# RUMOR SOURCE IDENTIFICATION IN COMPLEX NETWORKS

By

Jiaojiao Jiang

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Deakin University

January 2017

# DEAKIN UNIVERSITY
# ACCESS TO THESIS - A

I am the author of the thesis entitled

Rumor Source Identification In Complex Networks

submitted for the degree of Doctor of Philosophy

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

*'I certify that I am the student named below and that the information provided in the form is correct'*

**Full Name:** Jiaojiao Jiang
(Please Print)

**Signed:** Signature Redacted by Library

**Date:** 6 May 2017

# DEAKIN UNIVERSITY
# CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Rumor Source Identification In Complex Networks

submitted for the degree of Doctor of Philosophy

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

*'I certify that I am the student named below and that the information provided in the form is correct'*

**Full Name:** Jiaojiao Jiang
(Please Print)

**Signed:** Signature Redacted by Library

**Date:** 6 May 2017

DEAKIN UNIVERSITY
SCHOOL OF
INFORMATION TECHNOLOGY

The undersigned hereby certify that they have read and recommend to the Faculty of Science, Engineering and Built Environment for acceptance a thesis entitled "**RUMOR SOURCE IDENTIFICATION IN COMPLEX NETWORKS**" by **Jiaojiao Jiang** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dated: January 2017

External Examiner: _____

Research Supervisor: _____
Shui Yu

Examing Committee: _____

_____

*To my family...*

# Table of Contents

# List of Tables

# List of Figures

# Acknowledgements

Firstly, I would like to express my deepest gratitude and appreciation to my supervisors Dr. Shui Yu, Prof. Wanlei Zhou, and Dr. Simon James. They have given me so much advise, encouragement, and endless support during the three-year journey of my PhD study at Deakin University. I appreciate all the time they have spent on editing my papers, discussing my research ideas, listening to my problems, and taking away all my doubts and worries. I will always be very grateful for their guidance, support and encouragement.

Secondly, I would like to give my sincere thanks to my research collaborators, Prof. Yang Xiang, and Dr. Sheng Wen. I have acquired a rich set of skills and experiences from their research experiences, knowledge, and meticulous attitude to research during my PhD expedition. Every piece of helpful suggestion makes me better and makes me enjoy research more.

I would also like to thank the academic and general research staff of the School of Information Technology, Deakin University. I would like to give my thanks and appreciation to Ms. Alison Carr and Ms. Lauren Browne for their great secretarial work. Thanks to everyone working in the Research Center for Cyber Security Research, I was able to easily access a large collection of Research data to facilitate my research. I also owe a great deal of thanks to my friends who helped me in my study and daily life during my PhD study: Xiao Chen, Dr. Jun Zhang, Prof. Xinyi Huang, and so many others.

I also would like to thank my family for their support and love. Without their everlasting encouragement and understanding, this thesis wouldn't exist.

Melbourne, Australia — Jiaojiao Jiang
Date and Time

# List of Publications

## Refereed Journal Articles

1. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou, "Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies," *IEEE Communications Surveys and Tutorials*, accepted on September 17, 2014.

2. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou, "K-center: An Approach on the Multi-source Identification of Information Diffusion," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, 2015, pp. 2616-2626.

3. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou, "Rumor Source Identification in Social Networks with Time-varying Topology," *IEEE Transactions on Dependable and Secure Computing*, accepted on December 30, 2015.

4. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou, "The Structure of Communities in Scale-free Networks," *Concurrency and Computation: Practice and Experience*, accepted on January 1, 2016.

5. Sheng Wen, **Jiaojiao Jiang**, Yang Xiang, Shui Yu, Wanlei Zhou, and Weijia Jia, "To Shut Them Up or To Clarify: Restraining the Spread of Rumors in Online Social Networks," *IEEE Transactions on Parallel and Distributed Systems*,

vol. 25, issue 12, 2014, pp. 3306-3316.

6. Sheng Wen, **Jiaojiao Jiang**, Bo Liu, and Wanlei Zhou, "Are the Popular Users Always Important for the Information Dissemination in Online Social Networks?" IEEE Network, vol. 28, issue 5, 2014, pp. 64-67.

7. Sheng Wen, **Jiaojiao Jiang**, Yang Xiang, Shui Yu, and Wanlei Zhou, "Using Epidemic Betweenness to Measure the Influence of Users in Complex Networks," Journal of Network and Computer Applications, accepted on November 2, 2016.

8. Haibin Zhang, **Jiaojiao Jiang**, and Zhi-Quan Luo, "On the Linear Convergence of a Proximal Gradient Method for a Class of Nonsmooth Convex Minimization Problems," *Journal of the Operations Research Society of China*, vol. 1, issue 2, 2013, pp. 163-186.

# Refereed Conference Papers

1. **Jiaojiao Jiang**, Andi Zhou, Kasra Majbouri Yazdi, Sheng Wen, Shui Yu, and Yang Xiang, "Identifying Diffusion Sources in Large Networks: A Community Structure Based Approach," *IEEE Trustcom2015*, vol. 1, pp. 302-309, 2015.

2. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, Wanlei Zhou, and Yi Qian, "Analysis of the Spreading Influence Variations for Online Social Users under Attacks," *IEEE GlobeCom2016*, Washington, DC USA, December 4-8, 2016.

3. **Jiaojiao Jiang**, Sheng Wen, Shui Yu, and Wanlei Zhou, "Studying the Global Spreading Influence and Local Connections of Users in Online Social Networks," *IEEE SocialSec2016*, Yanuca Island, Fiji, December 7-10, 2016.

4. Sheng Wen, **Jiaojiao Jiang**, Kasra Majbouri Yazdi, Yang Xiang, and Wanlei Zhou, "The Relation between Local and Global Influence of Individuals in Scale-Free Networks," *Security and Privacy in Social Networks and Big Data (SocialSec), International Symposium on IEEE*, 2015, pp. 80-84.

5. BoHao Feng, Huachuan Zhou, Hongke Zhang, **Jiaojiao Jiang**, and Shui Yu, "A Popularity-based Cache Consistency Mechanism for Information-Centric Networking," *IEEE GlobeCom2016*, Washington, DC USA, December 4-8, 2016.

6. Bo Liu, Wanlei Zhou, **Jiaojiao Jiang**, and Kun Wang, "K-source: Multiple Source Selection for Traffic Offloading in Mobile Social Networks," *IEEE WCSP2016*, Yangzhou, Jiangsu, China, October 13-15, 2016.

# Abstract

In the modern world, the ubiquity of networks has made us vulnerable to various *network risks*. For instance, computer viruses propagate throughout the Internet and infect millions of computers. Misinformation spreads incredibly fast in online social networks, such as Facebook and Twitter. Infectious diseases, such as SARS, H1N1 or Ebola, have spread geographically and killed hundreds of thousands people. In essence, all of these situations can be modeled as ***a rumor spreading through a network***, where the goal is to find the source of the rumor in order to control and prevent these network risks. So far, extensive work has been done to develop new approaches to effectively identify rumor sources. However, current approaches still suffer from critical weaknesses. The most difficult one is the complex spatiotemporal diffusion process of rumors in *time-varying networks*, which is the bottleneck of current approaches. The second problem lies in the expensively computational complexity of identifying *multiple rumor sources*. The third important issue is the *huge scale of the underlying networks*, which makes it difficult to develop efficient strategies to quickly and accurately identify rumor sources. These weaknesses prevent rumor source identification from being applied in a broader range of real-world applications. This thesis aims to address these issues to make rumor source identification more effective and applicable.

The first issue is overcome by proposing an *analytical model* for modeling rumor spreading in dynamic networks. Traditional approaches assume firm connections between individuals (*i.e.*, static networks) so that people can trace back along the determined connections to reach the spreading sources. In this thesis, we consider the

physical mobility and online/offline status of individuals in modeling rumor spreading. Furthermore, we propose a novel reverse dissemination strategy to narrow down the scale of suspicious sources, which dramatically promotes the efficiency of our method. We then develop a Maximum-likelihood estimator, which can pinpoint the true source from the suspects with a high accuracy. Experiment results justify the effectiveness of the proposed method in real-world time-varying networks.

We address the second issue through developing an *optimization framework* for multi-source identification issue. We adapt K-means from data mining and effective distance to structure *diffusion pattern* of multi-source rumor spreading. After this, we formulate an optimization problem for multi-source identification, and develop a fast method to solve the problem. Theoretical analysis proves the efficiency of the proposed method, and the experiment results demonstrate the effectiveness of the proposed method in various real-world networks.

For the scalability issue in rumor source identification, we explore sensor techniques and develop a *community structure based method*. Instead of assigning sensors on high-centrality nodes in traditional methods, we propose placing sensors on community bridges. Thus, we can efficiently record the rumor diffusion between communities rather than between individuals. Then, we take the advantage of the linear correlation between rumor spreading time and rumor infection distance to develop a fast method which can locate the rumor source with high accuracy. Theoretical analysis proves the efficiency of the proposed method, and the experiment results verify the significant advantages of the proposed method in large-scale networks.

