the node belongs to. A non-ideal case can be regarded as the perturbed one of the ideal case; it is expected that the row vectors of H in the non-ideal case are close to those of the ideal case as long as the perturbation is small (see Ref. [101] and references therein for a quantitative argument). Note that, while the correct partition is obtained by the eigenvectors of the unnormalized graph Laplacian L (*i.e.* the indicator vectors) in the ideal case, the indicator matrix H which is obtained as the result of the discrete optimization problem does not necessarily consist of the eigenvectors of L in non-ideal cases.

For the normalized graph Laplacian L_{sym}, note that the eigenvectors are not the indicator vectors even in the ideal case, but are $D^{1/2}|1_j\rangle$ $(j = 1, \ldots, m)$, and such a difference appears in the non-ideal case as well. Analogously to the unnormalized graph Laplacian, by setting $a_j = 1/\sqrt{\mathrm{vol}(A_j)}$ in (4.21) and denoting the corresponding indicator matrix as $\hat{H}$, we have

$$\mathrm{Tr}\left(\hat{H}^T L \hat{H}\right) = \sum_{i=1}^{k} a_j^2 \langle 1_j | L | 1_j \rangle = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_j, \overline{A}_j)}{\mathrm{vol}(A_j)} = \mathrm{NCut}(A_1, \ldots, A_m). \tag{4.24}$$

We also have $\hat{H}^T D \hat{H} = I$. Therefore, under the relaxed condition of the continuous vector space, the problem becomes as follows:

$$\min_{\hat{H} \in \mathbb{R}^{|V| \times m}} \mathrm{Tr}\left(\hat{H}^T L \hat{H}\right) \qquad \text{subject to} \qquad \hat{H}^T D \hat{H} = I, \tag{4.25}$$

or using $\hat{T} = D^{1/2} \hat{H}$,

$$\min_{\hat{T} \in \mathbb{R}^{|V| \times m}} \mathrm{Tr}\left(\hat{T}^T L_{\mathrm{sym}} \hat{T}\right) \qquad \text{subject to} \qquad \hat{T}^T \hat{T} = I. \tag{4.26}$$

Thus, in order to find the partition using the k-means method, we can use the first m eigenvectors of L_{sym} with the smallest m eigenvalues in $\hat{T}$, as well as the first m left-eigenvectors of L_{rw}, or equivalently the first m generalized eigenvectors of $L|\hat{f}\rangle = \lambda D|\hat{f}\rangle$

in $\hat{H}$. In the case of clustering with $\hat{T}$, the elements of the eigenvectors may happen to be very small compared to that of $\hat{H}$ because of the factor $D^{1/2}$, *e.g.* a weighted network containing nodes with extremely low degrees. Hence, instead of using the m-dimensional row vectors themselves for the k-means method, it is required to normalize each norm to unity beforehand [101, 116].

Although the method with the unnormalized graph Laplacian L looks more tractable, there are some arguments [101] that one should use the normalized graph Laplacian, L_{rw} especially, rather than the unnormalized one, because it has many plausible properties that a correct partition should have.

4.4.3 Random walk interpretation of the spectral clustering

The method of Ncut can be interpreted in terms of the random walk. Here, we assume the knowledge of Sec. 5.3.1; and use Eqs. (5.26) and (5.8). Let us consider the stationary distribution of a random walk on a network. We assume that the network is connected and is non-bipartite so that the unique stationary state is obtained. We then have a relation

$$\mathrm{Ncut}(A, \overline{A}) = P(\overline{A}|A) + P(A|\overline{A}), \tag{4.27}$$

where $P(X|Y)$ is the transition probability from the module Y to the module X. It can be proved as follows. From Eqs. (5.26) and (5.8) in Sec. 5.3.1, we see that the stationary probability distribution that the random walker is in the module A is $\mathrm{vol}(A)/\mathrm{vol}(V)$ and the joint probability that the random walker is in the module A before a transition and is in the module $\overline{A}$ after the transition $P(\overline{A}, A)$ is

$$P(\overline{A}, A) = q_{A\curvearrowright} = \frac{1}{\mathrm{vol}(V)} \sum_{i \in A, j \in \overline{A}} w_{ij} = \frac{\mathrm{cut}(A, \overline{A})}{\mathrm{vol}(V)}. \tag{4.28}$$

Thus, the transition probability $P(\overline{A}|A)$ reads

$$P(\overline{A}|A) = \frac{P(\overline{A}, A)}{P(A)} = \frac{\mathrm{cut}(A, \overline{A})}{\mathrm{vol}(V)} \frac{\mathrm{vol}(V)}{\mathrm{vol}(A)} = \frac{\mathrm{cut}(A, \overline{A})}{\mathrm{vol}(A)}. \tag{4.29}$$

Hence, from the definition of Ncut, we have Eq. (4.27). It means that the correct partition in the sense of Ncut is the partition in which the random walker seldom walks between the modules. It is consistent with the interpretations of the other methods with a random walk.

4.5 Modularity

4.5.1 Modularity and its extensions

As we mentioned in section 4.2, the modularity was first introduced as a criterion to determine where to stop the algorithm of Girvan and Newman [57] by the same authors [114].

They claimed that one should accept the partition which gives the highest modularity. The modularity is originally defined in terms of the graph quantities.

The idea of the modularity is as follows. The modularity is a quality function of the partitions which evaluates how densely links are connected within the modules and the one with the highest value is regarded as the correct partition. The quality function compares the structure of the network in question to a network with no community structure in principle, which is called a *null model*. A partition scores high if the modules have the internal links more than they are expected to have in the null model. A typical model for the null model is *configuration model* (it is a part of the definition of the modularity in many papers). The configuration model is a network which is generated by randomizing the links of the original network data while keeping the degree of each node. Let us first consider the number of links expected for a pair of nodes i and j. For a given link, the probability that one end of the link is i and the other end of the links is j reads $k_i/2L \times k_j/2L$, where k_i and k_j are the degrees of the nodes i and j, respectively, and L is the total number of links. Since i and j can be either end of L links, the expectation number of the links between them is

$$2L \times \frac{k_i}{2L}\frac{k_i}{2L} = \frac{k_i k_j}{2L}. \tag{4.30}$$

In the original network, the number of links between i and j is given by the element of the adjacency matrix A_{ij}. With these quantities, the modularity is defined as follows:

$$Q = \frac{1}{2L}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2L}\right)\delta(C_i, C_j), \tag{4.31}$$

where the sum is taken over every node pair, and the delta function $\delta(C_i, C_j)$ gives one if the node i and j are in the same module. The modularity compares the actual connectivity of the node pair, A_{ij}, and the corresponding expectation value in the configuration model, $k_i k_j/2L$, for each module. The factor $1/2L$ is for the normalization. Equation (4.31) can also be written as the sum over modules as follows:

$$Q = \sum_{i=1}^{m}\left[\frac{l_i}{L} - \left(\frac{d_i}{2L}\right)^2\right]. \tag{4.32}$$

Here, the sum is taken over every module and l_i is the number of links within the module i. The total degree d_i of the module i is twice the number of links l_i within the module plus the number of links which is connected to outside of the module. The expression (4.32) is often more tractable. The larger the value of the modularity Q is, the stronger the community structure is. Therefore, the goal of the method of the modularity is to find the partition which gives the largest value of the modularity Q.

There are many extensions to the modularity. The spectral interpretation and optimization was discussed by Newman himself [113]. As we mentioned above, the extension to the directed networks were discussed in Refs. [81,89]. Muff *et al.* considered a quantity called *local modularity*, although the modularity is a quantity of the whole network. The philosophy of the local modularity is as follows. The configuration model that is used as the null model in the modularity assumes that any nodes have possibility to be connected irrespectively of the distance between them. In real networks, however, it is not true and there must be a scope for each node, and hence they modified the modularity to a local quantity. Reichardt and Bornholdt [124] considered the community detection as a Potts model of a spin system and obtained the generalization of the modularity as follows:

$$Q_\gamma = \sum_{i=1}^{m} \left[\frac{l_i}{L} - \gamma \left(\frac{d_i}{2L} \right)^2 \right], \tag{4.33}$$

where the parameter γ is the factor which controls the balance between the actual network and the null model.

4.5.2 Resolution limit of the modularity

As an important property of the modularity, its resolution limit was derived analytically by Fortunato and Barthélemy [54] in 2007. It is the limit of a module size below which the method cannot detect even when its quality function is correctly optimized. The modules with the size less than the resolution limit are merged in the process of optimization, and thus the resolution limit is the critical size above which a module does not get merged. We will discuss the origin and the meaning of the resolution limit in great detail in the next chapter; in this section, we go over how it is derived for the modularity. We follow the explanation by Good *et al.* [61] rather than the original one by Fortunato and Barthélemy [54].

As shown in Fig. 4.3, we consider two modules in a network and examine if the quality function Q increases or decreases by merging them. If the size of the modules are too small, the modularity Q gives a higher value when we merge them, *i.e.*, they are not resolved. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the modules to be evaluated and we refer to $\mathcal{M}_3$ as the rest of the network. Note that $\mathcal{M}_3$ may consist of many modules. We refer to $\mathcal{M}_{12}$ as the merged module of $\mathcal{M}_1$ and $\mathcal{M}_2$. According to Eq. (4.32), the difference of the modularity ΔQ reads

$$\begin{aligned}\Delta Q &= \frac{l_{\text{int}}}{L} - \left(\frac{d_1 + d_2}{2L} \right)^2 + \left(\frac{d_1}{2L} \right)^2 + \left(\frac{d_2}{2L} \right)^2 \\ &= \frac{l_{\text{int}}}{L} - \frac{d_1 d_2}{2L^2},\end{aligned} \tag{4.34}$$

where d_1 and d_2 are the total degrees of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, and l_{int} is the number of links between them. The modules of size d_1 and d_2 are not resolved when $\Delta Q > 0$, *i.e.*,

$$l_{\text{int}} > \frac{d_1 d_2}{2L}. \tag{4.35}$$

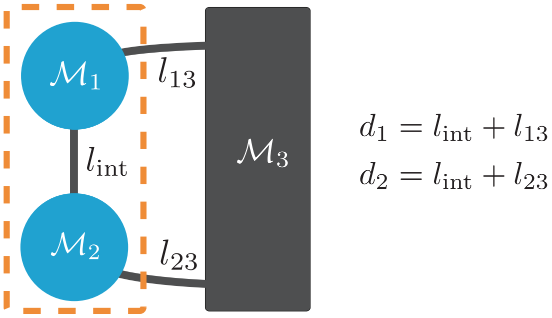


Figure 4.3: A schematic picture of two modules $\mathcal{M}_1$ and $\mathcal{M}_2$ in a network. The subgraph $\mathcal{M}_3$ indicates the rest of the network and may consist of many modules. The number of links between the module $\mathcal{M}_1$ and $\mathcal{M}_3$ is denoted by l_{13} and the number of links between the module $\mathcal{M}_2$ and $\mathcal{M}_3$ is denoted by l_{23}.

For simplicity, let us consider the case where the numbers of internal links are the same, $d_1 = d_2 = 2l$. The most extreme case is when $l_{\rm int} = 1$, and therefore we have the resolution limit

$$l < \sqrt{\frac{L}{2}}. \tag{4.36}$$

The resolution limit of the modularity is ruled by the total number of links L or the total degree K. It means that, if a module with size less than $\sqrt{2L} = \sqrt{K}$ was detected, one should doubt that the module may consist of more than two modules which we would intuitively regard as communities.

4.5.3 Random walk interpretation of the modularity

The interpretation of the modularity in terms of a random walk was shown by Delvenne *et al.* [44] in 2010. (See also [135].) They introduced a quality function called *stability* which can be regarded as a generalization of the modularity. We again assume the knowledge of Sec. 5.1 and use the opposite convention for the time evolution of the random walker to Ref. [44]. (We define the time evolution matrix as that of the ket state $|p\rangle$, instead of the bra state $\langle p|$.)

Now, let us consider the following stochastic process in order to introduce the stability. We partition the network into m modules and put a different label $\xi_i \in \mathbb{R}$ $(i = 1, 2, \ldots, m)$ to each of them as shown in Fig. 4.4; each node has the label of the module to which it belongs. We consider a random walk in its stationary state on the network and let the random walker emits the labels of the nodes as it walks around. Then, we regard the emitted label at a discrete time t as a stochastic variable X_t in the Markov chain. If the partitioning is the correct one, the random walker seldom escapes from a module, which means that the value of the auto-covariance of X_t, *i.e.*

$$\mathrm{cov}\left[X_{t+\tau}, X_t\right] \equiv \langle X_{t+\tau}\, X_t\rangle - \langle X_t\rangle^2 \qquad (t = 0, 1, 2, \ldots, N), \tag{4.37}$$

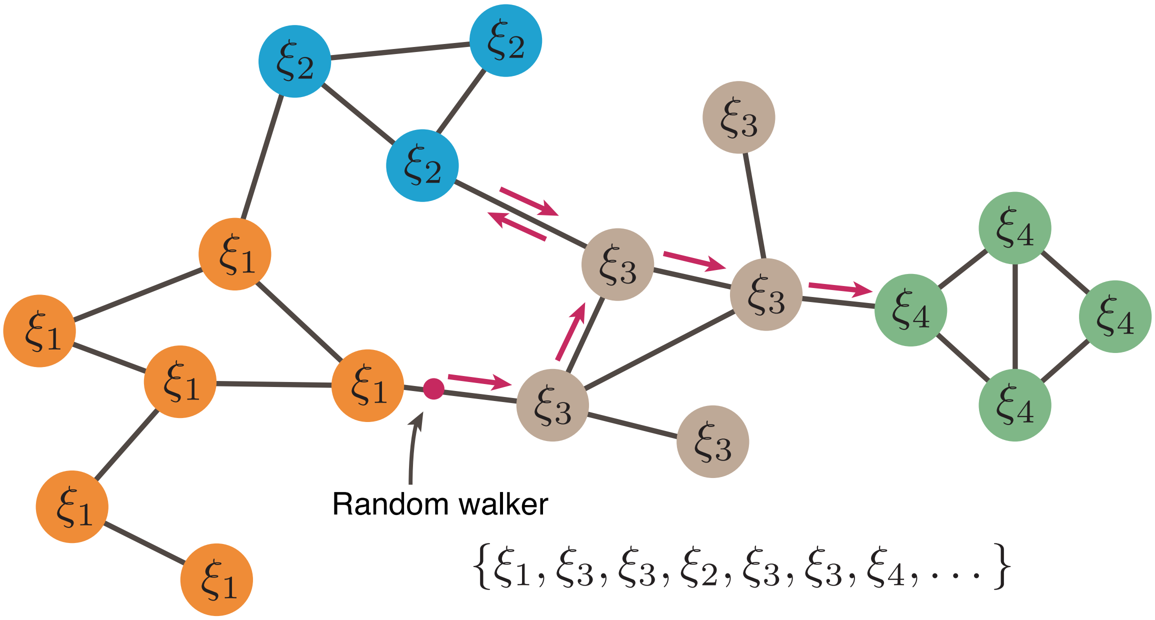


Figure 4.4: An example of the random walk and its array of labels for the stability.