In summary, this thesis makes three major contributions: 1) propose a novel approach to identify rumor sources in *time-varying networks*; 2) develop a fast approach to identify *multiple rumor sources*; 3) propose a community-based method to overcome the *scalability issue* in this research area. These contributions enable rumor source identification to be applied effectively in real-world networks, and eventually diminish rumor damages.

**Keywords:** Complex networks, rumor diffusion, source identification.

# Chapter 1

# Introduction

## 1.1 Motivation

With rapid urbanization and advancements in transportation technologies, the world has become more interconnected. A *contagious disease*, like SARS [63], H1N1 [29] and Ebola [103], can spread quickly through a population and lead to an epidemic [56]. It is crucial to quickly identify the set of epidemic sources, so that potential containment policies can be formulated to prevent further spreading of the disease [92]. In a similar vein, *computer viruses*, like Cryptolocker and Alureon [62], on a few servers of a computer network can quickly spread to other servers or computers in the network and cause a good share of cyber-security incidents [106]. Identifying the servers in the network that are first infected allows us to detect the latent points of weaknesses in the computer network, so that preventive measures can be taken to enhance the protection at these points. The source identification problem also arises in the study of *misinformation spreading* in a social network. A piece of misinformation like Barack Obama was born in Kenya started by a few individuals can spread quickly through the underlying social network [21,78]. In many cases, we are interested to find the sources

of the misinformation. For example, law enforcement agencies may be interested in identifying the perpetrators who fabricate misinformation to manipulate the market prices of certain stocks.

In essence, all of the above examples can be modeled as *a rumor spreading in a network of nodes*. In a population network, the rumor is the disease that is transmitted between individuals. In the example of a computer virus spreading in a network, the rumor is the computer virus, while for the case of a misinformation spreading in a social network, the rumor is the misinformation. In this thesis, we focus on the problem of identifying rumor diffusion sources: given a complex network and a partial observation of rumor diffusion, determine the rumor diffusion source(s).

From both practical and technical aspects, it is of great *significance* to identify rumor sources. Practically, it is important to accurately identify the 'culprit' of the rumor propagation for forensic purposes. Moreover, seeking the rumor sources as quickly as possible can find the causation of rumors, and therefore, mitigate the damages. Technically, the work in this field aims at identifying the sources of rumors based on limited knowledge of network structures and the states of a small portion of nodes. In academia, traditional identification techniques, such as IP trace back [91] and stepping-stone detection [93], are not sufficient to seek the sources of rumors, as they only determine the true source of packets received by a destination. In the propagation of rumors, the source of packets is almost never the source of the rumor propagation but just one of the many propagation participants [114]. Methods are needed to find propagation sources higher up in the application level and logic structures of networks, rather than in the IP level and packets.

In the past few years, researchers have proposed a series of methods to identify

rumor diffusion sources. The initial methods were designed to work on particular networks (*e.g.*, regular tree and regular graphs) and with the diffusion following the traditional susceptible-infected (SI) model [22, 42, 95, 96]. Later, some other works were proposed to deal with particular networks but with different epidemic models, such as the susceptible-infected-recovered (SIR) model and the susceptible-infected-susceptible (SIS) model [55,59,120,121]. The constraints on particular networks were then relaxed to generic network topologies but still assume that rumors spread along the breadth-first search (BFS) trees of networks [12, 25, 53, 84]. Recently, researchers proposed methods to identify propagation sources by using sensor techniques [1, 82, 94], but they are still restricted in the BFS trees of networks. In the real world, rumor diffusion is a complex process and it does not always follow the ideal BFS-tree spreading scheme. It can be affected by the impacts from the dynamic of individuals, the impacts from the structure of the underlying network, the impacts from other related rumors, etc. Therefore, previous methods of rumor sources identification are far from applying effectively and efficiently in real-world networks.

In many ways, current approaches of rumor source identification are facing the following three critical challenges.

- The underlying networks are often of *time-varying topology*. For example, in human contact networks, the neighborhood of individuals moving over a geographic space evolves over time, and the interaction between the individuals appears/disappears in online social network websites (such as Facebook and Twitter) [87]. Indeed, the spreading of rumors is affected by duration, sequence, and concurrency of contacts among nodes [13, 104]. Then, can we model the

way that rumor spreads in time-varying networks? Can we estimate the probability of an arbitrary node being infected by a rumor? How do we detect rumor source in time-varying networks? Can we estimate the infection scale and infection time of the rumor?

- Rumors often emerge from *multiple sources*. However, current methods mainly focus on the identification of a single rumor source in networks. A few approaches are proposed for identifying multiple rumor sources but they all suffer from extremely high computational complexity, which is not practical to be adopted in real-world networks. In this thesis, we will answer the following questions corresponding to multi-source identification. How many sources are there? Where did the diffusion emerge? When did the diffusion start?

- Another critical challenge in this research area is the *scalability issue*. Current methods generally require scanning the whole underlying network of rumor spreading to locate rumor sources. However, real-world networks of rumor diffusion are often of a huge scale and extremely complex structure. Thus, it is impractical to scan the whole network to locate the rumor sources. We develop efficient approaches to identify rumor sources by taking the structural features of networks and the diffusion patterns of rumors into account, and therefore address the scalability issue.

To address the above challenges, this thesis aims to achieve a breakthrough in rumor source identification to enable its effective applicability in real world applications. The approaches involve the complex network theory, information diffusion theory, probability theory, and applied statistics.

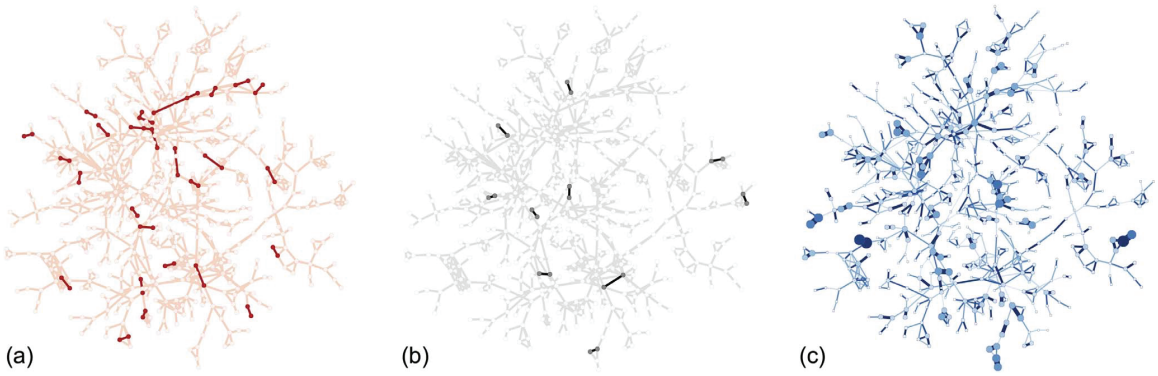



Figure 1.1: Illustration of time-varying mobile-phone call (MPC) network [44]. Panels (a), and (b) show calls within 3 hours between people in the same town in two different time windows. Panel (c) presents the total weighted social network structure, which was recorded by aggregating interactions during 6 months. Node size and colors describe the activity of users, while link width and color represent weight.

## 1.2 Research Questions

As presented in the previous section, current studies on rumor source identification encounter three critical challenges: *time-varying networks*, *multiple rumor sources* and *scalability issue*. In this section, we will introduce the research questions examined in each chapter in detail.

### 1.2.1 Rumor Diffusion in Time-varying Networks

In the real world, it takes different periods of time for nodes to transmit information to their neighbors. The temporal dynamic is an important factor, particularly when the propagation concerns human involvements [110]. Let us take the mobile phone call (MPC) network [44] as a example. Fig. 1.1(A) and 1.1(B) show two snapshots of the MPC network at different times covering a few hours of calls in a town. The

two plots capture dynamical interaction patterns not visible from the aggregated network representation (Fig. 1.1(C)). Traditional methods of rumor source identification assume the firm connection between individuals (*i.e.*, the network in Fig. 1.1(C)). However, this will dramatically overfit the actual time-varying network structure. This is also the reason that traditional methods present a low accuracy in identifying rumor sources in real-world networks.

Technically, the *temporal dynamic of networks* is complex. It involves the impact of the time zone and the population distribution [18]. Individual habits also strongly affect the temporal dynamic of rumor diffusion. Currently, litter literature considers temporal dynamics of the underlying network where rumors diffuse. We consider these factors in modeling rumor propagation. In other words, we model rumor propagation by considering the realistic temporal dynamics of individuals and their interactions. Based on the innovative model, we can trace back the rumor diffusion source and also predict its future trend. This will make a fundamental contribution to rumor source identification, and open up a new direction of modeling rumor propagation. Therefore, identifying rumor sources in time-varying networks is the first research question to address in this thesis.

### 1.2.2 Rumor Diffusion with Multiple Sources

In the real world, the propagation of rumors often initiates from multiple sources. For example, a contagious disease emerging from a small population can spread geographically and infect hundreds of thousands of people [56]. Culprits employ a botnet to spread computer viruses and finally infect millions of computers and servers [8,28]. Fig. 1.2 shows an example to illustrate the rumor diffusion starting from two sources.

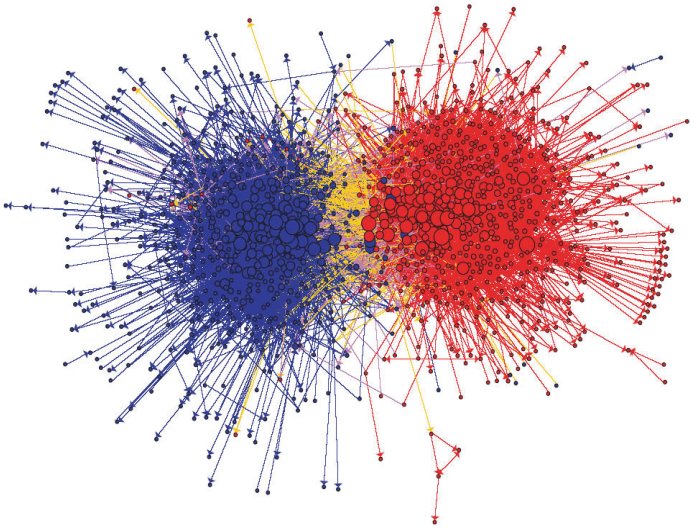


Figure 1.2: Illustration of the diffusion with two rumor sources. The blue group of nodes hear the rumor from one source, and the red group hear the rumor from the other source. The yellow nodes are those who receive rumors from both sources.

One source mainly infects the blue nodes, the other source mainly contaminates the red nodes, the yellow nodes are infected by both sources. Few of current methods are developed for multi-source identification. Some single-source identification methods can be adapted to identify multiple rumor sources. However, they all suffer from the expensive computational complexity (generally $O(N^k)$, where $N$ is number of infected nodes and $k$ is the number of sources). Therefore, they are not practical to be applied in real-world networks.

Technically, the methods of single source identification cannot be directly used for multiple source cases. This is because the spread initiated from multiple sources cannot be thought of as the superposition of multiple single-source propagation processes. Meanwhile, current multi-source identification methods are too computationally expensive to obtain results. Moreover, current methods ignore an important factor: *the*

*pattern of multi-source rumor diffusion*. This is the crucial factor that we consider in developing new methods for multi-source identification. Using the pattern of multi-source rumor diffusion, we substantially simplify the multi-source identification problem and develop a fast method to solve this problem. Therefore, the identification of multiple rumor sources is the second research question to address in this thesis.

### 1.2.3 Rumor Diffusion in Large-scale Networks

In the real world, rumor diffusion often occurs in large-scale networks, such as human contact networks, online social networks, computer networks or the World Wide Web. Let us take a real event on Twitter for an example. On April 15th of 2013, two explosions at the Boston Marathon finish line shocked the entire United States [101]. There were millions of tweets about it, and many of the tweets contained rumors and misinformation. Within a couple of days, multiple pieces of misinformation went viral on various social media. The huge scale of the Twitter network severely challenges traditional rumor source detection methods. Therefore quickly identifying rumor sources in large-scale networks is of great significance in practice.

Technically, current methods are too computationally expensive to quickly and accurately detect rumor sources in large-scale networks. This is because most of current methods require scanning the entire network to locate the rumor sources. Moreover, current methods ignore an important fact that rumor diffusion often presents a network-driven phenomenon. We show an example in Fig. 1.3 to illustrate one of the important structures of complex networks − community structure. The left plot presents the diverse communities in the network. The right plot shows the network

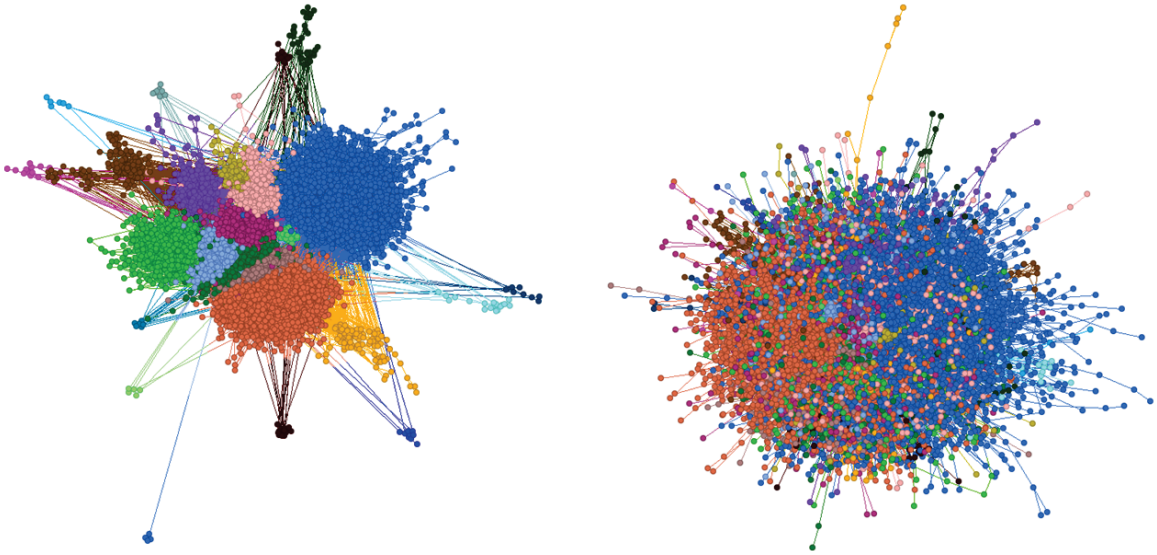


Figure 1.3: Left: The community construct of a network. Right: The observed network.

in our observation. Appropriately utilizing the structure of networks can facilitate our rumor source detection work in large-scale networks. This is the crucial factor we will consider in developing new methods for identifying rumor sources in large-scale networks. Accordingly, based on the structure of the underlying network, we dramatically decrease the scanning of the entire network to scanning a small community of the network. Therefore, we propose developing efficient and scalable source identification methods as the final research question in this thesis.

## 1.3 Thesis Outline

This section aims to establish the structural organization of the thesis. According to three research issues addressed in this thesis: *time-varying networks*, *multiple rumor sources* and *scalability issue*, the chapters are organized as follows.

- Chapter 2 introduces the *preliminary knowledge* adopted in this thesis, including

some basic concepts and network generating models in graph theory, information propagation models and the Maximum-Likelihood Estimation method.

- Chapter 3 presents a *comprehensive survey* on the development of rumor source identification, including relevant concepts, assumptions, and emerging techniques in this area. Efforts have been given to identify various research directions and emerging research issues.

- Chapter 4 focuses on the *time-varying networks* issue in the context of network-driven rumor propagation. This chapter specifically investigates how to model rumor diffusion in a dynamic network by reducing the network into a series of static network windows. This chapter also proposes an effective two-stage method to detect rumor sources in time-varying networks. The first stage narrows down the suspicious rumor sources. The second stage pinpoints the true source from the suspects. The method involves developing a reverse dissemination method to trace back rumor in time-varying networks, and a maximum-likelihood method is explored to compute the probability of the status of arbitrary individuals.

- Chapter 5 proposes a novel *K-center method* to address the *multiple rumor sources* issue. This chapter focuses on how to efficiently and effectively detect multiple rumor sources by exploring multi-source rumor diffusion patterns. Through combining the diffusion patterns and network partition techniques, this chapter formulates the multi-source identification problem as an optimization problem and proposes a fast method to solve the problem. This chapter also addresses three important problems in this area: (i) *estimating the number*

*of rumors sources*, (ii)*locating the topological places where the rumor emerges*, and (iii) *estimating the time when the rumor breaks out.*

- Chapter 6 aims at addressing the *scalability issue* in identifying rumor sources in *large-scale networks*. This chapter specifically explores how to utilize the structural patterns of networks for rumor propagation. With the consideration of community structure and sensor techniques in complex networks, this chapter proposed a community-structure based method. Compared with the traditional methods, the proposed community-structure based method dramatically decreases the searching scale of rumor sources from the entire network to a small community.

- Chapter 7 summarizes the contributions of this thesis, and presents some possible suggestions and extensions for further research.

To maintain the readability, each chapter is organized in a self-contained format, and some essential contents, e.g. definition, are briefly recounted in related chapters.

# Chapter 2

# Preliminaries

This chapter provides some preliminary work about rumor source identification. We first introduce some concepts related to complex networks, including node centralities, network generating models, and community detection methods. Secondly, we introduce three classic information diffusion models adopted in this thesis. Finally, we explain the maximum-likelihood estimation method adopted in this thesis.

## 2.1 Complex Networks

### 2.1.1 Node Centralities

**Degree** Given a node $i$, the degree of node $i$ is the number of edges connected to node $i$. In general, the larger degree of a node, the more influential of the node (see the black nodes in Fig. 2.1(A)).