stays high for a given time interval τ. Here, the angular bracket $\langle \cdots \rangle$ means the average with respect to the stationary distribution of the random walker and we used a property of the stationarity that $\langle X_{t+\tau} \rangle = \langle X_t \rangle$. We set N to the total number of nodes $|V|$ in the network V. The auto-covariance $\operatorname{cov}[X_{t+\tau}, X_t]$ can be expressed in the matrix form as follows. We denote the stationary distribution of the random walker as $|\pi\rangle$ and its one-step time evolution matrix as T, which we will explain in Sec. 5.1. Using the indicator matrix H in (4.21) with $a_j = 1$, we can express the value of the label at the node α, $x_t(\alpha) \in X_t$, as $x_t(\alpha) = \sum_{j=1}^{m} H_{\alpha j}\xi_j$. We then have the average of X_t,

$$\langle X_t \rangle = \langle x_t | \pi \rangle = \langle \xi | H^T | \pi \rangle, \tag{4.38}$$

where H^T is the transpose of H. The moment $\langle X_{t+\tau}\, X_t \rangle$ is given by

$$\begin{aligned}
\langle X_{t+\tau}\, X_t \rangle &= \sum_{\alpha\beta}^{N} x_{t+\tau}(\beta)\, x_t(\alpha)\, p_\tau(\beta, \alpha) \\
&= \sum_{\alpha\beta}^{N} x_{t+\tau}(\beta)\, x_t(\alpha)\, p_\tau(\beta|\alpha)\pi_\alpha \\
&= \sum_{\alpha\beta}^{N} \left(\sum_{i=1}^{m} \xi_i H^T_{i\beta} \right) \left(\sum_{j=1}^{m} H_{\alpha j}\xi_j \right) T^\tau_{\beta\alpha}\, \pi_\alpha \\
&= \langle \xi | H^T T^\tau \Pi H | \xi \rangle,
\end{aligned} \tag{4.39}$$

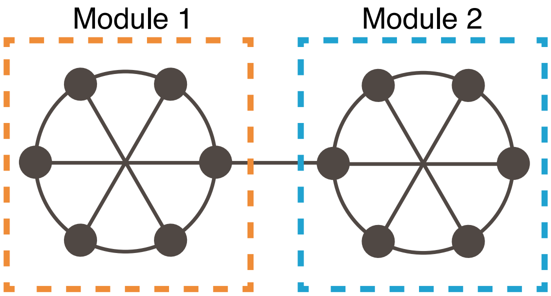


Figure 4.5: The random walker oscillates between the modules 1 and 2.

where $\Pi = \mathrm{diag}(\pi_\alpha)$. From Eqs. (4.38) and (4.39), we have

$$\mathrm{cov}\left[X_{t+\tau}, X_t\right] = \langle \xi | R_\tau | \xi \rangle, \tag{4.40}$$

$$R_\tau = H^T \left(T^\tau \Pi - |\pi\rangle\langle\pi|\right) H. \tag{4.41}$$

The matrix R_τ is called *clustered auto-covariance matrix* of the network.

If the network has a clear community structure, the random walker in the module i tends to stay in the same module, which means that value of $(R_\tau)_{ii}$ is large. Hence, the trace of the clustered auto-covariance matrix $\mathrm{tr}R_\tau$ sounds appropriate for the quality function. There is, however, one more restriction that we need to add.

Let us consider the case of a network in Fig. 4.5 as an example. In this network, the auto-covariance oscillates because the random walker tends to escape and come back to the module periodically. Since we are interested in the process that the random walker stays in a module and wish to eliminate the process that the random walker returns to the module where it left, the *stability* of the clustering is defined as

$$r(t; H) = \min_{0 \le s \le t} \mathrm{tr}R_s(H). \tag{4.42}$$

Finally, the stability curve $r(t)$ is obtained as

$$r(t) = \max_H r(t; H). \tag{4.43}$$

It means that we compute the minimum value $r(t; H)$ of $\mathrm{tr}R_t(H)$ for every possible partition and choose the one which gives the maximum among them.

At $t = 1$, the stability $r(1; H)$ coincides with the modularity Q. To see this, let us first consider the second term in Eq. (4.41). Observe that the element $(H^T|\pi\rangle)_i$ equals the sum of the visiting frequencies of the random walker to the nodes in the module i. We thus have, according to Eq. (5.3),

$$\mathrm{tr}(H^T|\pi\rangle\langle\pi|H) = \sum_{i=1}^{m} \left(\frac{\sum_{\alpha\in i} k_\alpha}{2L} \right)^2 = \sum_{i=1}^{m} \left(\frac{d_i}{2L} \right)^2. \tag{4.44}$$

For the first term in Eq. (4.41), since $T\Pi = AD^{-1}D/2L$, we have

$$\mathrm{tr}(H^T T\Pi H) = \frac{1}{2L}\mathrm{tr}(H^T AH) = \sum_{i=1}^{m} \frac{l_i}{L}. \tag{4.45}$$

Therefore, we obtain

$$Q = \mathrm{tr}R_{t=1}(H) = r(1; H). \tag{4.46}$$

When we set $t = 0$, the stability detects each node in the network as a single module, while in the case of larger values of the Markov time t, it detects larger modules. Hence, we can regard the Markov time t as a resolution parameter of communities. Other than the connection to the modularity, connections to other quality functions at different time are also discussed in Ref. [44].

Chapter 5

The map equation

In this chapter, we first explain the dynamics of a random walker on a network and then describe the formulation of the method of the map equation and some other extensions. After that, we will discuss the resolution limit of the map equation [78], which is one of the main results of the present thesis.

5.1 Time evolution of a random walker

We first take a look at the dynamics of a random walker. For a uniform random walk, the walker should have an equal probability of transition to each neighboring node, and thus the transition probability to one neighboring node from the node α should be $1/k_\alpha^{\rm out}$, where $k_\alpha^{\rm out}$ is the out-degree of the node α. Accordingly, the time evolution of the probability distribution vector $\mathbf{p}_t$ at discrete time (*Markov time*) t is expressed as

$$\mathbf{p}_{t+1} = AD^{-1}\mathbf{p}_t \equiv T\mathbf{p}_t, \tag{5.1}$$

where A is the adjacency matrix, $D = \mathrm{diag}(k_\alpha)$, and T is the transition matrix. Equation (5.1) can also be expressed as

$$\begin{aligned}\mathbf{p}_{t+1} - \mathbf{p}_t = \left(AD^{-1} - \mathbf{I}\right)\mathbf{p}_t &= -(D-A)D^{-1}\mathbf{p}_t \\ &\equiv -LD^{-1}\mathbf{p}_t \\ &\equiv -L_{\rm rw}\mathbf{p}_t, \end{aligned} \tag{5.2}$$

where I is the identity matrix, $L = D - A$ is called the unnormalized graph Laplacian, and $L_{\rm rw} = LD^{-1}$ and $L_{\rm sym} = D^{-1/2}LD^{-1/2}$ are called the normalized graph Laplacians. As we explained in the last chapter, the methods of the spectral clustering make use of the properties of the unnormalized and the normalized graph Laplacians [101].

In the case of an undirected unweighted network, since the ergodicity is satisfied, the stationary distribution $\mathbf{p}_{t\to\infty} = \pi$ is given by

$$\pi_\alpha = \frac{k_\alpha}{2L}. \tag{5.3}$$

The stationarity condition $T\pi = \pi$ can be readily confirmed by substituting Eq. (5.3) into Eq. (5.1). In the case of a directed network, the ergodicity may not be satisfied, *i.e.*, there may be some sources and sinks of flow. Thus, the random walker may not have a unique stationary state with a nonzero probability at each node, or it may not have a stationary state at all.

In order to avoid the problem of the ergodicity, one can consider a random surfer which has an additional process called *teleportation*. In the case where there is no node with no out-going links, the equation of the time evolution of a random surfer is expressed as

$$p_{\alpha,t+1} = \mu \sum_{\beta} T_{\alpha\beta}\, p_{\beta,t} + (1-\mu) v_{\alpha}, \tag{5.4}$$

where $1-\mu$ $(0 \leq \mu < 1)$ is the teleportation rate and v_α is an element of the preference vector, *i.e.* the frequency at which the random surfer teleports to the node α. It says that the random surfer moves in the same way as the random walk with probability μ and it teleports to other nodes according to the distribution $\mathbf{v}$ with probability $1-\mu$. If there exists a node with no out-going links, the random surfer always teleports at that node; mathematically, it corresponds to replacing the row of the transition matrix in which all elements are equal to zero with the elements of the preference vector. Since the preference vector satisfies the normalization condition $\sum_\alpha v_\alpha = 1$, the time evolution in Eq. (5.4) conserves the probability, $\sum_\alpha p_{\alpha,t+1} = 1$.

Thanks to the teleportation, every node has a nonzero probability, *i.e.* the ergodicity is recovered, and the existence of a unique stationary state is proved by the Perron-Frobineous theorem (as long as the dynamics is aperiodic). The method of the random surfer was used in the celebrated paper of the PageRank [34], the fundamental search technique of Google. If the teleportation rate $1-\mu$ were too large, the stationary distribution would become close to the uniform distribution and insensitive to the structure of the underlying network, and thus it needs to be small enough; it is empirically known that $1-\mu = 0.15$ shows good performance in many cases. Although the preference vector is often chosen to be uniform for each node, $v_\alpha = 1/N$ (N is the total number of nodes), it is arbitrary in principle as long as the ergodicity is obtained [84]. In the stationary limit, since we have $p_{\alpha,t+1} = p_{\alpha,t} = \pi_\alpha$, the formal stationary distribution of Eq. (5.4) becomes [84]

$$\begin{aligned} \pi_\alpha &= (1-\mu) \sum_{\beta} (I - \mu T)^{-1}_{\alpha\beta}\, v_\beta \\ &= v_\alpha + \sum_{\beta} \sum_{k=1}^{\infty} \mu^k \left(T^k_{\alpha\beta} - T^{k-1}_{\alpha\beta} \right) v_\beta. \end{aligned} \tag{5.5}$$

Here we expanded π_α with respect to μ.

5.2 The map equation

The method of the map equation was invented by M. Rosvall and C. Bergstrom in 2008 [131]. The map equation is the name of a quality function to be optimized in order to obtain the community structure and the algorithm to optimize the map equation is often called *Infomap* [7]. The method of the map equation is an information-theoretic approach to the community detection and is about the encoding of a trajectory of a (uniform) random walker on the network in the stationary state. As we mentioned above, although the module is thought of as a region where a random walker stays for a relatively long time, the communities are defined as the partition of the network which optimizes the quality function. We first review the formulation of the map equation and then analyze the properties of it in great detail.

5.2.1 Derivation of the (original) map equation

We start from encoding of the movement of a random walker on the network and consider the minimal description length of its trajectory. In this section, we assume that the network is undirected and unweighted. A simple way to describe it is to put labels on each node. By recording the nodes that the random walker visited, one can reconstruct the trajectory of the random walker using the array of the labels as shown in Fig. 5.1(a). Under this labeling, the minimal description length averaged over the transition steps is given by the Shannon entropy as

$$H = -\sum_{\alpha} p_{\alpha} \log p_{\alpha}, \tag{5.6}$$

where p_α is the stationary probability that the random walker exists at the node α and the basis of the logarithm is chosen to be two. This is one way of encoding the movement of the random walker. However, there is a better way of encoding, so that one can compress the description length even more.

The above description was a one-level description; we used only codes of one kind, or one *codebook*, to indicate the node where the random walker is. We now consider a two-level description. That is, we use two kinds of codebooks as follows:

module codebook Its code indicates the movement between modules.

node codebook Its code indicates the movement within a module and the exiting from the module.

We can readily see how the two-level description compresses the description length in the example of Fig. 5.1(b). If we divide the network into two modules, the blue module and the red module, the codes which are used for the movements in the blue module can be reused for the movements in the red module, and therefore, it results in compressing the description length. By choosing the partition of the network properly so that the random walker may rarely travel across the modules (notice that the intuitive definition of a module appears here!), one obtains the maximum compression, *i.e.* the minimum description length under this procedure. The minimum description length $L(\mathbf{M})$ under the partition $\mathbf{M}$ is given as follows:

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_{i\circlearrowright} H(\mathcal{P}^i), \tag{5.7}$$

where

$$q_{\curvearrowright} = \sum_{i=1}^{m} q_{i\curvearrowright}, \tag{5.8}$$

$$H(\mathcal{Q}) = -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{q_{\curvearrowright}} \log\left(\frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\right), \tag{5.9}$$

$$p_{i\circlearrowright} = q_{i\curvearrowright} + \sum_{\alpha \in i} p_{\alpha}, \tag{5.10}$$

$$H(\mathcal{P}^i) = -\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}} \log\left(\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}}\right) - \sum_{\alpha \in i} \frac{p_{\alpha}}{p_{i\circlearrowright}} \log\left(\frac{p_{\alpha}}{p_{i\circlearrowright}}\right). \tag{5.11}$$

The label i is for a module and m is the number of modules. As defined in Eq. (5.8), $q_{\curvearrowright}$ represents the sum of the exiting probability from each module $q_{i\curvearrowright}$ and $H(\mathcal{Q})$ is the Shannon entropy for the movement between modules. Similarly, $p_{i\circlearrowright}$ is the probability that the random walker stays inside the module i and the probability that it escapes from the module. Equation (5.11) is the corresponding Shannon entropy. Notice that the encoding for the exiting from a module is required in Eqs. (5.10) and (5.11). Once an exiting code appeared during the movement within a module, one realizes that the next code is for the transition between the modules, which means that the codes for the the transition within a modules can also be reused for the transition between the modules. Without them, one has to choose the codes for the transitions between the modules so that they may not overlap with the codes for the transition within a module.

Substituting Eqs. (5.8), (5.9), (5.10), and (5.11) into Eq. (5.7), we can write the description length given by the map equation as

$$\begin{aligned} L(\mathbf{M}) &= -\sum_{i=1}^{m} q_{i\curvearrowright} \log\left(\frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\right) - \sum_{i=1}^{m} q_{i\curvearrowright} \log\left(\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}}\right) - \sum_{i=1}^{m} \sum_{\alpha \in i} p_{\alpha} \log\left(\frac{p_{\alpha}}{p_{i\circlearrowright}}\right) & (5.12) \\ &= q_{\curvearrowright} \log q_{\curvearrowright} - 2\sum_{i=1}^{m} q_{i\curvearrowright} \log q_{i\curvearrowright} + \sum_{i=1}^{m} p_{i\circlearrowright} \log p_{i\circlearrowright} - \sum_{\alpha} p_{\alpha} \log p_{\alpha}. & (5.13) \end{aligned}$$

Hence, $L(\mathbf{M})$ in (5.7) and (5.13) are the quality function, and the correct modules are obtained as the partition where $L(\mathbf{M})$ gives the minimum value. Hereafter, we use the word the description length to indicate the description length given by the map equation, for simplicity. Note that we can neglect the last term in (5.13) because it does not depend on how we partition the network.