**Betweenness** The betweenness of a node stands for the number of shortest paths from all nodes to all others that pass through the node [31]. Researchers have found some nodes which do not have large degrees also play a vital role in the information diffusion [38, 52]. As shown in Fig. 2.1(B), the degree of node $E$

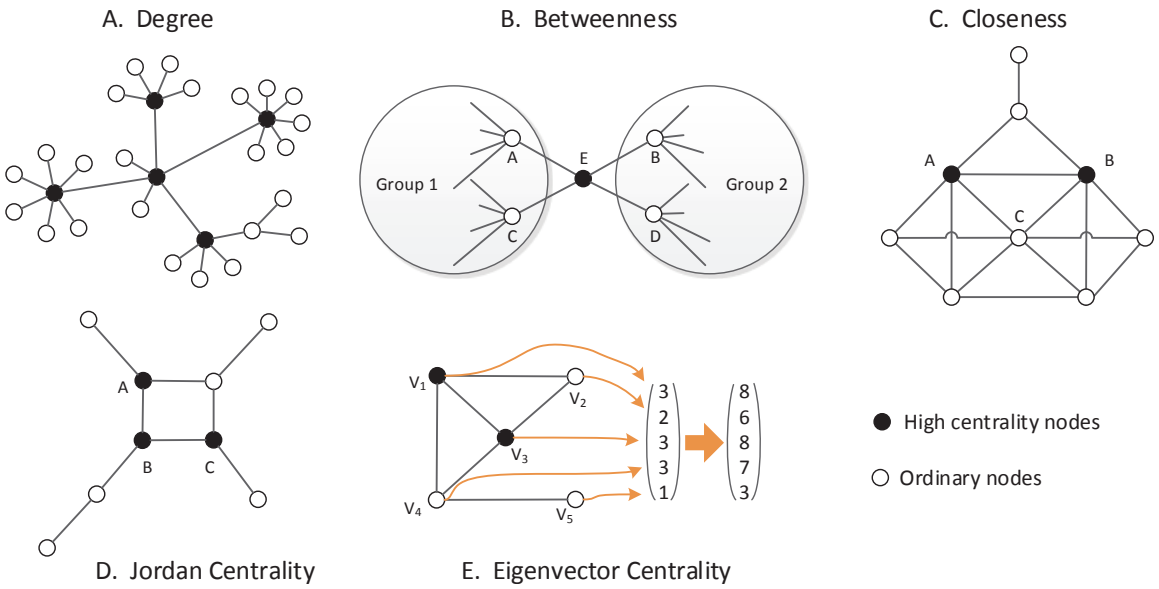



Figure 2.1: Illustration of different centrality measures. (A) Degree; (B) Betweenness; (C) Closeness; (D) Jordan centrality; (E) Eigenvector centrality.

is smaller than nodes $A$, $B$, $C$ and $D$. However, node $E$ is noticeably more important to information spread as it is the connector of two large groups. To locate this kind of nodes, researchers introduced the measure of betweenness.

**Closeness** The closeness of a node is defined as the average length of the shortest path between the node and all other reachable nodes [31, 74]. As shown in Fig. 2.1(C), this measure discloses the nodes that can rapidly disseminate information to all the other nodes. This measure concentrates more on the information propagation speed rather than the connectivity of a network [74].

**Jordan centrality** The Jordan centrality of a node measures the maximum geodesic distance (shortest-path distance) from the node to a given set of infected nodes in the network [20, 35]. *The Jordan centers* stand for the nodes that have

minimum Jordan centrality. Suppose all the nodes are infected in the graph in Fig. 2.1(D), then nodes $A$, $B$, $C$ are the Jordan centers of the graph with Jordan centrality 3. Equivalently, Jordan center is equal to the radius of a network [102].

**Eigenvector centrality** It is defined as the eigenvector of the adjacency matrix associated to the largest eigenvalue [11, 71]. Equivalently, the eigenvector centrality of a node is proportional to the sum of the eigenvector centrality of all its neighboring nodes. In the real world, an important node is characterized by its connectivity to other important nodes. Thus, a node with a high eigenvector centrality is a well-connected node and has a dominant influence on the surrounding network. As shown in Fig. 2.1(E), node $V_1$ and $V_3$ have the highest eigenvector centrality in the graph. Readers could refer to [11] for further computation methods.

### 2.1.2 Network Generating Models

Various models devoted to reproducing the growth and evolution of network topology have been developed to capture different characteristics of complex networks. In the following subsections, we will introduce three classic network generating models.

#### Random Networks

The first network generating model proposed in 1959 by Erdos and Renyi [1960] described the process of growing a *random network*: $n$ nodes connected by $m$ edges randomly selected from all $n(n-1)/2$ possible edges with equal probability $p$. The degree of nodes in the *Erdos-Renyi (ER) network* presents a Poisson distribution. The

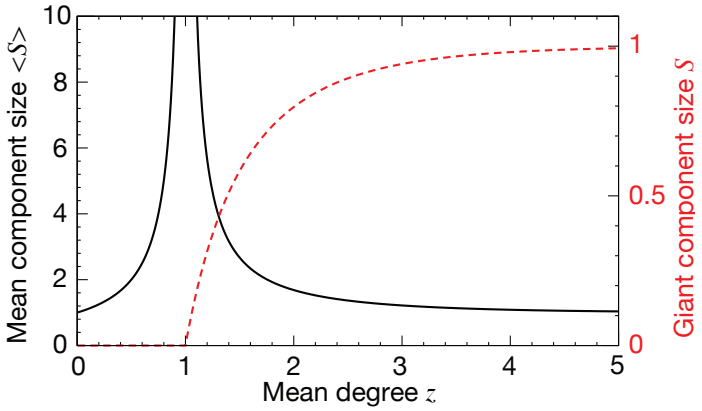



Figure 2.2: The plot of the mean component size excluding the giant component if there is one (black solid line), and the giant component size (red dashed line), for the ER random network [69]. The mean degree $z = p(n-1)$.

other key feature is a sudden change of the network connectivity with the increase of $p$: when $p$ is small, many clusters are small and isolated, but once $p$ increases to be larger than a critical value, the network suddenly becomes very dense where almost all the nodes are linked to each other in a giant connected component (see Fig. 2.2).

### Small-World Networks

The *small-world network* originated from the experiment of Milgram [65], in which selected persons were asked to deliver a letter to a target receiver by only passing the letter to their acquaintances. Among all the successful instances, the average length of these communication chains was short, around six steps. The phenomenon is well known as "small-world effect" or "six degrees of separation". A small-world network has acquaintanceship-based edges and the distance between a random pair of people is smaller than expected. In the real world setting, the small-world effect implies that most of the friends of an individual are people living around, but he may also

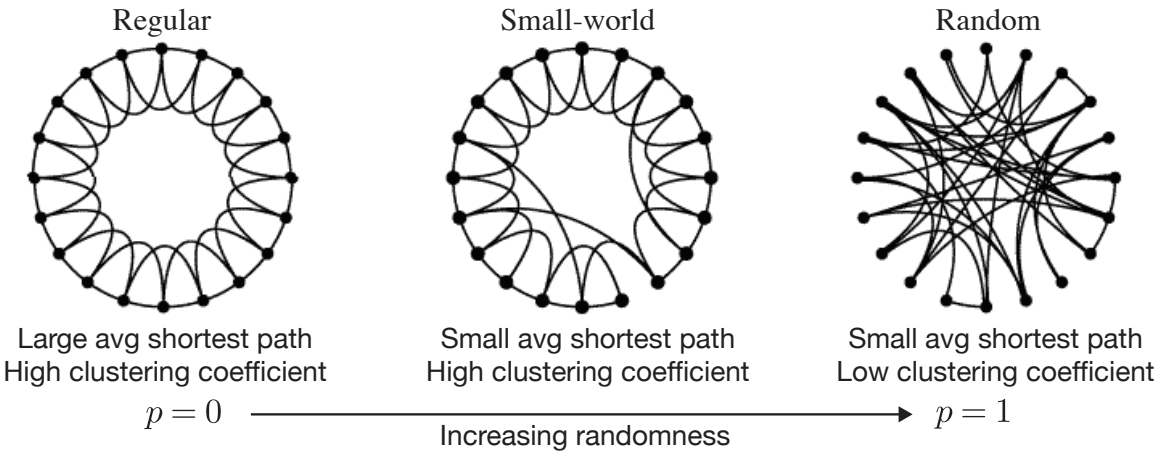



Figure 2.3: The Watts-Strogatz model reproduces the small-world phenomenon by rewiring edges in a regular network according to the randomness parameter $p$ [108].

have a few friends far away. People are moving around, but the geographic distance limits the strength of social relationships. The *Watts-Strogatz model* was designed to reproduce the small-world phenomenon by rewiring each link in a regular network with a probability $p$ [108]. As shown in Fig. 2.3, when $p = 0$, the network is fully ordered; when $p = 1$, every edge is rewired so as to create a random network; when $0 < p < 1$, we obtain a small-world network with small average shortest path and high clustering coefficient [108].

### Scale-Free Networks

A scale-free network has a power-law degree distribution, commonly seen in many real-world networks, such as the Internet, the film actor network, the scientific collaboration network, the citation network, and many others (see Fig. 2.4) [4, 9, 69]. Highly unbalanced degree distribution in a social network indicates that, in a large group of people, only a few are extremely popular and most others do not have too

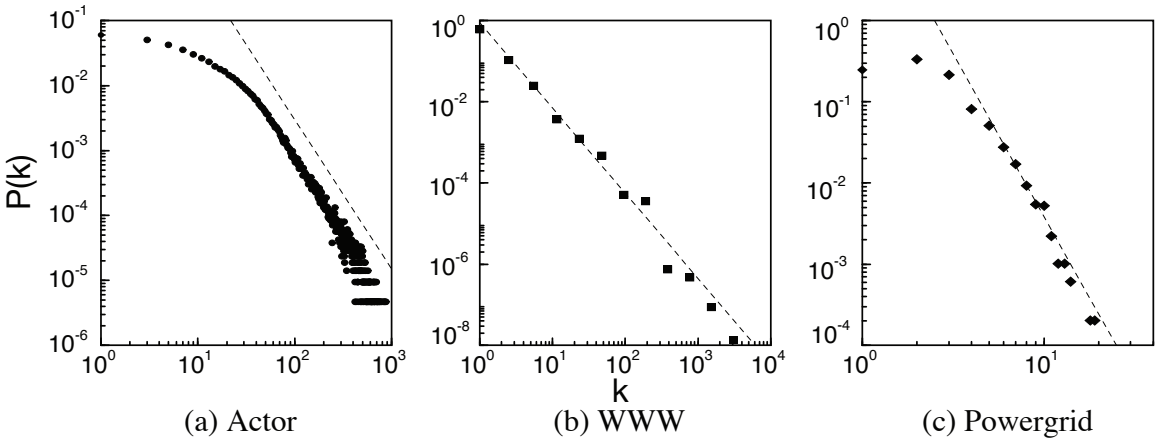



Figure 2.4: The connectivities of various large real-world networks have scale-free distributions, (a) actor collaboration graph, (b) the World Wide Web, and (c) the power grid network [9].

many contacts. It has been suggested to be the most critical feature of social networks [73].

Among many models that can capture the heterogeneous distribution in connectivity [23, 27, 47–49, 73], Barabasi-Albert model was the first to generate a scale-free network with two simple mechanisms: continuously adding new nodes into the system ("growth") and connecting with other nodes with preference to the high-degree ones ("preferential attachment") [9]. Motivated by the structure of the Web graph, the copying model added a new node into the network and linked it to a random existing node or its neighbors [47, 49]. Another model proposed by Newman et al. [73] aimed to build up a random graph with the arbitrary degree setting. The ranking model grew the network according to a rank of the nodes by any given prestige measure; the probability of linking a target node could be any power law function of its rank, resulting in a power-law degree distribution [27].

### 2.1.3 Community Detection

A *community* is a group of densely connected nodes in a graph. The community structure is claimed to be one key property of various complex networks, suggesting that a network can be partitioned into several (potentially overlapped) clusters so that nodes in one cluster are densely connected internally but not externally; such clustering might derive from common interests of people, geographical divisions of power grids, or functional similarity of proteins [32, 72]. How to detect communities has been widely studied [26], popular methods including modularity optimization [70], Louvain method [10], infomap [89], clique percolation [76], and link clustering [2]. The two methods, *Infomap* and *link clustering*, applied in the thesis are introduced as follows.

**Infomap** The infomap community detection method is built on the assumption that a random walker is more likely to be trapped in communities than to travel between communities. The path of a random walker can be encoded, and then compressed given a hierarchical network partition so that the encoded description is minimum. The duality between finding community structure in a network and the coding problem is: to find an efficient code, it looks for a module partition $M$ of $n$ nodes into $m$ modules so as to minimize the expected description length of a random walk. By using the module partition $M$, the average description length of a single step is given by

$$L(M) = q_{\curvearrowright} H(\mathcal{L}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i), \tag{2.1.1}$$

where $H(\mathcal{L})$ is the entropy of module names in $M$; $H(\mathcal{P}^i)$ is the entropy of

intra-module movements; $q_{\curvearrowright}$ gives the probability that the random walk switches modules on a given step; $p^i_{\circlearrowright}$ is the sum of the probability of intra-module movements inside the module $i$ and the probability of exiting $i$. The first part of the formula describes the entropy of the movement between communities, and the second part sums up the entropy within each community. Eventually infomap applies computational search algorithm to find the best partition as the outcome [89].

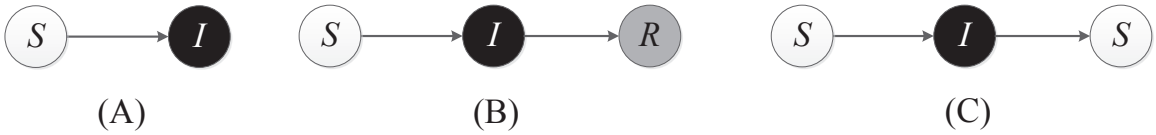
**Link clustering** Different from the Infomap method, the *link clustering algorithm* aims at discovering overlapped communities in which a node is allowed to belong to multiple groups. The link clustering algorithm reinvents communities as groups of links rather than nodes. The set of neighbors of a node $i$ is denoted as $N_i$. Given a pair of links with one shared node, $e_{ij}$ and $e_{jk}$, the similarity between these two links is the Jaccard similarity between neighbor sets of distinct nodes:

$$S(e_{ij}, e_{jk}) = \frac{|N_i \cap N_k|}{|N_i \cup N_k|}. \tag{2.1.2}$$

Then a dendrogram is built up according to these similarities using single-linkage hierarchical clustering and cutting the dendrogram at some level produces the overlapped community structure. Given a partition $P = \{P_1, P_2, ..., P_C\}$, a partition density $D$ can be computed by the average partition density weighted by the fraction of present links in each partition:

$$D = \sum_c \frac{m_c}{M} D_c = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}, \tag{2.1.3}$$

where $m_c$ and $n_c$ are the numbers of edges and nodes in the partition $P_c$, respectively. The cutting threshold in the dendrogram can be determined by achieving




(A) (B) (C)

Figure 2.5: Illustration of three classic epidemic spreading models. (A) SI model; (B) SIR model; (C) SIS model.

a maximum partition density.

## 2.2 Information Diffusion Models

Early models concerning communication dynamics were inspired by studies of epidemic spreading [6, 7, 19, 33, 85]. Similar to how an infectious disease is transmitted among the population, a piece of information can pass from one individual to another through social connections and "infected" individuals can, in turn, propagate the information to others, possibly generating a full-scale contagion. The *susceptible-infected (SI)* [45,46], *susceptible-infected-recovered (SIR)* [6], and *susceptible-infected-susceptible (SIS)* [7] models are three classical models in epidemiology, in which the infected population grows exponentially until the rate of infection is balanced by the rate of recovery, or the contagion finally dies off when the recovery rate prevails. As another foundation for this field, different models refer to different scenarios in seeking propagation origins. Currently, researchers mainly employ these three epidemic models in rumor source identification:

### 2.2.1 Susceptible-Infected model

In this model, nodes are initially susceptible and can be infected along with the propagation of rumor (Fig. 2.5(A)). Once a node is infected, it remains infected forever. This model focuses on the infection process $S \rightarrow I$, regardless of the recovery process.

### 2.2.2 Susceptible-Infected-Recovered model

Recovery processes are considered in this model (Fig. 2.5(B)). Similarly, nodes are initially susceptible and can be infected along with the propagation. Infected nodes can then be recovered, and never become susceptible again. This model deals with the infection and curing process $S \rightarrow I \rightarrow R$.

### 2.2.3 Susceptible-Infected-Susceptible model

In this model, infected nodes can become susceptible again after they are cured (Fig. 2.5(C)). This model stands for the infection and recovery process $S \rightarrow I \rightarrow S$.

There are also other epidemic models, such as SIRS [99], SEIR [116], MSIR [37], SEIRS [17]. Readers could refer to the work of [113] and [106] for more epidemic models. Future work may consider these models in rumor source identification.

## 2.3 Maximum-Likelihood Estimation

*Maximum-likelihood estimation (MLE)* [Cowan, 1998] is a method of estimating the parameters $\theta$ of a statistical model $M$, given the independent observed data $X = \{x_1, x_2, ..., x_n\}$. Let us assume the probability of observing $x_i$ in the model $M$ given

parameters $\theta$ is $f(x_i|\theta)$. Then the likelihood of having parameters $\theta$ equals to the probability of observing $X$ given $\theta$:

$$\mathcal{L}(\theta|X) = f(X|\theta) = f(x_1|\theta)f(x_2|\theta)...f(x_n|\theta) = \Pi i = 1^n f(x_i|\theta). \tag{2.3.1}$$

$$\log\mathcal{L}(\theta|X) = \sum_{i=1}^{n} \log f(x_i|\theta). \tag{2.3.2}$$

$$\hat{\theta} = \arg\max_\theta \log \mathcal{L}(\theta|X) = \arg\max \sum_{i=1}^{n} \log f(x_i|\theta). \tag{2.3.3}$$

To find the optimal parameter $\hat{\theta}$ which best describes the observed data given the model and thus provides the largest log-likelihood value, we can solve the Eq. (2.3.3) or computationally search for the best solution in the parameter space. This thesis mainly adopts MLE to estimate the probability of a node being a candidate rumor source.

# Chapter 3

# Rumor Source Identification

This chapter provides an extensive literature review on *rumor source identification* by tracing research trends and hierarchically reviewing the contributions along each research line regarding *rumor source identification*.