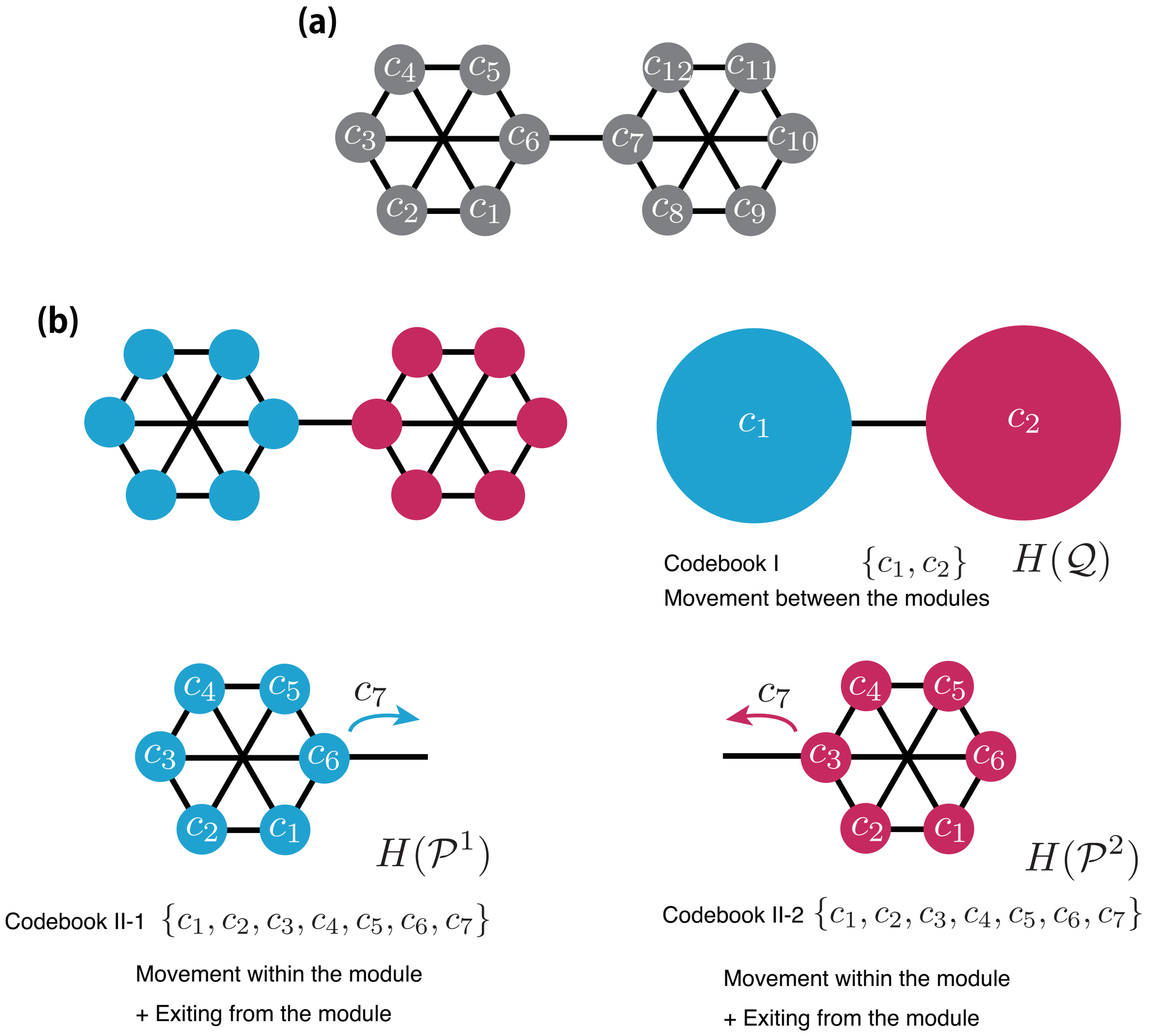

(a)
c_4 c_5 c_3 c_6 c_2 c_1
c_{12} c_{11} c_7 c_{10} c_8 c_9
(b)
c_1 c_2
Codebook I $\{c_1, c_2\}$ $H(\mathcal{Q})$
Movement between the modules
c_4 c_5 c_3 c_6 c_2 c_1 c_7
$H(\mathcal{P}^1)$
Codebook II-1 $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
Movement within the module
+ Exiting from the module
c_7 c_4 c_5 c_3 c_6 c_2 c_1
$H(\mathcal{P}^2)$
Codebook II-2 $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
Movement within the module
+ Exiting from the module

Figure 5.1: (a) Labeling of nodes without partitioning into the modules. (b) Labeling of nodes with the module structure.

The program code for the map equation and a nice demonstration is distributed on [8]. In the demonstration, the encoding is done according to the Huffman coding. Note, however, that the choice of the coding algorithm is not important in both theory and application; we always deal with the Shannon entropy which is the lower bound of the code length, and thus the probability distribution of the random walker is enough for the optimization.

5.2.2 The map equation in directed networks

In the case of directed networks, the ergodicity may not be satisfied. In such a case, as we discussed in Sec. 5.1, we need to consider a random surfer instead of a random walker.

There are two ways to discuss the description length of the random surfer; the encoding which records the teleportation [131] and the one which ignores the teleportation [84,133].

In the former case, we use the stationary distribution π_α in Eq. (5.5) for the node visiting probability. For the probability that the random surfer exits from a module, we have [131]

$$q_{i,\curvearrowright} = \mu \sum_{\beta \notin i} \sum_{\alpha \in i} T_{\beta\alpha}\, p_\alpha + (1-\mu) \frac{N - N_i}{N-1} \sum_{\alpha \in i} p_\alpha, \tag{5.14}$$

where N_i is the number of nodes in the module i. The scheme which records the teleportation is a straightforward extension of the formulation for the undirected network and was the one suggested in the original paper of the map equation [131]. The teleportation process, however, ignores the topology of the network, and thus causes incorrect cohesion between the nodes.

By ignoring the teleportation, we obtain a result with better accuracy. Following Ref. [133], we define the node visiting probability as

$$\widetilde{\pi}_\alpha = \sum_\beta q_{\beta \curvearrowright \alpha}. \tag{5.15}$$

The summand $q_{\beta \curvearrowright \alpha}$ is the transition probability from the node β to α,

$$q_{\beta \curvearrowright \alpha} = \mu\, T_{\alpha\beta}\, \pi_\beta, \tag{5.16}$$

where μ is the probability that the random surfer does not teleport and π_β is the node visiting probability in Eq. (5.5). Hence, Eq. (5.15) is the node visiting probability in which the teleportation is ignored. Note that the distribution $\widetilde{\pi}_\alpha$ is not a stationary distribution; the in-flow and the out-flow of the probability are no longer equal, *i.e.* the balance condition is broken, because we ignore the teleportation. Although the map equation fundamentally considers the stationary distribution of a random surfer, we can formulate the map equation for an arbitrary state in principle.

We need another modification to the map equation when we ignore the teleportation. Remember that, in Eq. (5.7), we used the exiting probability $q_{i\curvearrowright}$ for the entropy to specify the module that the random surfer is in and it was due to the balance condition. Now we need to distinguish the exiting probability $q_{i\curvearrowright}$ and the *entering probability* $q_{i\curvearrowleft}$, *i.e.*, the probability that the random surfer enters the module i. Hence, we replace $q_{\curvearrowright} H(\mathcal{Q})$ with

$$q_{\curvearrowleft} H(\mathcal{Q}) = -q_{\curvearrowleft} \sum_{i=1}^{m} \frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}} \log\left(\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}\right), \tag{5.17}$$

where $q_{\curvearrowleft} = \sum_i q_{i\curvearrowleft}$. For the probability $p_{i\circlearrowright} = q_{i\curvearrowright} + \sum_{\alpha \in i} p_\alpha$, we do not replace $q_{i\curvearrowright}$ with $q_{i\curvearrowleft}$.

5.2.3 Hierarchical version of the map equation

There is a way to compress the description length of the random walker further more. In the original version of the map equation, we considered the two types of the codebook, the one to describe the movement between the modules and the other to describe the movement within a module (and escape from it); it is sometimes called the two-level method. One can increase the levels of description and the extended version is called the multi-level method or the *hierarchical map equation* [133]. That is, in the case of the three-level method for example, in addition to the structure of the nodes and the modules, the supermodules, *i.e.* the modules of modules, are considered. An additional codebook records the transitions between the supermodules. One can extend the hierarchy as long as the description length is compressed. Hierarchical clustering does not only compress the code length, but is also a natural way to observe the network structure. In a large network, it is usual that a module at a coarser level has some modules of finer scale inside. Therefore, it is not fair to detect the modules only of one scale.

The multi-level method is formulated as follows. Here, we consider directed unweighted networks and use the scheme which ignores the teleportation, *i.e.*, we need to distinguish the exiting probability and the entering probability of a module. At the coarsest level with the partition $\mathbf{M}$, we have the quality function of the multi-level method as

$$L(\mathbf{M}) = q_{\curvearrowleft} H(\mathcal{Q}) + \sum_{i=1}^{m} L(\mathbf{M}_i). \tag{5.18}$$

The first term is the same as the two-level method and we have the sum of the description length $L(\mathbf{M}_i)$ of each module. For the description length $L(\mathbf{M}_i)$ of the module i with m_i submodules, we have

$$L(\mathbf{M}_i) = q_{i\circlearrowright} H(\mathcal{Q}_i) + \sum_{j=1}^{m_i} L(\mathbf{M}_{ij}). \tag{5.19}$$

The probability $q_{i\circlearrowright}$ in this case represents

$$q_{i\circlearrowright} = q_{i\curvearrowright} + \sum_{j} q_{ij\curvearrowleft}, \tag{5.20}$$

i.e., the sum of the exiting probability $q_{i\curvearrowright}$ and the probability $q_{ij\curvearrowleft}$ that the random surfer enters into the submodule j in the module i. In the first term of Eq. (5.19), $H(\mathcal{Q}_i)$ is the corresponding Shannon entropy. The hierarchy of the codebooks of the submodules continues until it reaches the finest level. There, we have

$$L(\mathbf{M}_{ij\ldots k}) = p_{ij\ldots k\circlearrowright} H(\mathcal{P}_{ij\ldots k}), \tag{5.21}$$

where $p_{ij\ldots k\circlearrowright}$ is the sum of the exiting probability from the submodule k and the node-visit frequencies in it, whereas $H(\mathcal{P}_{ij\ldots k})$ is the corresponding Shannon entropy. Assembling them all, we can express the multi-level method as follows:

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} q_{i\circlearrowright} H(\mathcal{Q}_i) + \sum_{i=1}^{m}\sum_{j=1}^{m_i} q_{ij\circlearrowright} H(\mathcal{Q}_{ij}) + \cdots + \sum_{ij\ldots k} p_{ij\ldots k\circlearrowright} H(\mathcal{P}_{ij\ldots k}), \tag{5.22}$$

with

$$H(\mathcal{Q}) = -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{q_{\curvearrowright}} \log \frac{q_{i\curvearrowright}}{q_{\curvearrowright}}, \tag{5.23}$$

$$\sum_{i=1}^{m} H(\mathcal{Q}_i) = -\sum_{i=1}^{m} \frac{q_{i\curvearrowleft}}{q_{i\circlearrowright}} \log \frac{q_{i\curvearrowleft}}{q_{i\circlearrowright}} - \sum_{i=1}^{m}\sum_{j=1}^{m_i} \frac{q_{ij\curvearrowright}}{q_{i\circlearrowright}} \log \frac{q_{ij\curvearrowright}}{q_{i\circlearrowright}}, \tag{5.24}$$

$$\sum_{ij\ldots k} H(\mathcal{P}_{ij\ldots k}) = -\sum_{ij\ldots k} \frac{q_{ij\ldots k\curvearrowleft}}{p_{ij\ldots k\circlearrowright}} \log \frac{q_{ij\ldots k\curvearrowleft}}{p_{ij\ldots k\circlearrowright}} - \sum_{ij\ldots k}\sum_{\alpha\in ij\ldots k} \frac{p_\alpha}{p_{ij\ldots k\circlearrowright}} \log \frac{p_\alpha}{p_{ij\ldots k\circlearrowright}}. \tag{5.25}$$

As before, p_α is the probability that the random walker is at the node α.

Note that the hierarchical extension here is quite natural, because it is not an algorithmic extension, but the extension of the quality function itself. In Sec. 5.3.4, we will show that the hierarchical map equation does not only give the modules of the different scales, but also makes the resolution of the modules better. The readers might have an impression that the hierarchical map equation is just analogous to the original map equation, and hence the two-level method is enough for most of the time. It is not true, however. One should always use the hierarchical map equation whenever it is possible because it simply gives the result with better quality.

5.2.4 Other extensions and some features of the map equation

There are some other extensions of the method of the map equation. Kim and Jeong [80] built a link-community version of the map equation. That is, the modules are assigned for links instead of nodes, so that the nodes can belong to multiple modules. The map equation with overlapping modules was considered by Esquivel and Rosvall [48] as well. The time evolution of the modules was considered in Ref. [132] and the memory effect of the random walker was took into account in Ref. [134]. Other than the hierarchical map equation by Rosvall and Bergstrom [133], a generalization in order to obtain the different module size was done by Schaub *et al.* [136] as well. Recently, Lambiotte and Rosvall [84] discussed the modification of the teleportation process for the directed network and its effect to the map equation.

The method of the map equation is outstanding in many ways. Because it is a flow method, it naturally takes account of the directedness of the network; some methods are formulated for undirected networks first and the extension for the directed networks is considered later, and thus it is debatable whether the extension is natural [89]. The idea of the map equation is rather simple and is numerically efficient, so that it is tractable in the analysis of large networks. It was also proved in the benchmark test by Lancichinetti *et al.* [85] that the map equation has high detectability, *i.e.*, it can resolve fuzzy modules well. Many classical methods of the community detection require the number of modules as an input, even though it is not known *a priori* in many cases. The map equation does not require such an input; the algorithm automatically determines the optimum number of modules in the process of minimization of the value of the description length. Indeed, this is the heart of the present thesis. The fact that the method does not require the number of modules as an input means that there exists an intrinsic scale that the quality function possesses and the modules are detected according to that scale.

5.3 Resolution limit of the map equation

As we explained in the last chapter, the modularity has the resolution limit of the scale $\mathcal{O}(\sqrt{K})$, where K is the total degree of the network. This remark is very important because it says that, even if one wants to search for the modules with the size less than $\mathcal{O}(\sqrt{K})$, it is impossible no matter how one tunes the optimization algorithm.

The fact that the modularity has a resolution limit is rather obvious. The modularity does not require the number of modules as an input, and hence an algorithm automatically determines the size and the number of modules from the information of the adjacency matrix according to the intrinsic scale that the modularity has. Imagine that one wants to partition the world map into modules. It makes sense that it should be partitioned into countries, rather than partitioned into cities or villages. In contrast, if a method has a very fine resolution limit, it implies that even if one wants to see the structure of the scale which is larger than the resolution limit, the modules of a such scale will be broken down if they contain modular structures inside. Such a problem can usually be solved by adopting a hierarchical algorithm or a hierarchical extension of the quality function.

The map equation does not require the number of modules as an input, and therefore, it does have its own resolution limit. The resolution limit of the map equation had been discussed in the literature and is empirically known that it has a very fine resolution limit. Nevertheless, its analytical expression had been missing. It is important to evaluate it analytically and see what exactly determines it; we know that the resolution limit is ruled by a global quantity (or quantities) of a network, but it is not obvious whether it is the total degree, the number of nodes, the number of module-connecting links, *i.e.* the cut size, or something else.

It seems difficult to derive the resolution limit of the map equation in a very general case including directed networks; for a directed network without the ergodicity, one needs to introduce teleportation, and hence the stationary distribution of a random surfer depends on the teleportation rate μ. Moreover, as we mentioned in Sec. 5.1, the preference vector $\mathbf{v}$ of the teleportation process does not need to be uniform, *i.e.*, the resolution limit depends on the details of the teleportation. For this reason, we restrict ourselves to the case of undirected networks. We also treat the unweighted networks for simplicity. The extension to the weighted network is trivial; we just replace the number of links to the

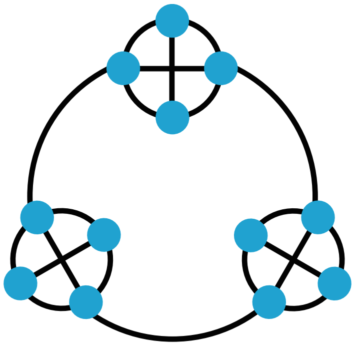


Figure 5.2: Detectable region of a module of the complete graph with n nodes in a ring of such modules. The ring consists of m modules. The figure shows the case of $n = 4$ and $m = 3$.