This chapter consists of four parts. First, we introduce different types of observations on rumor diffusion. Second, for each type of observations, we review the existing approaches and analyze their pros and cons. Third, comparative studies are provided according to various experiment settings and diffusion scenarios. Finally, we summarize the analysis and comparative studies, and conclude the perspective research issues in this area.

## 3.1 Categories of Observations

One of the major premises in rumor source identification is the observation of node states during the propagation process. Diverse observations lead to a great variety of methods. According to the literature, there are three main categories of observations: *complete observations*, *snapshots*, and *sensor observations*. An illustration of these three categories of observations is shown in Fig. 3.1. It is clear that the snapshot and

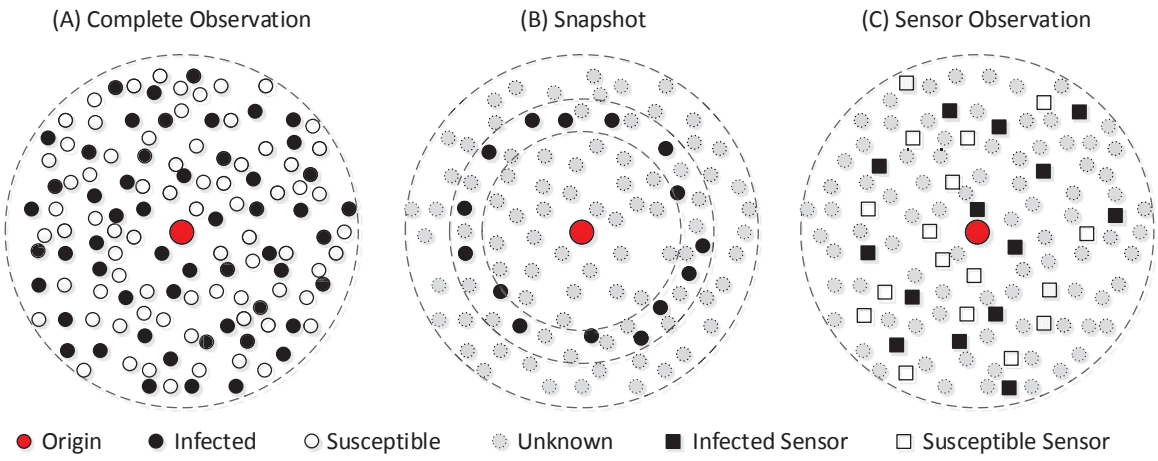



Figure 3.1: Illustration of three categories of observation in networks. (A) Complete observation; (B) Snapshot; (C) Sensor observation.

sensor observation provide much less information for identifying propagation sources compared with the complete observation.

### 3.1.1 Complete Observation

Given a time $t$ during the propagation, *complete observation* presents the exact state for each node in the network at time $t$. The state of a node stands for the node having been infected, recovered, or remaining susceptible. This type of observation provides comprehensive knowledge of a transient status of the network. Through this type of observation, source identification techniques are advised with sufficient knowledge. An example of the complete observation is shown in Fig. 3.1(A).

### 3.1.2 Snapshot Observation

A *snapshot* provides partial knowledge of network status at a given time $t$. Partial knowledge is presented in four forms: (i) nodes reveal their infection status with

probability $\mu$; (ii) we recognize all infected nodes, but cannot distinguish susceptible or recovered nodes; (iii) only a set of nodes were observed at time $t$ when the snapshot was taken; (iv) only the nodes who were infected exactly at time $t$ were observed. An example of the 4-th type of snapshots is shown in Fig. 3.1(B).

### 3.1.3 Sensor Observation

*Sensors* are firstly injected into networks, and then the propagation dynamics over these sensor nodes are collected, including their states, state transition time and infection directions. In fact, sensors also stand for users or computers in networks. The difference between sensors and other nodes in networks is that they are usually monitored by network administrators in practice. Therefore, the sensors can record all details of the rumor propagation over them, and their life can be theoretically assumed to be everlasting during the propagation dynamics. This is different from the mobile sensor devices which may be out of work when their batteries run out. As an example, we show the sensor observation in Fig. 3.1(C).

In the following three sections, we analyze different techniques for source identification and discuss their pros and cons. We classify the source identification methods into three categories in accordance with the three different types of observations in Section 3.1. The taxonomy of current methods is shown in Fig. 3.2.

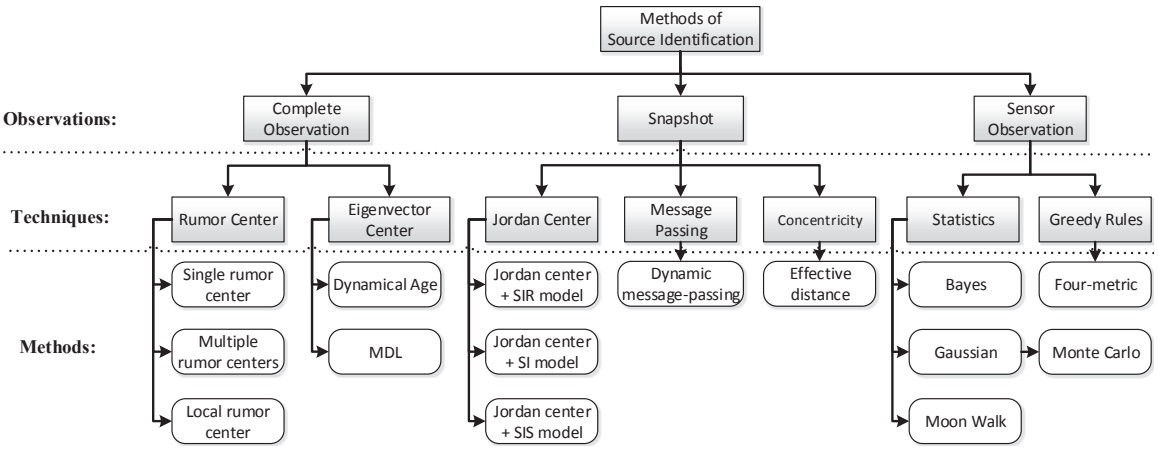



Figure 3.2: Taxonomy of current source identification methods.

## 3.2 Rumor Source Identification Based on Complete Observations

In this section, we summarize the methods of source identification developed under complete observations. There are two main techniques in this category: rumor center and eigenvector center based methods (see Fig. 3.2).

### 3.2.1 Single Rumor Center

Shah and Zaman [95], [96] introduced rumor centrality for source identification. They assume that information spreads in tree-like networks and the information propagation follows the SI model. They also assume each node receives information from only one of its neighbors. Since we consider the complete observations of networks, the source node must be one of the infected nodes. This method is proposed for the propagation of rumors originating from a single source.

***Method*** Assuming an infected node as the source, its rumor centrality is defined

as the number of distinct propagation paths originating from the source. The node with the maximum rumor centrality is called the rumor center. For regular trees, the rumor center is considered as the propagation origin. For generic networks, researchers employ BFS trees to represent the original networks. Each BFS tree corresponds to a probability $\rho$ of a rumor that chooses this tree as the propagation path. In this case, the source node is revised as the one that holds the maximum product of rumor centrality and $\rho$.

***Analysis*** In essence, the method is to seek a node from which the propagation matches the complete observation the best. As proven in [95, 96], the rumor center is equivalent to the closeness center for a tree-like network. However, for a generic network, the closeness center may not equal the rumor center. The effectiveness of the method is further examined by the work in [97]. The authors proved that the rumor center method can still provide guaranteed accuracy when relaxing two assumptions: the exponential spreading time and the regular trees. This method was further explored in the snapshot scenario that nodes reveal whether they have been infected with probability $\mu$ [42]. When $\mu$ was large enough, the authors proved the accuracy of the rumor center method can still be guaranteed. Z. Wang et al. [107] extended the discussion of the single rumor center into a more complex scenario with multiple snapshots. Although snapshot only provided partial knowledge of rumor spreading, the authors proved that multiple independent snapshots could dramatically improve temporally sequential snapshots. The analysis in [107] suggested that the complete observation of rumor propagation could be approximated by multiple independent snapshots.

***Discussion*** There are several strong assumptions far from reality. First, it is

considered on a very special class of networks: infinite trees. Generic networks have to be reconstructed into BFS trees before seeking propagation origins. Second, rumors are implicitly assumed to spread in a *unicast way* (*i.e.*, an infectious node can only infect one of its neighbors at one time step). Third, the infection probability between neighboring nodes is equal to 1. In the real world, however, networks are far more complex than trees, with rumors often spreading in *multicast* or *broadcast* ways, and the infection probability between neighboring nodes differing from each other.

### 3.2.2 Local Rumor Center

Following the assumptions of the single rumor center method, Dong et al. [22] proposed a *local rumor center method* to identify rumor sources. This method designates a set of nodes as suspicious sources. Therefore, it reduces the scale of seeking origins.

***Method*** Dong et al. [22] extended the approaches and results in [95] and [96] to identify the source of propagation in networks. Following the definition of the rumor center, they defined the *local rumor center* as the node with the highest rumor centrality compared to other suspicious infected nodes. The local rumor center is considered as the rumor source.