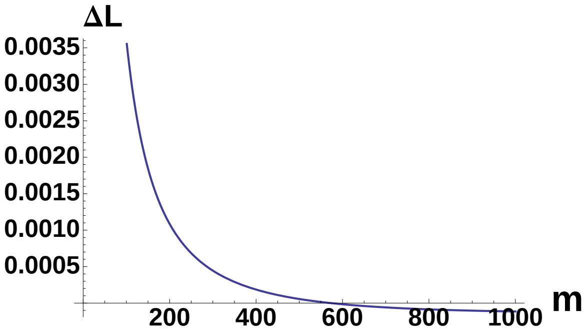


Figure 5.3: The value of the change in the description length ΔL when we merged two of the modules, as a function of the number of modules m.

sum of weights.

Before we tackle on the resolution limit formally, let us observe how the map equation behaves in a simple example. We consider a ring of m modules, each consists of a complete graph of n nodes as shown in Fig. 5.2. Figure 5.3 shows the change in the description length ΔL when we merged the two of the complete graphs as a single module, as a function of the number of modules m. Indeed, the sign of ΔL changes at a certain value of m; such a critical point corresponds to the resolution limit of the map equation. We will come back to this example after the argument of a general case.

5.3.1 Value of the description length in terms of the graph quantities

The description length $L(\mathbf{M})$ is written in terms of the Shannon entropies. It consists of the probabilities of the random walker such as the stationary distribution of the existence probability in a module π_i and the exiting probability $q_{i\curvearrowright}$ from the module i. In the case of the undirected networks, however, we can write down those quantities in terms of the number of links as follows. First, according to Eq. (5.3), the existence probability in a module π_i reads

$$\pi_i = \sum_{\alpha \in i} \frac{k_\alpha}{K} = \frac{2l_i + l_i^{\text{out}}}{K}, \tag{5.26}$$

where l_i is the number of links in the module i and l_i^{out} is the number of links which connect the nodes in the module i and the nodes in the other modules. The exiting probability $q_{i\curvearrowright}$ is the joint probability that the random walker is in the module i before the transition and is out of it after the transition. Since we have $T_{\beta\alpha} = 1/k_\alpha$ (where β is a neighbor of α) for the transition probability, we have

$$q_{i\curvearrowright} = \sum_{\beta \notin i} \sum_{\alpha \in i} T_{\beta\alpha} p_\alpha = \sum_{\beta \notin i} \sum_{\alpha \in i} \frac{1}{k_\alpha} \frac{k_\alpha}{K} = \frac{l_i^{\mathrm{out}}}{K}. \tag{5.27}$$

The sum over $\beta \notin i$ counts the number of links which are connected to the nodes outside of the module i, l_i^{out}. Substituting them into the original expression of the description length, Eq. (5.13), we have

$$\begin{aligned} L(\mathbf{M}) &= \frac{\left(\sum_{i=1}^m l_i^{\mathrm{out}}\right)}{K} \log \frac{\left(\sum_{i=1}^m l_i^{\mathrm{out}}\right)}{K} - 2 \sum_{i=1}^m \frac{l_i^{\mathrm{out}}}{K} \log \frac{l_i^{\mathrm{out}}}{K} \\ &\qquad + \sum_{i=1}^m \frac{2(l_i + l_i^{\mathrm{out}})}{K} \log \frac{2(l_i + l_i^{\mathrm{out}})}{K} - \sum_\alpha \frac{k_\alpha}{K} \log \frac{k_\alpha}{K} \\ &= \frac{1}{K} \left[\left(\sum_{i=1}^m l_i^{\mathrm{out}} \right) \log \left(\sum_{i=1}^m l_i^{\mathrm{out}} \right) - 2 \sum_{i=1}^m l_i^{\mathrm{out}} \log l_i^{\mathrm{out}} \right. \\ &\qquad \left. + 2 \sum_{i=1}^m (l_i + l_i^{\mathrm{out}}) \log 2(l_i + l_i^{\mathrm{out}}) - \sum_\alpha k_\alpha \log k_\alpha \right] \\ &= \frac{1}{K} \left[2C \log 2C + 2 \sum_{i=1}^m \mathcal{L}_i + K + 2C - \sum_\alpha k_\alpha \log k_\alpha \right]. \end{aligned} \tag{5.28}$$

Here, the sum $\sum_{i=1}^m l_i^{\mathrm{out}}$ corresponds to the twice of the quantity called the cut size C in the spectral clustering and we denote the local quantities of a module as $\mathcal{L}_i$, *i.e.*,

$$\mathcal{L}_i = -l_i^{\mathrm{out}} \log l_i^{\mathrm{out}} + (l_i + l_i^{\mathrm{out}}) \log(l_i + l_i^{\mathrm{out}}). \tag{5.29}$$

From Eq. (5.28), we readily see that there is no resolution limit caused by the total degree K; when we consider the difference of the description length $\Delta L(\mathbf{M})$ caused by a minimization step of $L(\mathbf{M})$, the value of K never changes its sign. The cut size C is, however, a global quantity of a network, and thus cause the resolution limit. In other words, as long as the minimization procedure does not change the cut size C, the map equation does not receive restrictions by global quantities.

Let us now consider a local update for minimization such that the cut size is decreased by δ, where $C \gg \delta > 0$. We let the partition before the update be $\mathbf{A}$ and let the partition after the update be $\mathbf{B}$. Expanding the difference of the description length $\Delta L(\mathbf{M}) = L(\mathbf{B}) - L(\mathbf{A})$ with respect to δ up to the first order, we obtain

$$\begin{aligned} \Delta L(\mathbf{M}) &= \frac{1}{K} \left[2\left(C - \delta\right) \log 2\left(C - \delta\right) - 2C \log 2C + 2R - 2\delta \right] \\ &\simeq \frac{2}{K} \left[-\delta \left(2 + \log(\mathrm{e}\, C) \right) + R \right], \end{aligned} \tag{5.30}$$

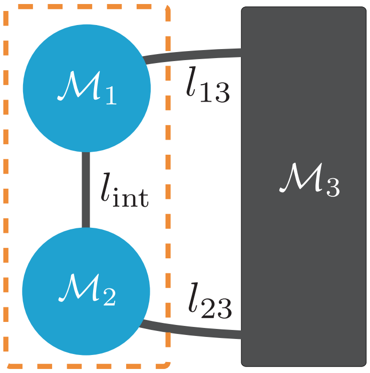


Figure 5.4: A schematic picture for the greedy update of two modules $\mathcal{M}_1$ and $\mathcal{M}_2$. Note that $\mathcal{M}_3$ may consist of many modules.

where e is the basis of natural logarithm and we defined

$$R \equiv \sum_{i'} \mathcal{L}_{i'}(\mathbf{B}) - \sum_{i} \mathcal{L}_{i}(\mathbf{A}). \tag{5.31}$$

The specific value of R depends on the type of update that we consider. From (5.30), we see that any local update would be accepted whenever the following condition is satisfied:

$$R \lesssim \delta \left(2 + \log(\mathrm{e}\, C)\right). \tag{5.32}$$

For the updates which increase the cut size, *i.e.* $\delta < 0$, on the other hand, no local update would be accepted whenever the cut size C is sufficiently large. In general, the modification of small modules gives a small value of R and thus it tends to be affected by the global structure. Depending on the balance of the size of modules to be evaluated and the links around them, however, large modules may be affected by the global structure as well. We will see such an example in Appendix B, where we discuss the update called fine tuning and coarse tuning.

5.3.2 Estimation of the resolution limit

We apply the equation for a general update (5.32) in order to analyze the resolution limit of the map equation. As was done for the discussion of the modularity [54] in the last chapter, let us compare the description length for the partition in which small modules are resolved and the one in which they are not resolved. As shown in Fig. 5.3.2, we denote the modules to be evaluated by $\mathcal{M}_1$ and $\mathcal{M}_2$, which have l_{int} inter-connecting links between them, the merged module of those by $\mathcal{M}_{12}$, and the rest of the network as $\mathcal{M}_3$. Note that $\mathcal{M}_3$ may consist of many modules and thus Fig. 5.3.2 represents a completely general situation. We consider an update from the partition **A** in which $\mathcal{M}_1$ and $\mathcal{M}_2$ are separated to the partition **B** in which $\mathcal{M}_1$ and $\mathcal{M}_2$ are merged. Then we have

$$
\begin{aligned}
R = &-l_3^{\text{out}} \log l_3^{\text{out}} + l_2^{\text{out}} \log l_2^{\text{out}} + l_1^{\text{out}} \log l_1^{\text{out}} \\
&+ (l_1 + l_1^{\text{out}} + l_2 + l_2^{\text{out}} - l_{\text{int}}) \log(l_1 + l_1^{\text{out}} + l_2 + l_2^{\text{out}} - l_{\text{int}}) \\
&- (l_2 + l_2^{\text{out}}) \log(l_2 + l_2^{\text{out}}) - (l_1 + l_1^{\text{out}}) \log(l_1 + l_1^{\text{out}}),
\end{aligned} \tag{5.33}
$$

where $l_3^{\text{out}} = l_{13} + l_{23}$, l_{13} is the number of links between $\mathcal{M}_1$ and $\mathcal{M}_3$, and l_{23} is the number of links between $\mathcal{M}_2$ and $\mathcal{M}_3$, respectively.

Here we consider the extreme case in which $l_{\text{int}} = 1$, namely $\delta = 1$, and set the sizes of two modules equal, namely $l_1 = l_2 = l_c$, because it gives the greatest value of R for the same module size. We can readily see it from (5.33); if we set $l_2 = \xi l_1 (\xi < 1)$, R is always less than in the case of $\xi = 1$ because the function $x \log x$ is super-linear. We also set $l_{13} = l_{23} = h$. Then, using the approximation that $l_c + h \gg 1$, we have

$$
R \simeq 1 + 2\left[l_c + (1+h)\log(1+h) - h \log h\right] - \log\left[\mathrm{e}(l_c + h)\right]. \tag{5.34}
$$

When $h = 1$, according to Eq. (5.32), we obtain the inequality for the resolution limit

$$
\frac{2^{2l_c + \epsilon}}{l_c + 1} \lesssim C, \tag{5.35}
$$

where $\epsilon = 3 - 2\log \mathrm{e} \simeq 0.1146$. When h is fairly larger than unity, because $(1+h)\log(1+h) - h\log h \simeq \log[\mathrm{e}(1+h)]$, we have

$$
\frac{(1+h)^2 2^{2l_c - 1}}{l_c + h} \lesssim C. \tag{5.36}
$$

Therefore, the map equation cannot detect the module with less than l_c links whenever the cut size C satisfies the above conditions. This limit is much lower than the resolution limit of the modularity [54], *i.e.*, the intrinsic scale of the map equation is much smaller than that of the modularity. That is, whether it is good or bad, the map equation detects smaller modules if they exist. Note that Eqs. (5.35) and (5.36) are the resolution limits only when we evaluate it around the global minimum of the description length; otherwise, because the value of the cut size C may change during updates, they are only practical restrictions during an optimization process.

The following two examples are worth mentioning. Again, we let $l_1 = l_2 = l$ and $l_{13} = l_{23} = h$ in Fig. 5.3.2. First, when $l_{\text{int}} = 0$, we have $\Delta L(\mathbf{M}) = 4l/K > 0$; thus, the modules without direct links never get merged as it should. Second, we let $\mathcal{M}_3$ be a single module. Calculating ΔL according to Eq. (5.28), we obtain $\Delta L(h=1) > 0$ for $l \geq 2$, $\partial \Delta L / \partial h > 0$ and $\Delta L(h \to \infty) = 4(l-1)/K$ for any l. Therefore, the map equation can detect modules of arbitrary size with $l \geq 2$ in such a case.

5.3.3 Illustrations of the resolution limit

As an illustration, we again consider the ring of modules that we introduced at the beginning of Sec. 5.3; the ring has m modules and each module consists of a complete graph of n nodes. The elements of the description length $L(\mathbf{M})$ are

$$K = 2m\left(\frac{n(n-1)}{2} + 1\right), \tag{5.37}$$

$$q^{\mathbf{A}}_{\curvearrowright} = \frac{2m}{K}, \qquad q^{\mathbf{B}}_{\curvearrowright} = \frac{2(m-1)}{K}, \tag{5.38}$$

$$q_{1\curvearrowright} = q_{2\curvearrowright} = q_{12\curvearrowright} = \frac{2}{K}, \tag{5.39}$$

$$q_{1\circlearrowright} = q_{2\circlearrowright} = \frac{n(n-1)+4}{K}, \tag{5.40}$$

$$q_{12\circlearrowright} = \frac{2n(n-1)+6}{K}. \tag{5.41}$$

For this graph, we have $C = m$. The plot in Fig. 5.5 shows an excellent agreement between the numerical result and the approximated one above; points are the exact value of m at which $\Delta L(\mathbf{M}) = 0$ for a given n and the solid line shows the curve of Eq. (5.35).

The resolution limit of the modularity (dashed line) is also shown in Fig. 5.5. The modules with the internal links l_i less than $\sqrt{L/2}$ cannot be resolved, *i.e.*, in terms of n and m,

$$\frac{n(n-1)}{2} < \left[\frac{m}{2}\left(\frac{n(n-1)}{2} + 1\right)\right]^{1/2}, \tag{5.42}$$

$$\frac{n^2(n-1)^2}{2+n(n-1)} < m. \tag{5.43}$$

The resolution limit of the modularity is much below that of the map equation.

5.3.4 How to eliminate the resolution limit

Although we observed that the resolution of the map equation is extremely small typically, it does exist. The next natural question is whether we can eliminate the resolution limit and how we can achieve it. Although it is an intrinsic scale that the quality function has, it is possible to control by altering the quality function itself. Both the modularity and the map equation have a factor which favors the large modules and a factor which favors the small modules. If we modify the balance of their competition, we can make the quality function have higher resolution, *i.e.*, make it favor the partition with smaller modules. Such an adjustment for the modularity was proposed by Reichardt and Bornholdt [124] before the original paper of the resolution limit by Fortunato and Barthélemy [54]. Such an approach has a problem, however, that we usually do not know the value of γ in Eq. (4.33) to choose *a priori*. As an alternative way to overcome the resolution limit, there is a method proposed by Berry *et al.* [29], in which they consider re-weighting of the links between the modules.

In the case of the map equation, we can naturally eliminate the resolution limit without introducing an extra parameter. We here propose to use the hierarchical map equation.

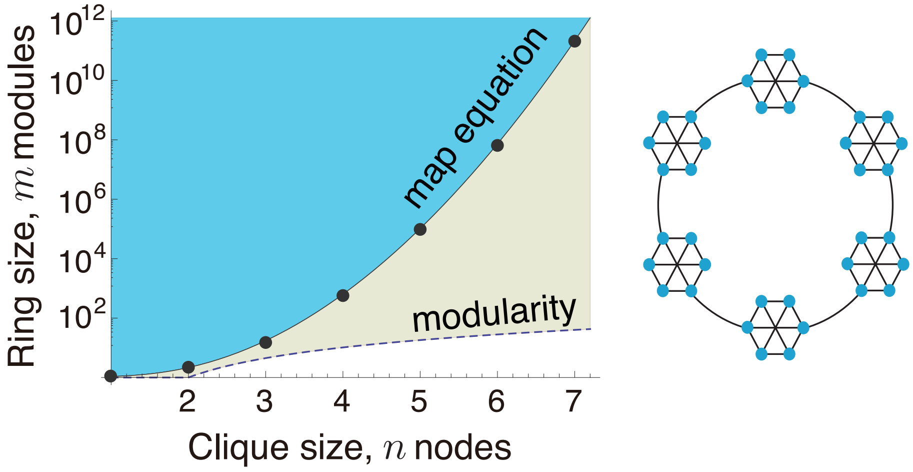


Figure 5.5: Detectable region of a module of the complete graph with n nodes in a ring of m modules. The right figure is the case of $n = 6$ and $m = 6$. Each point shows the numerically exact value of m which gives $\Delta L(\mathbf{M}) = 0$ for a given n. The solid and the dashed lines show the resolution limits of the map equation (Eq. (5.35)) and the modularity, above which the module with n nodes are not resolved, respectively.

As we did for the two-level method, we can write down (5.22) in the following form:

$$\begin{aligned} L(\mathbf{M}) = {} & q_{\curvearrowright} \log q_{\curvearrowright} + \sum_{i_1} q_{i_1 \circlearrowright} \log q_{i_1 \circlearrowright} + \sum_{i_1 i_2} q_{i_1 i_2 \circlearrowright} \log q_{i_1 i_2 \circlearrowright} + \cdots \\ & + \sum_{i_1 i_2 \ldots i_k} q_{i_1 i_2 \ldots i_k \circlearrowright} \log q_{i_1 i_2 \ldots i_k \circlearrowright} - \sum_{\alpha} p_\alpha \log p_\alpha \\ & - 2 \left(\sum_{i_1} q_{i_1 \curvearrowright} \log q_{i_1 \curvearrowright} + \sum_{i_1 i_2} q_{i_1 i_2 \curvearrowright} \log q_{i_1 i_2 \curvearrowright} + \cdots + \sum_{i_1 i_2 \ldots i_k} q_{i_1 i_2 \ldots i_k \curvearrowright} \log q_{i_1 i_2 \ldots i_k \curvearrowright} \right). \end{aligned} \tag{5.44}$$

In the case where the balance condition of the probability flow is not satisfied, we needed to distinguish the entering probability and the exiting probability [133]. Note, however, that they are equal in undirected networks, and hence we use a single notation. Let us then consider the difference of the description length by an update of minimization process. The modification of the partition at a certain level does not affect the partitions in higher and lower levels, while it alters the normalization factor for the probabilities of the movements between submodules at one level below. Therefore, the difference of the description length for the alteration in the xth level is

$$\begin{aligned}\Delta L(\mathbf{M}) = {} & q^{\mathbf{B}}_{i_1 i_2 \ldots i_{x-1} \circlearrowright} \log q^{\mathbf{B}}_{i_1 i_2 \ldots i_{x-1} \circlearrowright} - q^{\mathbf{A}}_{i_1 i_2 \ldots i_{x-1} \circlearrowright} \log q^{\mathbf{A}}_{i_1 i_2 \ldots i_{x-1} \circlearrowright} \\ & - 2 \left(\sum_{i'_x} q^{\mathbf{B}}_{i_1 i_2 \ldots i'_x \curvearrowright} \log q^{\mathbf{B}}_{i_1 i_2 \ldots i'_x \curvearrowright} - \sum_{i_x} q^{\mathbf{A}}_{i_1 i_2 \ldots i_x \curvearrowright} \log q^{\mathbf{A}}_{i_1 i_2 \ldots i_x \curvearrowright} \right) \\ & + \left(\sum_{i'_x} q^{\mathbf{B}}_{i_1 i_2 \ldots i'_x \circlearrowright} \log q^{\mathbf{B}}_{i_1 i_2 \ldots i'_x \circlearrowright} - \sum_{i_x} q^{\mathbf{A}}_{i_1 i_2 \ldots i_x \circlearrowright} \log q^{\mathbf{A}}_{i_1 i_2 \ldots i_x \circlearrowright} \right), \end{aligned} \tag{5.45}$$

which is analogous to that of two-level method,

$$\begin{aligned}\Delta L_{\text{two-level}}(\mathbf{M}) = {} & q^{\mathbf{B}}_{\curvearrowright} \log q^{\mathbf{B}}_{\curvearrowright} - q^{\mathbf{A}}_{\curvearrowright} \log q^{\mathbf{A}}_{\curvearrowright} - 2 \left(\sum_{i'=1}^{m'} q^{\mathbf{B}}_{i' \curvearrowright} \log q^{\mathbf{B}}_{i' \curvearrowright} - \sum_{i=1}^{m} q^{\mathbf{A}}_{i \curvearrowright} \log q^{\mathbf{A}}_{i \curvearrowright} \right) \\ & + \left(\sum_{i'=1}^{m'} p^{\mathbf{B}}_{i' \circlearrowright} \log p^{\mathbf{B}}_{i' \circlearrowright} - \sum_{i=1}^{m} p^{\mathbf{A}}_{i \circlearrowright} \log p^{\mathbf{A}}_{i \circlearrowright} \right). \end{aligned} \tag{5.46}$$

Instead of $q_{\curvearrowright}$ in the two-level method in the above equation, we have

$$q_{i_1 i_2 \ldots i_{x-1} \circlearrowright} = q_{i_1 i_2 \ldots i_{x-1} \curvearrowright} + \sum_{i_x} q_{i_1 i_2 \ldots i_x \curvearrowright} \tag{5.47}$$

for the multi-level method; hence, other than the extra term $q_{i_1 i_2 \ldots i_{x-1} \curvearrowright}$ in $q_{i_1 i_2 \ldots i_{x-1} \circlearrowright}$, the mathematical structure of the hierarchical map equation is analogous to that of the two-level method.

The resolution limit still exists for the hierarchical map equation in principle. There is, however, a big quantitative difference for the detection of modules in a supermodule. When we modify the partition of modules in a supermodule, the nonlocal part of the difference of the description length is a function of the links inside of the supermodule. Thus, the limitation comes from the structure of the supermodule, not from the whole network. For this reason, even when we do not explicitly use the higher levels of the hierarchy, we may obtain a higher resolution with the hierarchical map equation.

5.3.5 Illustration of the elimination of the resolution limit by the hierarchical map equation

A good example that demonstrates this fact is the graph of the Sierpinski triangles as shown in the inset of Fig. 5.6. We consider the Sierpinski triangles of many sizes by changing its depth of hierarchy. Using the code distributed at [8], we detected the modules of those graphs with the two-level method and the multi-level method, respectively. For the multi-level method, we focus on the results in the finest level, where every module should be a triangle of three nodes; the cut size C that each graph is supposed to have is known.

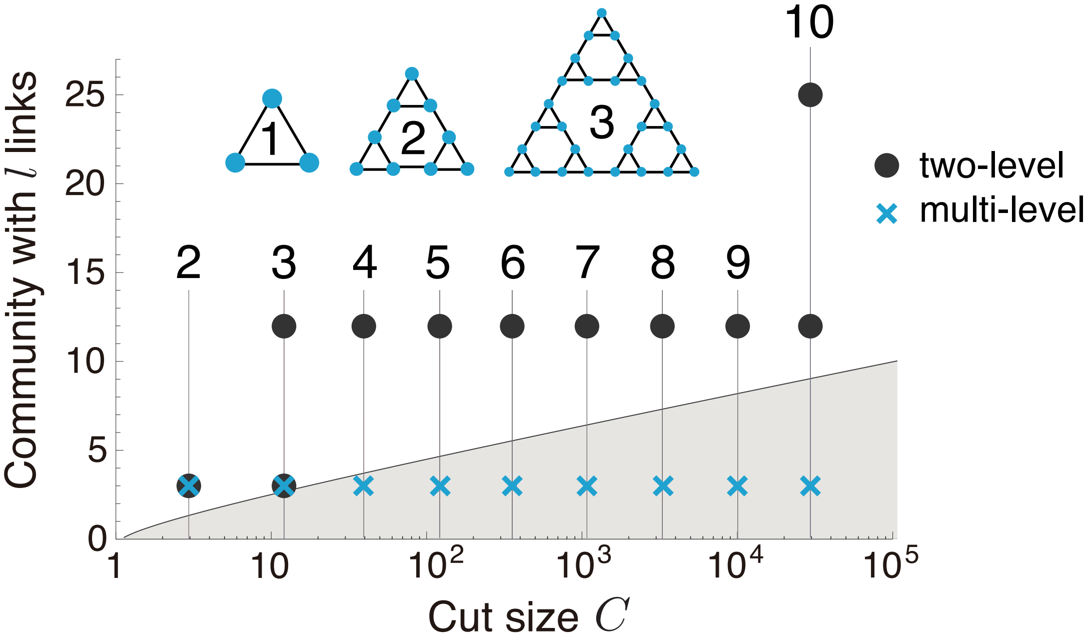


Figure 5.6: Detected size of modules for each Sierpinski triangle of different size with the two-level method (circular points) and the multi-level method (cross points). The Sierpinski triangles up to three hierarchies are shown at the top. If we partition the graph into the modules of three nodes, each graph has a unique value of the cut size C; for example, the Sierpinski triangle with three hierarchies has $C = 12$. The boundary of the shaded region shows the resolution limit of the module size which is calculated by Eq. (5.35).

The points in the plot of Fig. 5.6 show the size of the detected modules l in each graph. The curve is the resolution limit of the two-level method, Eq. (5.35). While the result of the two-level method (circular points) is harmed by the resolution limit, the multi-level method (cross points) detects the correct modules in any network size. Even in the case where the algorithm does not reach the global minimum, the small modules may be detected because the cut size C changes during the updates.

The effect of the resolution limit as well as the difference between the two-level method and the multi-level one can be seen in the partitioning of real networks as well. Figure 5.7 shows the module size distribution of the rating network in Amazon.com [9, 73, 98, 108]. Although the size of a module here means the number of nodes, not the number of links within a module, the multi-level method detects small modules much more than the two-level method does. This is not trivial, because the multi-level solution is not constructed as a simple decomposition of the two-level solution, but by an attempt to find a global optimal modular description. It implies that the multi-level method detects the modules which could not be detected by the two-level method because of the resolution limit.

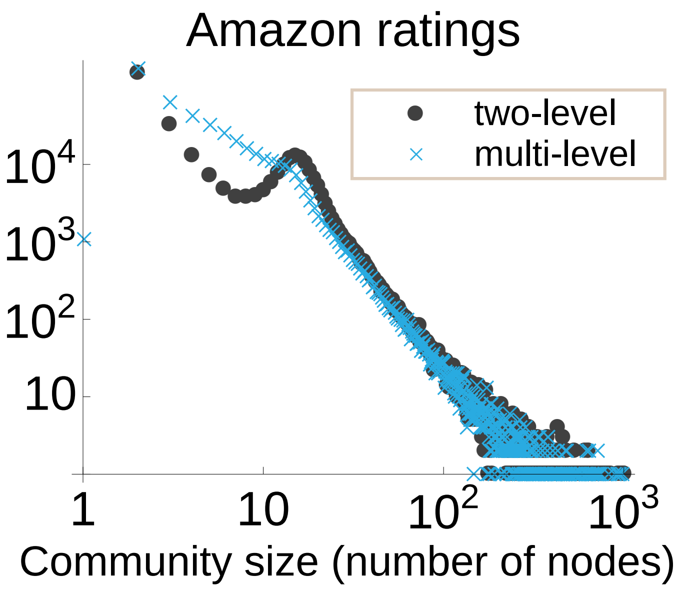


Figure 5.7: Module size distributions of the rating network in Amazon.com (bipartite network). The circular points represent the result of the two-level method and the cross points represent the result of the multi-level method.

The result of Fig. 5.7 is consistent with Eq. (5.35). In this network, the total number of modules found by the multi-level method is $483,657$ and the total number of links is $5,743,258$. Since the value of cut size is bounded below by the number of modules (minus one) and above by the total number of links in the network, Eq. (5.35) predicts $l_c \approx 12$ (see Table 5.2). Although the histogram in Fig. 5.7 is plotted with respect to the number of nodes n in a module, $n \approx 12$ is about where the results of the two-level method and the multi-level method deviate from each other. Note that the number of nodes n within a module is bounded by the number of links l within the module by $n \leq l + 1$. The equality holds for a module of a tree. Therefore, the results of the two-level method and the multi-level method should deviate from each other by about $n \sim l$.

The exponent of the community size distribution is changed suddenly below the point where the deviation occurs; this sudden change seems to happen because the multi-level solution typically has a three-level structure below the size where the deviation occurs and has a two-level structure above it.

In addition to the rating network of Amazon.com, we list the results for four more real networks with the distributed code of Infomap [8]. Figure 5.8 shows the module size distributions of (a) the citation network of publications in the arXiv's High Energy Physics – Theory (hep-th) section (direction is ignored) [91], (b) the road network of California in the U. S. A. [94], (c) the collaboration graph of authors of scientific papers from DBLP computer science bibliography [95], (d) the friendship network of Facebook [147], (e) the membership network of the software development hosting site GitHub (bipartite

Table 5.1: Data of real networks.

	# of nodes	# of links	# of communities (two-level)	# of communities (multi-level)
Amazon	3,376,972	5,743,258	349,016	483,657
arXiv_hep-th	48,239	352,807	1,355	2,211
California	1,965,206	2,766,607	82,322	344,485
Facebook	63,731	817,090	2,268	2,819
Pennsylvania	1,088,092	1,541,898	46,899	189,109
DBLP	1,103,412	4,225,686	82,494	118,868
GitHub	177,386	440,237	24,855	30,386

network) [36], and (f) the road network of Pennsylvania in the U. S. A. [92] in the log-log scale. We used the data distributed at [9]. The size of each network and the total number of detected modules are listed in Table 5.1. Note that the restriction of the two-level method affects the resolution of modules gradually in real networks, so that we cannot always expect a clear branch from the plot of the multi-level method, as we observed in the Amazon network. Nonetheless, the multi-level method always detects small modules more often than the two-level method does in all cases.