***Analysis*** For regular trees with every node having degree $d$, the authors analyze the *accuracy* $\gamma$ of the local rumor center method. To construct a regular tree, the degree $d$ of each node should be at least 2. For regular trees, Dong et al. [22] derived the following conclusions. (i) When $d = 2$, the accuracy of the local rumor center method follows $O(1/\sqrt{n})$, where $n$ is the number of infected nodes. Therefore, when $n$ is sufficiently large, the accuracy is close to 0. (ii) When the suspicious set degenerates into the entire network, the accuracy $\gamma$ grows from 0.25 to 0.307 as $d$ increases from 3

to $+\infty$. This means that the minimum accuracy $\gamma$ is 25% and the maximum accuracy is 30.7%. (iii) When the suspicious nodes form a connected subgraph of the network, the accuracy $\gamma$ significantly exceeds $1/k$ when $d = 3$, where $k$ is the the number of suspicious nodes. (iv) When there are only two suspect nodes, the accuracy $\gamma$ is at least 0.75 if $d = 3$, and $\gamma$ increases with the distance between the two suspects. (v) When multiple suspicious nodes form a connected subgraph, the accuracy $\gamma$ is lower than when these nodes form several disconnected subgraphs.

***Discussion*** The local rumor center is actually the node with the highest rumor centrality in the priori set of suspects. The advantage of the local rumor center method is that it dramatically reduces the source-searching scale. However, it has the same drawbacks as the single rumor center method.

### 3.2.3 Multiple Rumor Centers

Luo et al. [58] extended the single rumor center method to identify *multiple sources*. In addition to the basic assumptions, they further assumed the number of sources was known for the method of identifying multiple rumor centers.

***Method*** Based on the definition of rumor centrality for a single node, Luo et al. [58] extended rumor centrality to a set of nodes, which is defined as the number of distinct propagation paths originating from the set. They proposed a two-source estimator to compute the rumor centrality when there were only two sources. For multiple sources, they proposed a two-step method. In the first step, they assumed a set of infected nodes as sources. All infected nodes were divided into different partitions by using the Voronoi partition algorithm [36] on these sources. The single rumor center method was then employed to identify the source in each partition.

In the second step, estimated sources were calibrated by the two-source estimator between any two neighboring partitions. These two steps were iterated until the estimated sources become steady.

***Analysis*** Luo et al. [58] are the first to employ the rumor center method to identify multiple rumor sources. They further investigate the performance of the two-source estimator on geometric trees [96]. The accuracy approximates to 1 when the infection graph becomes large. This method has also been extended to identify multiple sources with *snapshot observations*. Because snapshots only provide partial knowledge about the spreading dynamics of rumors in networks, W. Zang et al. [118] introduce a score-based method to assess the states of other nodes in networks, which indirectly form a complete observation on networks.

***Discussion*** According to the definition of rumor centrality of a set of nodes, we need to calculate the number of distinct propagation paths originating from the node set. It is too computationally expensive to obtain the result. Even though Luo et al. have proposed a two-step method to reduce the complexity, the two-step method still needs $O(N^k)$ computations, where $k$ is the number of rumor sources. This method can hardly be used in the real world, especially for large-scale networks.

### 3.2.4 Minimum Description Length

Prakash et al. [83, 84] proposed a *minimum description length (MDL)* method for source identification. This method is considered for generic networks. They assumed rumor propagation following the SI model.

***Method*** Given an arbitrary infected node as the source node, minimum description length corresponds to the probability of obtaining the infection graph. For generic networks, it is too computationally expensive to obtain the probability. Instead, Prakash et al. [84] introduced an *upper bound of the probability* and detected the origin by maximizing the upper bound. They claimed that to maximize the upper bound is to find the smallest eigenvalue $\lambda_{min}$ and the corresponding eigenvector $u_{min}$ of the *Laplacian matrix* of the infection graph. The Laplacian matrix is widely used in spectral graph theory and has many applications in various fields. This matrix is mathematically defined as $L = D - A$, where $D$ is the diagonal degree matrix and $A$ is the adjacency matrix. According to Prakash et al.'s work in [83, 84], the node with the largest score in the eigenvector $u_{min}$ refers to the propagation source.

***Analysis*** This method can also be used to seek multiple sources. The authors adopted the minimum description length (MDL) cost function [34]. This was used to evaluate the 'goodness' of a node being in the source set. To search the next source node, they first removed the previous source nodes from the infected set. Then, they replayed the process of searching the single source in the remaining infection graph. These two steps were iterated until the MDL cost function stopped decreasing.

***Discussion*** Due to the high complexity in computing matrix eigenvalues, generally $O(N^3)$, the DML method is not suitable for identifying sources in large-scale networks. Moreover, the number of true sources is generally unknown. Further to this, the gap between the upper bound and the real value of the probability has not been studied, and therefore, the accuracy of this method is not guaranteed.

### 3.2.5 Dynamic Age

Fioriti et al. [25] introduced the *dynamic age method* for source identification in generic networks. The assumption for this method is the same as the MDL method.

***Method*** Fioriti et al. took the advantage of the correlation between the eigenvalue and the 'age' of a node in a network. The 'oldest' nodes which were associated to those with largest eigenvalues were considered as the sources of a propagation [119]. Meanwhile, they utilized the dynamical importance of node in [86]. It essentially calculated the reduction of the largest eigenvalue of the adjacency matrix after a node had been removed. A large reduction after the removal of a node implied the node was relevant to the 'aging' of a propagation. By combing these two techniques, Fioriti et al. proposed the concept of *dynamical age* for an arbitrary node $i$ as follows,

$$DA_i = |\lambda_m - \lambda_m^i|/\lambda_m, \tag{3.2.1}$$

where $\lambda_m$ was the maximum eigenvalue of the adjacency matrix, and $\lambda_m^i$ was the maximum eigenvalue of the adjacency matrix after node $i$ was removed. The nodes with the highest dynamic age were considered as the sources.

***Analysis*** This method is essentially different from the previous MDL method. The MDL method is to find the smallest eigenvalues and the corresponding eigenvectors of Lapacian matrices, while the dynamic age method is to find the largest eigenvalues of the adjacency matrix.

***Discussion*** Similar to the MDL method, the dynamic age method is not suitable for identifying sources in large-scale networks due to the complexity of calculating eigenvectors. Moreover, since there is no threshold to determine the oldest nodes, the number of source nodes is uncertain.

## 3.3 Rumor Source Identification based on Snapshots

In the real world, a complete observation of an entire network is hardly possible, especially for large-scale networks. Snapshots are observations closer to reality. It only provides partial knowledge of a propagation in networks. There are three techniques of source identification developed on snapshot: Jordan center, message passing and concentricity based methods (see the taxonomy in Fig. 3.2).

### 3.3.1 Jordan Center

Zhu and Ying [120] proposed the *Jordan center method* for rumor source identification. They assumed rumor propagated in tree-like networks and the propagation followed SIR model. All infected nodes were given, but the susceptible nodes and and recovered nodes were undistinguishable. This method was proposed for single source propagation.

***Method*** Zhu and Ying [120] proposed a sample path based approach to identify the propagation source. An optimal sample path was the one which most likely leaded to the observed snapshot of a network. The source associated with the optimal sample path was proven to be the *Jordan center of the infection graph*. Jordan center was then considered as the rumor source.

***Analysis*** Zhu and Ying [121] further extended the sample path based approach to the heterogeneous SIR model. Heterogeneous SIR model means the infection probabilities between any two neighboring nodes are different, and the recovery probabilities of infected nodes differ from each other. They proved that on infinite trees, the source

node associated with the optimal sample path was also the Jordan center. Moreover, Luo et al. [57, 59] investigated the sample path based approach in the SI and SIS models. They obtained the same conclusion as in the SIR model.

***Discussion*** Similar to rumor center based methods, the Jordan center method is considered on infinite tree-like networks which are far away from real-world networks.

### 3.3.2 Dynamic Message Passing

Lokhov et al. [53] proposed the *dynamic message-passing (DMP) method* by assuming that propagation follows the SIR model in generic networks. Only propagation time $t$ and the states of a set of nodes at time $t$ are known.

***Method*** The DMP method is based on the dynamic equations approach proposed in [43]. Assuming an arbitrary node as the source node, it first estimates the probabilities of other nodes to be in different states at time $t$. Then, it multiplies the probabilities of the observed set of nodes being in the observed states. The source node which can obtain the maximum product is considered the propagation origin.

***Analysis*** The DMP method takes into account the spreading dynamics of the propagation process. This is very different from the previous centrality based methods (e.g., rumor center and Jordan center based methods). Lokhov et al. [53] claimed that the DMP source identification method dramatically outperformed the previous centrality based methods.