Table 5.2: Table of the left-hand side of Eq. (5.35).

l_c	$2^{2l_c+\epsilon}/(l_c+1)$ (approx.)
1	2.2
2	5.8
3	17.3
4	55.4
5	184.8
6	633.5
7	2,217.3
8	7,883.9
9	28,381.9
10	103,206.8
11	378,424.9
12	1,397,261.1
13	5,189,826.9
14	19,375,353.9
15	72,657,576.9
16	273,534,407.3

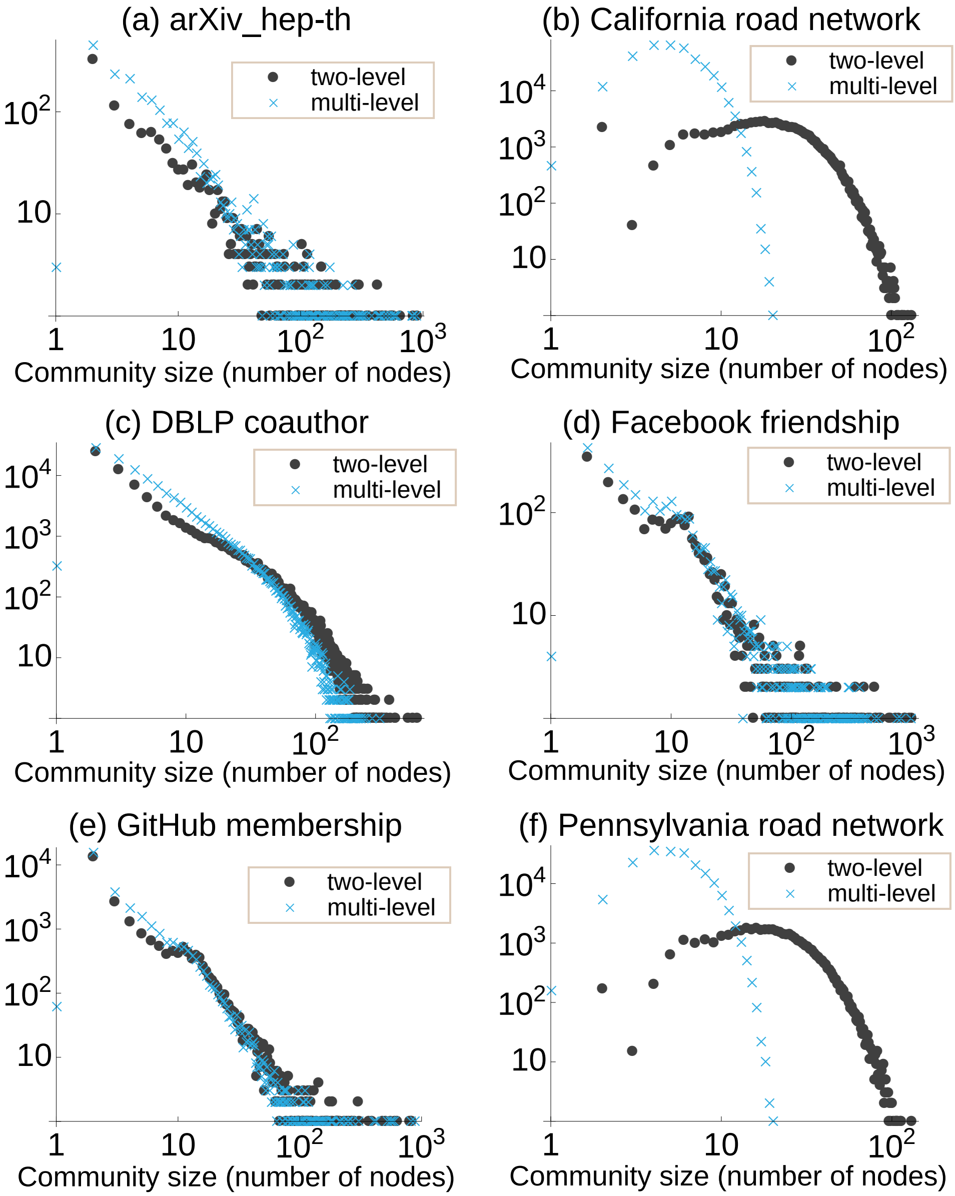


Figure 5.8: The community size distributions of (a) the citation network of publications in the arXiv's High Energy Physics – Theory (hep-th) section (direction is ignored), (b) the road network of California in the U. S. A., (c) the collaboration graph of authors of scientific papers from DBLP computer science bibliography, (d) the friendship network of Facebook, (e) the membership network of the software development hosting site GitHub (bipartite network), and (f) the road network of Pennsylvania in the U. S. A. in the log-log scale. The circular points represent the result of the two-level method and the cross points represent the result in the finest level of the multi-level method. Note that the size of a module here does not mean the number of links, but the number of nodes within the module.

Chapter 6

Conclusion

In the present thesis, we discussed the information diffusion and the community detection of complex networks. In both topics, the underlying frameworks are the stochastic processes.

In the first half, we focused on the information diffusion in complex networks, especially in online social networks. The diffusion of information such as news or stories is ubiquitous in online social networks thanks to the functions which encourage the spreading, such as retweet in Twitter. We proposed a model to describe it (Sec. 3.2) and analytically analyzed the behavior of diffusion predicted by the model. Although the Galton-Watson branching process is known as one the fundamental models for such a diffusion process, we found a behavior which cannot be described with it in the Twitter data which we collected.

In order to characterize the diffusion, we divided the users around the seed user in generations, namely the distance from the seed user, and measured retweet rate, *i.e.*, the ratio of users who retweet among the users in each generation. We defined the retweet rates as stochastic variables. While each retweet rate should obey a Poisson distribution in the Galton-Watson branching process, we observed a fat-tail distribution, which is well fitted by a lognormal distribution (Fig. 3.5). Although we are not yet sure its microscopic origin, our model phenomenologically takes account of this fact; by setting the retweet rate always equals to one, our model reduces to the Galton-Watson branching process.

Furthermore, based on the model that we proposed, we analyzed the possibility of the viral diffusion and the effect of the correlation between the retweet rates on it. In the case where the diffusion goes viral, it is natural to expect that the neighbors of the user who retweeted are likely to retweet as well, *i.e.* there exists a correlation between the generations. By assuming weak correlation of a form which can be treated analytically, we calculated how it alters the tipping point of the viral diffusion (Eq. (3.52)); although the tipping point of the viral diffusion seems extremely high in the case of the uncorrelated case, we found that the tipping point largely decreases due to the correlation (Fig. 3.9).

Investigating how our model can be described microscopically is an important future work. Moreover, it is interesting to see what information we can obtain from the higher moments of the statistics of the number of retweets.

In the second half, we discussed analytical properties of the map equation, which is one of the state-of-the-art methods of community detection. The map equation makes use of a random walker in a network. Although the quality function of the map equation seems difficult to handle analytically because it is formulated in terms of the Shannon entropies, for undirected networks, it turns out to be tractable if we write the entropies down in terms of the graph quantities.

We estimated the so-called resolution limit of the map equation (Eq. (5.35)), the lower bound of the module size which the method can detect. It is orders of magnitude lower (high resolution) than it is for the modularity in practice. More importantly, the resolution limit of the map equation is less restrictive, because it is determined by the number of links between modules, where the resolution limit of modularity is determined by the total degree of the network. Furthermore, we showed that it is possible to overcome the resolution limit naturally by using the hierarchical version of the map equation. We confirmed our results with some synthetic graphs and showed what we obtain for the real networks.

In this work, we thoroughly investigated features of the resolution limit for undirected unweighted networks. It is, however, still unknown what can be said in the case of a directed network; it has a crucial difference from the undirected case, because we may need to introduce a step called teleportation in order to make the random surfer to have a stationary state.

Methods with very high resolution limits may show poor performances in detecting large communities, because they tend to decompose a large module into smaller ones. Although the tendency seems more difficult to analyze in general, it is important to understand when such a decomposition occurs and how we can avoid it.

It is expected that the landscape of the quality function of the map equation can also be investigated based on a similar analysis as we did for the resolution limit. It will give us an insight in which conditions the optimization becomes really hard.

A better accuracy of detection methods enables us to observe the community structure even more precisely and we can understand more about the properties of the complex networks. Fortunately, there are many kinds of network data nowadays. The community detection is, however, not only about finding the densely connected components. What is essential is to detect structures of the network; the dense connection among nodes is just one aspect of them. There should be many structures hidden in the complex networks (some of them may be dynamical) and we may be able to detect them from the aspect which is totally different from the typical perspective of today.

Appendix A

Diagonalization of the covariance matrix

In this Appendix, we explain the diagonalization of the inverse of the covariance matrix Σ^{-1} in Sec. 3.7.2. In the following, we denote the matrix to diagonalize as A.

In order to obtain for the eigenvalues and eigenvectors of A of the form

$$A = \begin{bmatrix} \sigma^{-2} & C^{-1} & 0 & 0 & \cdots \\ C^{-1} & \sigma^{-2} & C^{-1} & 0 & \cdots \\ 0 & C^{-1} & \sigma^{-2} & C^{-1} & \cdots \\ 0 & 0 & C^{-1} & \sigma^{-2} & \ddots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix}, \tag{A.1}$$

we consider the following matrix B:

$$B = \delta_{i+1,j} + \delta_{i-1,j} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 1 & 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & 1 & \cdots \\ 0 & 0 & 1 & 0 & \ddots \\ \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix}, \qquad \textit{i.e.}, \;\; A = \sigma^{-2} I + C^{-1} B. \tag{A.2}$$

Since we can take B as a tight-binding model with the fixed ends, it is easy to imagine that the eigenvectors are given as $\vec{v}_n = (\sin(k_n), \sin(2k_n), \cdots, \sin(Nk_n))/Z$, where $k_n = \pi n/(N+1)$ and Z is the norm. Therefore, the unitary matrix which diagonalizes B would be

$$U_{mn} = \frac{1}{Z}\sin(mk_n), \qquad \left(k_n = \frac{\pi n}{N+1}\right) \tag{A.3}$$

$$Z^2 = \sum_{m=1}^{N} \sin^2(mk_n) = \frac{1}{2}\sum_{m=1}^{N}\left[1 - \cos\left(\frac{2\pi mn}{N+1}\right)\right] = \frac{N+1}{2}. \tag{A.4}$$

For the last equality, note that the sum from $m = 1$ to $m = N + 1$ reads

$$\sum_{m=1}^{N+1} \cos\left(\frac{2\pi mn}{N+1}\right) = 0, \tag{A.5}$$

and the sum that we consider here is from one to N, and thus for $n \neq 0$,

$$\sum_{m=1}^{N} \cos\left(\frac{2\pi mn}{N+1}\right) = -1. \tag{A.6}$$

Let us confirm that U actually diagonalizes B and solve for the eigenvectors. We have

$$\begin{aligned}
U_{\alpha\gamma} B_{\gamma\delta} U^{\dagger}_{\delta\beta} &= \frac{1}{Z^2} \sum_{\gamma,\delta} \sin(\alpha k_\gamma)(\delta_{\gamma+1,\delta} + \delta_{\gamma-1,\delta}) \sin(\beta k_\delta) \\
&= \frac{1}{Z^2} \sum_{\gamma} \sin(\alpha k_\gamma) \left[\sin\left(\frac{\pi\beta(\gamma+1)}{N+1}\right) + \sin\left(\frac{\pi\beta(\gamma-1)}{N+1}\right)\right] \\
&= \frac{1}{Z^2} \cos\left(\frac{\pi\beta}{N+1}\right) \sum_{\gamma} \sin\left(\frac{\pi\alpha\gamma}{N+1}\right) \sin\left(\frac{\pi\beta\gamma}{N+1}\right) \\
&= \frac{1}{Z^2} \cos\left(\frac{\pi\beta}{N+1}\right) \sum_{\gamma} \left[\cos\left(\frac{\pi\gamma(\alpha-\beta)}{N+1}\right) - \cos\left(\frac{\pi\gamma(\alpha+\beta)}{N+1}\right)\right].
\end{aligned} \tag{A.7}$$

For $\alpha \neq \beta$, we have zero. For $\alpha = \beta$, we have

$$\sum_{\gamma} \left[\cos\left(\frac{\pi\gamma(\alpha-\beta)}{N+1}\right) - \cos\left(\frac{\pi\gamma(\alpha+\beta)}{N+1}\right)\right] = N - \sum_{\gamma} \cos\left(\frac{2\pi\gamma\alpha}{N+1}\right) = N + 1. \tag{A.8}$$

Hence, it is confirmed that U actually diagonalizes B. The eigenvalue λ_α then reads

$$\lambda_\alpha = U_{\alpha\gamma} B_{\gamma\delta} U^{\dagger}_{\delta\alpha} = \frac{N+1}{Z^2} \cos\left(\frac{\pi\alpha}{N+1}\right) = 2\cos k_\alpha. \tag{A.9}$$

Appendix B

Fine tuning and coarse tuning in the map equation

The analysis which we carried out for the resolution limit of the map equation corresponds to the greedy update for the minimization procedure, *i.e.*, we consider whether the two selected modules should be merged or not. We can also analyze the restriction to another type of update by the global structure using the result for an arbitrary update. Although the greedy algorithm is commonly used to optimize a quality function in practice, as long as we execute the greedy algorithm, there is no chance to separate two modules once they are merged. In order to correct an implausible partitioning, there are updates called *fine tuning* and *coarse tuning* [133], which we analyze here. As shown in Fig. B.1, fine tuning evaluates the movement of a node from a module to another. Similarly, coarse tuning evaluates the movement of a module from a supermodule to another.

In this section, we discuss the property of fine tuning by evaluating R in Eq. (5.31) of the two-level method. Figures B.2(a) and B.2(b) show the general schematic pictures of modules for fine tuning. We consider the difference of the map equation due to the movement of the node $\mathcal{M}_*$. We refer to the partition before the movement as $\mathbf{A}$; we denote $\mathcal{M}_1 + \mathcal{M}_* =: \mathcal{M}_{\mathrm{I}}^{\mathbf{A}}$ and $\mathcal{M}_2 =: \mathcal{M}_{\mathrm{II}}^{\mathbf{A}}$. We refer to the partition after the movement as $\mathbf{B}$; we denote $\mathcal{M}_1 =: \mathcal{M}_{\mathrm{I}}^{\mathbf{B}}$ and $\mathcal{M}_* + \mathcal{M}_2 =: \mathcal{M}_{\mathrm{II}}^{\mathbf{B}}$. As we did in the main text, we denote the rest of the network by $\mathcal{M}_3$, which may consist of many modules. Recalling the evaluation of a general update, we have

$$R \lesssim \delta \left(2 + \log \mathrm{e}\, C\right) \tag{B.1}$$

with $\delta = -(l_{1*} - l_{2*})$ (note that δ is defined as the minus of the change in the cut size C)

and

$$
\begin{aligned}
R &= \mathcal{L}_{\mathrm{I}}^{\mathbf{B}} + \mathcal{L}_{\mathrm{II}}^{\mathbf{B}} - \mathcal{L}_{\mathrm{I}}^{\mathbf{A}} - \mathcal{L}_{\mathrm{II}}^{\mathbf{A}}, && \text{(B.2)}\\
\mathcal{L}_{\mathrm{I}}^{\mathbf{B}} &= -(l_{1*} + l_{12} + l_{13}) \log(l_{1*} + l_{12} + l_{13}) \\
&\qquad\qquad + (l_1 + l_{1*} + l_{12} + l_{13}) \log(l_1 + l_{1*} + l_{12} + l_{13}), && \text{(B.3)}\\
\mathcal{L}_{\mathrm{II}}^{\mathbf{B}} &= -(l_{1*} + l_{3*} + l_{12} + l_{23}) \log(l_{1*} + l_{3*} + l_{12} + l_{23}) \\
&\qquad\qquad + (l_2 + l_{*}^{\mathrm{out}} + l_{12} + l_{23}) \log(l_2 + l_{*}^{\mathrm{out}} + l_{12} + l_{23}), && \text{(B.4)}\\
\mathcal{L}_{\mathrm{I}}^{\mathbf{A}} &= -(l_{2*} + l_{3*} + l_{12} + l_{13}) \log(l_{2*} + l_{3*} + l_{12} + l_{13}) \\
&\qquad\qquad + (l_1 + l_{*}^{\mathrm{out}} + l_{12} + l_{13}) \log(l_1 + l_{*}^{\mathrm{out}} + l_{12} + l_{13}), && \text{(B.5)}\\
\mathcal{L}_{\mathrm{II}}^{\mathbf{A}} &= -(l_{2*} + l_{12} + l_{23}) \log(l_{2*} + l_{12} + l_{23}) \\
&\qquad\qquad + (l_2 + l_{2*} + l_{12} + l_{23}) \log(l_2 + l_{2*} + l_{12} + l_{23}). && \text{(B.6)}
\end{aligned}
$$

As shown in Figs. B.2(a) and B.2(b), $l_{xy}\,(x, y = \{1, 2, 3, *\})$ represents the number of links connecting $\mathcal{M}_x$ and $\mathcal{M}_y$ and $l_{*}^{\mathrm{out}} = l_{1*} + l_{2*} + l_{3*}$. A large value of $\sum_{i=1}^{m} l_i^{\mathrm{out}}$ encourages an update when $l_{2*} > l_{1*}$ ($\delta > 0$) and discourages an update when $l_{2*} < l_{1*}$ ($\delta < 0$). As an example, let us evaluate R by fixing $l_{12} = l_{3*} = 0$, $l_{13} = l_{23} = 1$, and changing l_{1*} and l_{2*}; see Figs. B.3(a) and B.3(b). Each plot in Fig. B.4 shows a contour plot of R for various values of l_1 and l_2 with a certain set of l_{1*} and l_{2*}. The update is accepted when the inequality (B.1) is satisfied, *i.e.*, the node $\mathcal{M}_*$ is moved from the module $\mathcal{M}_1$ to the module $\mathcal{M}_2$. The plots show that the map equation tries to equate the size of modules; even when the node $\mathcal{M}_*$ is more densely connected to the module $\mathcal{M}_2$ than to the module $\mathcal{M}_1$, *i.e.* $l_{2*} > l_{1*}$, the node $\mathcal{M}_*$ would not be moved to $\mathcal{M}_2$ if l_1 is not large enough compared to l_2 and vice versa. A similar discussion holds for coarse tuning. In summary, we analyzed how the map equation behaves under a tuning update, in addition to that of the greedy update which we analyzed for the resolution limit.

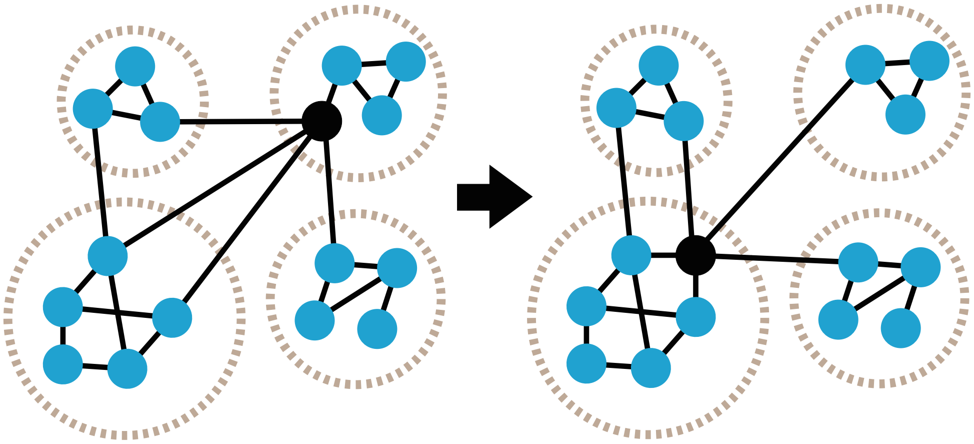


Figure B.1: An example of tuning update. Fine tuning considers the movement of a node from a module to another and coarse tuning considers the movement of a module from a supermodule to another.

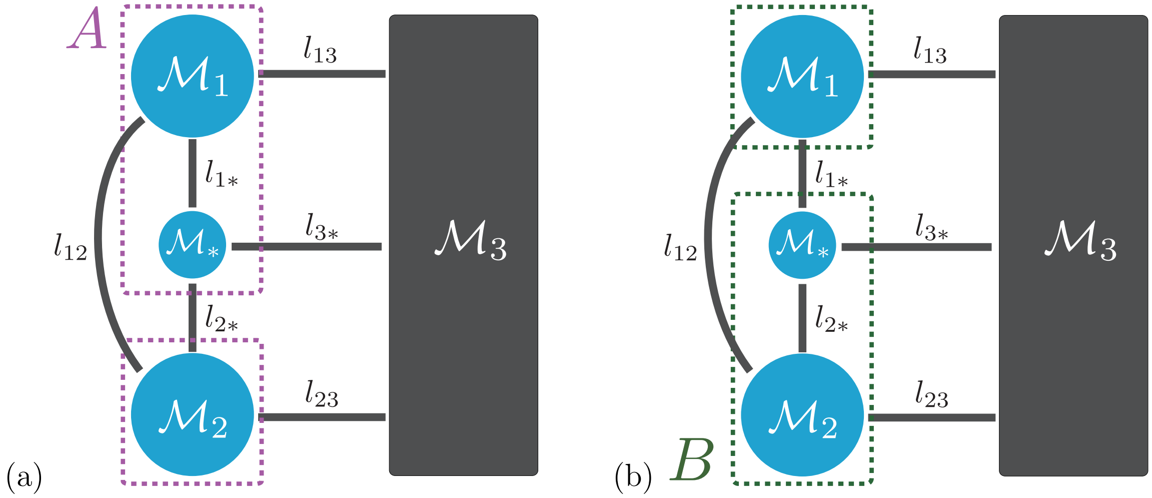


Figure B.2: General schematic picture for tuning; (a) represents the partition before the update and (b) represents the partition after the update.

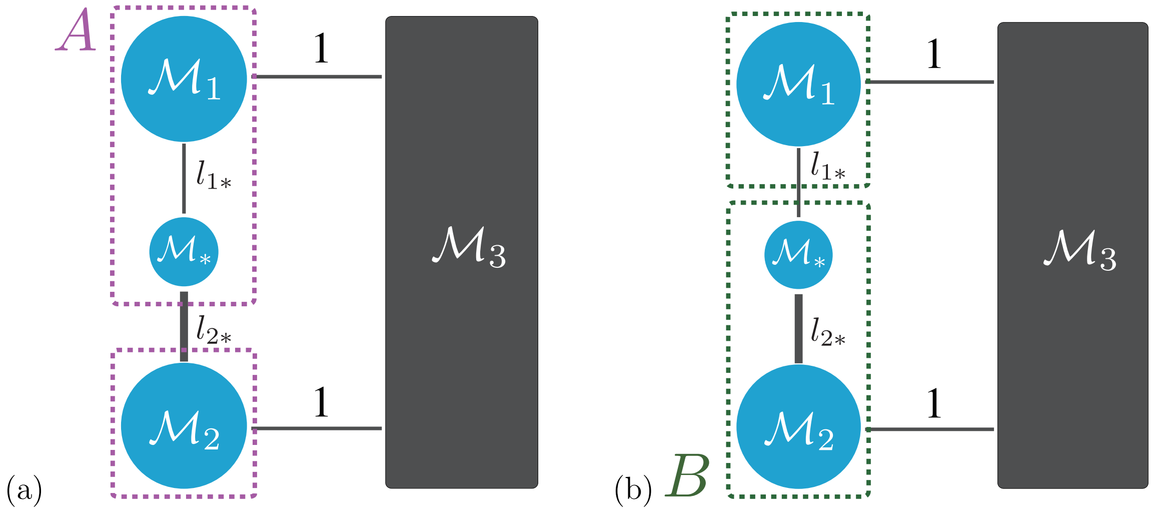


Figure B.3: A schematic picture of the specific fine tuning update that we analyze in Fig. B.4.

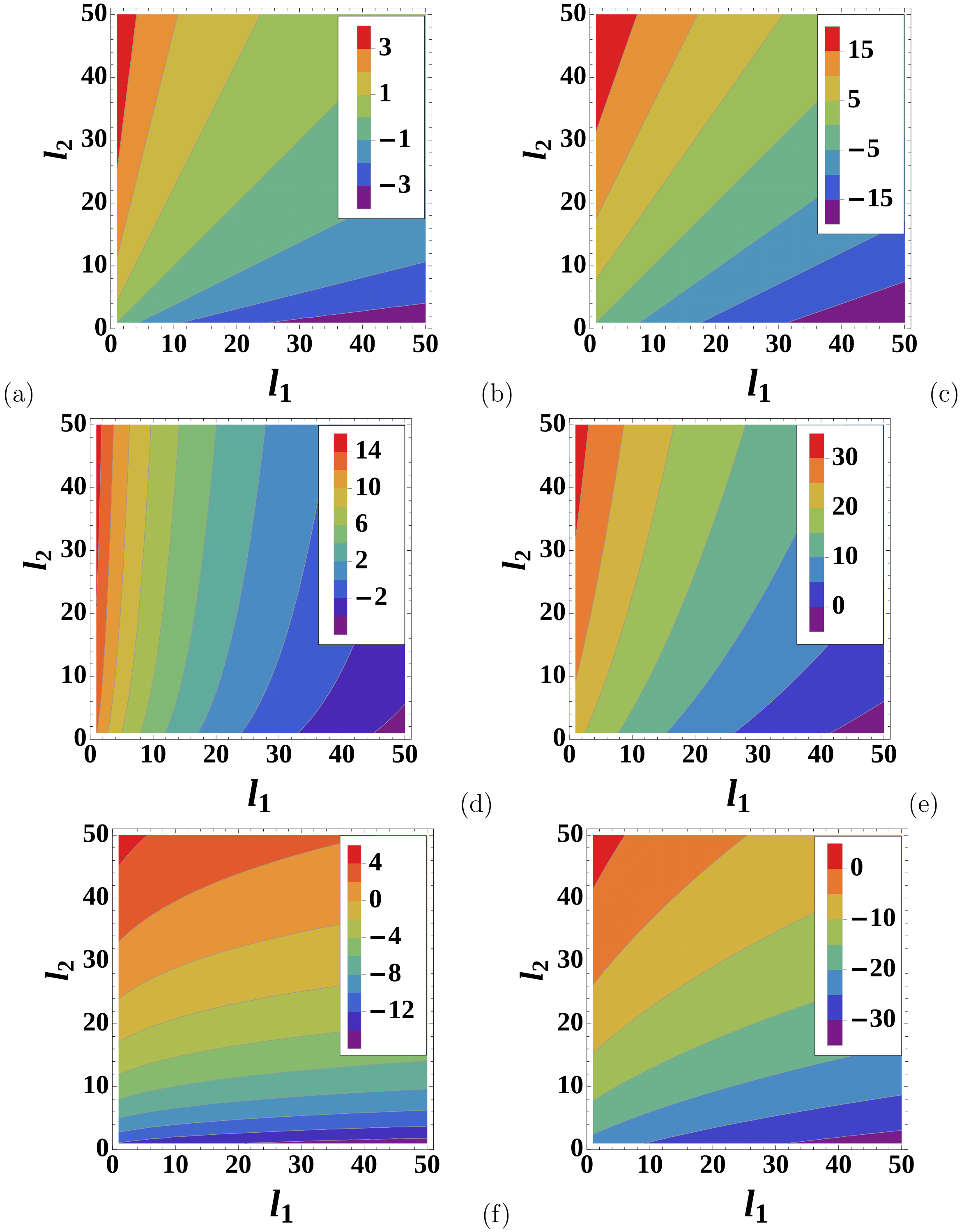


Figure B.4: Contour plots of R for various values of l_1 and l_2, where (a) $(l_{1*}, l_{2*}) = (1, 1)$, (b) $(l_{1*}, l_{2*}) = (10, 10)$, (c) $(l_{1*}, l_{2*}) = (1, 5)$, (d) $(l_{1*}, l_{2*}) = (5, 10)$, (e) $(l_{1*}, l_{2*}) = (5, 1)$, and (f) $(l_{1*}, l_{2*}) = (10, 5)$.

Acknowledgements

First, I would like to appreciate my supervisor Naomichi Hatano for valuable discussions, comments, and letting me choose the research topics freely throughout the course of the graduate school. During the course of the graduate school in The University of Tokyo, I had been supported by many people in Hatano lab. and the people who had been in the lab. I would like to thank all of them. I especially thank Tomotaka Kuwahara, Akinori Nishino, and Manabu Machida for many discussion in physics and about many miscellaneous topics. I also enjoyed the talk during the tea time with Naoya Sato.

Related to the research on the resolution limit of the map equation, I thank all the people in the Kawarabayashi ERATO Large Graph Project and Martin Rosvall. I am especially grateful to the discussion with Taro Takaguchi at the stage before I started the work on that research.

Outside the lab., I would like to thank Taketo Ariki for always reminding me (implicitly from his fashion and his action) that a true physicist should not remain ordinary, and also Hiroyuki Abe and Takeshi Watanabe for spending enormous amount of time with me to study physics together during the period of the undergraduate course and the master course.

Finally I appreciate my family for understanding the way I live and giving me a chance to have an education up to the graduate school.

Bibliography

[1] http://en.wikipedia.org/wiki/Computer_science.

[2] https://twitter.com/.

[3] https://www.facebook.com/.

[4] http://en.wikipedia.org/wiki/Galton-Watson_process.

[5] http://digg.com/.

[6] https://dev.twitter.com/docs/api, https://dev.twitter.com/docs/streaming-api.

[7] Private communication with Martin Rosvall.

[8] http://www.mapequation.org/.

[9] http://konect.uni-koblenz.de/.

[10] S. Abdullah and X. Wu. *IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 163–169, 2011.

[11] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. *Nature*, 466(7307):761–4, 2010.

[12] R. Albert and A.-L. Barabási. *Rev. Mod. Phys.*, 74(1):47, 2002.

[13] R. Aldecoa and I. Marín. *PloS ONE*, 6(9):e24195, 2011.

[14] R. Aldecoa and I. Marín. *Phys. Rev. E*, 85(2):026109, 2012.

[15] R. Aldecoa and I. Marín. *Sci. Rep.*, 3:2216, 2013.

[16] E. Aramaki, S. Maskawa, and M. Morita. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '11*, pages 1568–1576. Association for Computational Linguistics, 2011.

[17] A. Arenas, A. Díaz-Guilera, and C. Pérez-Vicente. *Phys. Rev. Lett.*, 96(11):114102, 2006.

[18] A. Arenas, A. Fernández, and S. Gómez. *New J. Phys.*, 10(5):53039, 2008.

[19] K. B. Athreya and P. E. Ney. *Branching processes*, volume 28. Springer-Verlag Berlin, 1972.

[20] J. P. Bagrow. *Phys. Rev. E*, 85(6):066118, 2012.

[21] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining - WSDM '11*, pages 65–74. ACM, 2011.

[22] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. In *Proceedings of the 21st International Conference on World Wide Web - WWW '12*, pages 519–528. ACM, 2012.

[23] A.-L. Barabási and R. Albert. *Science*, 286(5439):509–512, 1999.

[24] A. Baronchelli and V. Loreto. *Phys. Rev. E*, 73(2):026103, 2006.

[25] M. Barthélemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani. *Phys. Rev. Lett.*, 92(17):178701, 2004.

[26] M. Barthélemy and A. Flammini. *Phys. Rev. Lett.*, 100:138702, 2008.

[27] R. Bellman and T. E. Harris. *Proc. Natl. Acad. Sci. U.S.A.*, 34(12):601, 1948.

[28] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. In *Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference - CEAS '10*, volume 6, 2010.

[29] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. *Phys. Rev. E*, 83(5):056119, 2011.

[30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. *J. Stat. Mech.*, 2008(10):P10008, 2008.

[31] M. Boguñá, R. Pastor-Satorras, and A. Vespignani. *Phys. Rev. Lett.*, 90(2):028701, 2003.

[32] J. Bollen, H. Mao, and X. Zeng. *J. Comp. Sci.*, 2(1):1–8, 2011.

[33] J. Borondo, A. J. Morales, J. C. Losada, and R. M. Benito. *Chaos*, 22(2):023138, 2012.

[34] S. Brin and L. Page. *Computer Networks and ISDN Systems*, 30(1—7):107–117, 1998.

[35] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media - ICWSM '10*, 10:10–17, 2010.

[36] S. Chacon. The 2009 GitHub contest. https://github.com/blog/466-the-2009-github-contest, 2009.

[37] J. Chen and B. Yuan. *Bioinformatics*, 22(18):2283–90, 2006.

[38] A. Clauset, M. E. J. Newman, and C. Moore. *Phys. Rev. E*, 70(6):066111, 2004.

[39] P. Cogan, M. Andrews, M. Bradonjic, W. S. Kennedy, A. Sala, and G. Tucci. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research - HotSocial '12*, pages 25–31. ACM, 2012.

[40] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media - ICWSM '11*, 2011.

[41] L. D. F. Costa and L. E. da Rocha. *Eur. Phys. J. B*, 50(1-2):237–242, Apr. 2006.

[42] R. Crane and D. Sornette. *Proc. Natl. Acad. Sci. U.S.A.*, 105(41):15649–53, 2008.

[43] D. J. Daley and D. G. Kendall. *Nature*, 204:1118, 1964.

[44] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. *Proc. Natl. Acad. Sci. U.S.A.*, 107(29):12755–60, 2010.

[45] R. Diestel. *Graph Theory.* Springer-Verlag, 2005.

[46] S. N. Dorogovtsev and J. F. F. Mendes. *Adv. Phys.*, 51(4):1079–1187, 2002.

[47] P. Erdős and A. Rényi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[48] A. V. Esquivel and M. Rosvall. *Phys. Rev. X*, 1(2):021025, 2011.

[49] E. Estrada and N. Hatano. *Phys. Rev. E*, 77(3):036111, 2008.

[50] T. Evans and R. Lambiotte. *Phys. Rev. E*, 80(1):016105, 2009.

[51] M. Fiedler. *Czech. Math. J.*, 23(2):298–305, 1973.

[52] L. Fontoura Costa and F. N. Silva. *J. Stat. Phys.*, 125(4):841–872, 2006.

[53] S. Fortunato. *Phys. Rep.*, 486(3-5):75–174, 2010.

[54] S. Fortunato and M. Barthélemy. *Proc. Natl. Acad. Sci. U.S.A.*, 104(1):36–41, 2007.

[55] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. In *Proceedings of the Third Workshop on Online Social Networks - WOSN '10*, page 3. USENIX Association, 2010.

[56] J. Gani. *Environmental Modelling & Software*, 15(8):721–725, 2000.

[57] M. Girvan and M. E. J. Newman. *Proc. Natl. Acad. Sci. U.S.A.*, 99(12):7821–6, 2002.

[58] J. Goldenberg, B. Libai, S. Solomon, N. Jan, and D. Stauffer. *Physica A*, 284(1):335–347, 2000.

[59] B. Golub and M. O. Jackson. *Proc. Natl. Acad. Sci. U.S.A.*, 107(24):10833–6, 2010.

[60] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. *ACM Transactions on Knowledge Discovery from Data - TKDD '12*, 5(4):21, 2012.

[61] B. H. Good, Y.-A. de Montjoye, and A. Clauset. *Phys. Rev. E*, 81(4):046106, 2010.

[62] P. Grassberger. *Math. Bio.*, 63(2):157–172, 1983.

[63] R. Guimerà and L. A. N. Amaral. *Nature*, 433:895–900, 2005.

[64] L. Hagen and A. B. Kahng. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[65] T. E. Harris. *The theory of branching processes.* Courier Dover Publications, 2002.

[66] P. Holme, M. Huss, and H. Jeong. *Bioinformatics*, 19(4):532–538, 2003.

[67] C. Honey and S. C. Herring. In *Proceedings of the Forty-Second Hawai 'i International Conference on System Sciences - HICSS-42*, pages 1–10. IEEE, 2009.

[68] L. Hong, O. Dan, and B. D. Davison. *Proceedings of the 20th International Conference Companion on World Wide Web - WWW '11*, page 57, 2011.

[69] B. Huberman, D. Romero, and F. Wu. *Available at SSRN 1313405*, 2008.

[70] J. Iribarren and E. Moro. *Phys. Rev. Lett.*, 103(3):038702, 2009.

[71] J. L. Iribarren and E. Moro. *Phys. Rev. E*, 84(4):046116, 2011.

[72] A. Java, X. Song, T. Finin, and B. Tseng. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[73] N. Jindal and B. Liu. In *Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08*, pages 219–230, New York, NY, USA, 2008. ACM.

[74] B. Karrer, E. Levina, and M. E. J. Newman. *Phys. Rev. E*, 77(4):046119, 2008.

[75] B. Karrer and M. E. J. Newman. *Phys. Rev. E*, 83(1):016107, 2011.

[76] T. Kawamoto. *Physica A*, 392(16):3470–3475, 2013.

[77] T. Kawamoto and N. Hatano. *arXiv:1211.2555*, 2012.

[78] T. Kawamoto and M. Rosvall. *in preparation.*

[79] M. D. Kermark and A. G. Mckendrick. In *Proc. R. Soc. Lond. A*, volume 115, pages 700–721, 1927.

[80] Y. Kim and H. Jeong. *Phys. Rev. E*, 84(2):026110, 2011.

[81] Y. Kim, S.-W. Son, and H. Jeong. *Phys. Rev. E*, 81(1):016103, 2010.

[82] B. Krishnamurthy, P. Gill, and M. Arlitt. In *Proceedings of the First Workshop on Online Social Networks - WOSN '08*, pages 19–24. ACM, 2008.

[83] H. Kwak, C. Lee, H. Park, and S. Moon. In *Proceedings of the 19th International Conference on World Wide Web - WWW '10*, pages 591–600. ACM, 2010.

[84] R. Lambiotte and M. Rosvall. *Phys. Rev. E*, 85(5):056107, 2012.

[85] A. Lancichinetti and S. Fortunato. *Phys. Rev. E*, 80(5):056117, 2009.

[86] A. Lancichinetti, S. Fortunato, and F. Radicchi. *Phys. Rev. E*, 78(4):46110, 2008.

[87] A. Lancichinetti, F. Radicchi, and J. J. Ramasco. *Phys. Rev. E*, 81(4):046110, 2010.

[88] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. *PloS ONE*, 6(4):e18961, 2011.

[89] E. A. Leicht and M. E. J. Newman. *Phys. Rev. Lett.*, 100(11):118703, 2008.

[90] J. Leskovec, L. A. Adamic, and B. A. Huberman. *ACM Transactions on the Web*, 1(1):5, 2007.

[91] J. Leskovec, J. Kleinberg, and C. Faloutsos. *ACM Trans. Knowledge Discovery from Data*, 1(1):1–40, 2007.

[92] J. Leskovec, K. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[93] J. Leskovec, K. J. Lang, D. Anirban, and M. M. W. *Internet Mathematics*, 6(1):29–123, 2009.

[94] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. In *Proc. Int. World Wide Web Conf.*, pages 695–704, 2008.

[95] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proc. Int. Symp. on String Processing and Information Retrieval*, pages 1–10, 2002.

[96] J. Li and C. Cardie. *arXiv:1309.7340*, 2013.

[97] D. Liben-Nowell and J. Kleinberg. *Proc. Natl. Acad. Sci. U.S.A.*, 105(12):4633–4638, 2008.

[98] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10*, pages 939–948, New York, NY, USA, 2010. ACM.

[99] S. Lloyd. Least squares quantization in PCM. *Trans. Inf. Theory, IEEE*, 28(2):129–137, 1982.

[100] D. Lusseau. *Proc. Royal Soc. London B*, 270:S186–S188, 2003.

[101] U. Luxburg. *Statistics and Computing*, 17(4):395–416, 2007.

[102] J. MacQueen. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, page 14. California, USA, 1967.

[103] N. Masuda and N. Konno. **複雑ネットワーク** *(in Japanese).* 近代科学社, 2010.

[104] V. Mayer-Schönberger and K. Cukier. *Big Data: A Revolution that Will Transform how We Live, Work, and Think.* Eamon Dolan/Houghton Mifflin Harcourt, 2013.

[105] B. Meeder, B. Karrer, A. Sayedi, R. Ravi, C. Borgs, and J. Chayes. In *Proceedings of the 20th International Conference on World Wide Web - WWW '11*, pages 517–526. ACM, 2011.

[106] Y. Moreno, M. Nekovee, and A. F. Pacheco. *Phys. Rev. E*, 69(6):066130, 2004.

[107] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. *Science*, 328(5980):876–8, 2010.

[108] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web - WWW '12*, pages 191–200, New York, NY, USA, 2012. ACM.

[109] S. A. Myers, C. Zhu, and J. Leskovec. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12*, page 33, 2012.

[110] M. Newman. *Phys. Rev. E*, 66(1):016128, 2002.

[111] M. Newman. *Social Networks*, 25(1):83–95, 2003.

[112] M. E. J. Newman. *SIAM*, 45(2):167–256, 2003.

[113] M. E. J. Newman. *Proc. Natl. Acad. Sci. U.S.A.*, 103(23):8577–82, 2006.

[114] M. E. J. Newman and M. Girvan. *Phys. Rev. E*, 69(2):026113, 2004.

[115] M. E. J. Newman and E. A. Leicht. *Proc. Natl. Acad. Sci. U.S.A.*, 104(23):9564–9, 2007.

[116] A. Y. Ng, M. I. Jordan, and Y. Weiss. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.

[117] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. *Nature*, 435(7043):814–8, 2005.

[118] R. K. Pan and J. Saramäki. *Phys. Rev. E*, 84:016105, 2011.

[119] R. Pastor-Satorras and A. Vespignani. *Phys. Rev. Lett.*, 86(14):3200–3203, 2001.

[120] M. J. Paul and M. Dredze. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media - ICWSM '11*, 2011.

[121] T. P. Peixoto. *Phys. Rev. Lett.*, 110(14):148701, 2013.

[122] M. A. Porter, P. J. Mucha, M. E. J. Newman, and C. M. Warmbrand. *Proc. Natl. Acad. Sci. U.S.A.*, 102(20):7057–62, 2005.

[123] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. *Proc. Natl. Acad. Sci. U.S.A.*, 101(9):2658–63, 2004.

[124] J. Reichardt and S. Bornholdt. *Phys. Rev. Lett.*, 93(21):218701, 2004.

[125] J. Reichardt and S. Bornholdt. *J. Stat. Mech.*, 2007(06):P06016–P06016, 2007.

[126] J. Reichardt and M. Leone. *Phys. Rev. Lett.*, 101(7):078701, 2008.

[127] S. A. Rice. *Am. Polit. Sci. Rev.*, 21:619, 1927.

[128] A. W. Rives and T. Galitski. *Proc. Natl. Acad. Sci. U.S.A.*, 100(3):1128–33, 2003.

[129] D. M. Romero and J. M. Kleinberg. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media - ICWSM '10*, 2010.

[130] M. Rosvall and C. Bergstrom. *Proc. Natl. Acad. Sci. U.S.A.*, 2007(104):7327–7331, 2007.

[131] M. Rosvall and C. Bergstrom. *Proc. Natl. Acad. Sci. U.S.A.*, 105(4):1118–1123, 2008.

[132] M. Rosvall and C. T. Bergstrom. *PloS ONE*, 5(1):e8694, 2010.

[133] M. Rosvall and C. T. Bergstrom. *PloS ONE*, 6(4):e18209, 2011.

[134] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte. *arXiv:1305.4807*, 2013.

[135] M. T. Schaub, J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. *PloS ONE*, 7(2):e32210, 2012.

[136] M. T. Schaub, R. Lambiotte, and M. Barahona. *Phys. Rev. E*, 86(2):026112, 2012.

[137] J. Shi and J. Malik. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[138] Y. Sohn, M.-K. Choi, Y.-Y. Ahn, J. Lee, and J. Jeong. *PLoS Comput. Biol.*, 7(5):e1001139, 2011.

[139] V. Spirin and L. A. Mirny. *Proc. Natl. Acad. Sci. U.S.A.*, 100(21):12123–8, 2003.

[140] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. *Proceedings of the 2010 IEEE Second International Conference on Social Computing - SocialCom10*, pages 177–184, 2010.

[141] Y. Takhteyev, A. Gruzd, and B. Wellman. *Social Networks*, 34(1):73–81, Jan. 2012.

[142] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. *The Information Society*, 21(2):143–153, 2005.

[143] R. van der Lans, G. van Bruggen, J. Eliashberg, and B. Wierenga. *Marketing Science*, 29(2):348–365, 2010.

[144] A. Vazquez. *Phys. Rev. Lett.*, 96(3):038702, 2006.

[145] A. Vazquez. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:163–179, 2006.

[146] A. Vazquez, B. Rácz, A. Lukács, and A.-L. Barabási. *Phys. Rev. Lett.*, 98(15):158702, 2007.

[147] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in Facebook. In *Proc. Workshop on Online Social Networks*, pages 37–42, 2009.

[148] D. Wang, Z. Wen, H. Tong, C.-Y. Lin, C. Song, and A.-L. Barabási. In *Proceedings of the 20th International Conference on World Wide Web - WWW '11*, pages 735–744. ACM, 2011.

[149] D. J. Watts. *Small worlds: the dynamics of networks between order and randomness.* Princeton university press, 1999.

[150] D. J. Watts and S. H. Strogatz. *Nature*, 393(6684):440–442, 1998.

[151] R. S. Weiss and E. Jacobson. *Am. Sociol. Rev.*, 20:661, 1955.

[152] M. J. Welch, U. Schonfeld, D. He, and J. Cho. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining - WSDM '11*, page 327, 2011.

[153] M. B. Wilk and R. Gnanadesikan. *Biometrika*, 55(1):1–17, 1968.

[154] D. Wilkinson and B. Huberman. *Proc. Natl. Acad. Sci. U.S.A.*, 101:5241–8, 2004.

[155] D. M. Wilkinson. *Proceedings of the 9th ACM Conference on Electronic Commerce - EC '08*, page 302, 2008.

[156] R. J. Wilson. *Introduction to Graph Theory, Fourth edition.* Pearson Education, 1996.

[157] F. Wu and B. A. Huberman. *Proc. Natl. Acad. Sci. U.S.A.*, 104(45):17599–601, 2007.

[158] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. *Proceedings of the 20th international conference on World wide web - WWW '11*, page 705, 2011.

[159] K.-K. Yan and M. Gerstein. *PloS ONE*, 6(5):e19917, 2011.

[160] J. Yang and S. Counts. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media - ICWSM '10*, 10:355–358, 2010.

[161] J. Yang and J. Leskovec. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining - WSDM '11*, page 177, 2011.

[162] D. Yin, L. Hong, and B. D. Davison. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11*, page 1163, 2011.

[163] D. Yin, L. Hong, X. Xiong, and B. D. Davison. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information - SIGIR '11*, page 1235, 2011.

[164] W. W. Zachary. *J. Anthropol. Res.*, pages 452–473, 1977.

[165] J. Zhou, Z. Liu, and B. Li. *Phys. Lett. A*, 368(6):458–463, 2007.