***Discussion*** An important prerequisite of the DMP method is that we must know the propagation time $t$. However, the propagation time $t$ is generally unknown. Besides, the computational complexity of this method is $O(tN^2d)$, where $N$ is the number of nodes in a network and $d$ is the average degree of the network. If the

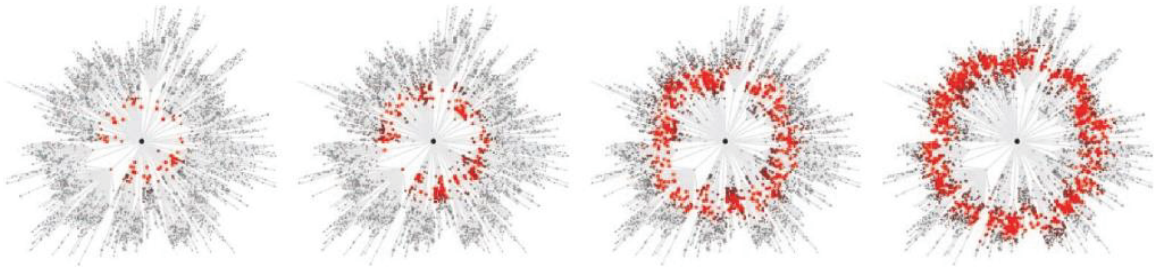


Figure 3.3: Illustration of wavefronts in the shortest path tree v. Readers can refer to the work "The Hidden Geometry of Complex, Network-driven Contagion Phenomena" [12] for the details of the wavefronts.

underlying network is strongly connected, it will be computationally expensive to use the DMP method to identify the propagation source.

### 3.3.3 Effective Distance Based Method

Assuming propagation follows SI model in weighted networks, Brockmann and Helbing [12] proposed an *effective distance based method* for rumor source identification. This method is considered in another case of snapshot−*wavefront*.

***Method*** Brockmann and Helbing [12] first proposed a new concept, *effective distance*, to represent the propagation process. The effective distance from node $n$ to a neighboring node $m$, $d_{mn}$, is defined as

$$d_{mn} = 1 - \log P_{mn}, \tag{3.3.1}$$

where $P_{mn}$ is the fraction of a propagation with destination $m$ emanating from $n$. From the perspective of a chosen source node $v$, the set of shortest paths in terms of effective distance to all other nodes constitutes a shortest-path tree $\Psi_v$. Brockmann and Helbing [12] empirically obtain that the propagation process initiated from node

$v$ on the original network can be represented as *wavefronts* on the shortest-path tree $\Psi_v$. To illustrate this process, a simple example is shown in Fig. 3.3 (refers to [12]). According to the propagation process of wavefronts, the spreading concentricity can only be observed from the perspective of the propagation source. Then, the node, which has the minimum standard deviation and mean of effective distances to the nodes in the observed wavefront, is considered as the source node.

***Analysis*** The information propagation process in networks is complex and network-driven. The combined multiscale nature and intrinsic heterogeneity of real-world networks make it difficult to develop an intuitive understanding of these processes. Brockmann and Helbing [12] reduce the complex *spatiotemporal patterns* to a simple wavefront propagation process by using effective distance.

***Discussion*** To use the effective distance based method for source identification, we need to compute the shortest distances from any suspicious source to the observed infected nodes. This leads to high computational complexity, especially for large-scale networks.

## 3.4 Rumor Source Identification based on Sensor Observations

In the real world, a further strategy to identify propagation sources is based on sensors in networks. The sensors report the direction in which information transmits through them, and the time at which the information arrives at them. There are two techniques developed in this category: statistics and greedy rules (see the taxonomy in Fig. 3.2).

### 3.4.1 Gaussian Estimator

Assuming propagation follows SI model in tree-like networks, Pinto et al. [82] proposed a *Gaussian method* for single source identification. They also assume there is a deterministic propagation time on each edge, which are independent and identically distributed with *Gaussian distribution*.

***Method*** This method is divided into two steps. In the first step, they reduce the scale of seeking origins. According to the direction in which information arrived at the sensors, it uniquely determines a subtree $T_a$. The subtree $T_a$ is guaranteed to contain the propagation origin [82]. In the second step, they use the following Gaussian technique to seek the source in $T_a$. On the one hand, given a sensor node $o_1$, they calculate the 'observed delay' between $o_1$ and the other sensors. On the other hand, assuming an arbitrary node $s \in T_a$ as the source, they calculate the 'deterministic delay' for every sensor node relative to $o_1$ by using the deterministic propagation time of the edges. The node, which can minimize the distance between the 'observed delays' and the 'deterministic delays' of sensor nodes, is considered as the propagation source.

***Analysis*** This method is considered on tree-like networks. For generic networks, Pinto et al. [82] assume that information spreads along the BFS tree, and search rumor source in the BFS trees. It is improved by combining community recognition techniques [82] in order to reduce the number of deployed sensors in networks. By choosing the nodes between communities and with high betweenness values as sensors, A. Louni et al. [54] reduce 3% fewer sensors than the original method [82].

***Discussion*** For generic networks, the Gaussian estimator is of complexity $O(N^3)$. Again, it is too computationally expensive to use this method for large-scale networks.

### 3.4.2 Monte Carlo Method

Agaskar and Lu [1] proposed a *fast Monte Carlo method* for source identification in generic networks. They assume propagation follows the heterogeneous SI model in which the infection probabilities between any two neighboring nodes are different. In addition, the observation of sensors is obtained in a fixed time window.

***Method*** This method consists of two steps. In the first step, assuming an arbitrary node as the source, they introduce an alternate representation for the infection process initiated from the source. The alternate representation is derived in terms of the infection time of each edge. Based on the alternate representation, they sample the infection time for each sensor. In the second step, they compute the gap between the observed infection time and the sampled infection time of sensors. They further use the Monte Carlo approach to approximate the gap. The node which can minimize the gap is considered as the propagation origin.

***Analysis*** The computational complexity of this method is $O(LN\log(N)/\varepsilon)$, where $L$ is the number of sensor nodes, and $\varepsilon$ is the assumed error. The complexity is lower than other source identification methods, which are normally $O(N^2)$ or even $O(N^3)$.

***Discussion*** When sampling infection time for each edge, Agaskar and Lu [1] assume that information always spreads along *the shortest paths* to other nodes. However, in the real world, information generally reaches other nodes by a random walk. Therefore, this method may not be suitable for other propagation schemes, such as *random spreading* or *multicast spreading*.

### 3.4.3 Bayesian Estimator

Distinguished from the DMP method which adopts the message-passing propagation model (see Section 3.3.2), F. Altarelli et al. [5] proposed using the *Bayesian belief propagation model* to compute the probabilities of each node being at any state. This method can work with different types of observations and in different propagation scenarios, however guaranteed accuracy is only obtained in tree-like networks.

***Method*** This method consists of three steps. The propagation of rumors are first presented by SI, SIR or other isomorphic models [106]. Second, given an observation on the infection of a network, either through a group of sensors or a snapshot at an unknown time, the belief propagation equations are derived for the posterior distribution of past states on all network nodes. By constructing a factor graph based on the original network, these equations provide the exact computation of posterior marginal in the models. Third, belief propagation equations are iterated with time until they converge. Nodes are then ranked according to the posterior probability of being the source.

***Analysis*** This method provides the exact identification of source in tree-like networks. This method is also effective for synthetic and real networks with cycles, both in a static and a dynamic context, and for more general networks, such as DTN [123]. This method relies on belief propagation model in order to be used with different observations and in various scenarios.

***Discussion*** The accuracy of this method can not be guaranteed other than in tree-like networks. Particularly for dynamically evolving networks [100], the average success rate is only $0.53 \pm 0.06$ and the average error reaches $0.76 \pm 0.23$.

### 3.4.4 Moon-walk Method

Xie et al. proposed a *post-mortem technique* on traffic logs to seek the origin of a worm (a kind of computer virus) [114]. There are four assumptions for this technique. First, it focuses on the scanning worm [109]. This kind of worm spreads on the Internet by making use of OS vulnerabilities. Victims will proceed to scan the whole IP space for vulnerable hosts. Famous examples of this kind of worm includes Code Red [125] and Slammer [66]. Second, logs of infection from sensors cover the majority of the propagation processes. Third, the worm propagation forms a tree-like structure from its origin. Last, the attack flows of a worm do not use spoofed source IP addresses.

***Method*** Based on traffic logs, the network communication between end-hosts are modeled by a directed host contact graph. Propagation paths are then created by sampling edges from the graph according to the time of corresponding logs. The creation of each path stops when there is no contiguous edge within $\Delta t$ seconds to continue the path. As the sampling is performed, a count is kept of how many times each edge from the contact graph is traversed. If the worm propagation follows a tree-like structure, the edge with maximum count will most likely be the top of the tree. The start of this directed edge will be considered as the propagation source.

***Analysis*** There are several issues on this technique that need to be further analyzed. First, it is reasonable to assume worm do not use the IP spoof technique. In the real world, the overwhelming majority of worm traffic involved in propagation is initiated by victims instead of the original attacker. Spoofed IP addresses would only decrease the number of successful attacks without providing further anonymity to the attacker. Second, IP trace back techniques [93] are related to Moonwalk and other methods discussed in this article. However, trace back on its own is not sufficient to

track worms to their origin, as trace back only determines the true source of the IP packets received by a destination. In an epidemic attack, the source of these packets is almost never the origin of the attack, but just one of the many infected victims. The methods introduced in this article are still needed to find the hosts higher up in the propagation casual trees. Third, this method relies only on traffic logs. This feature benefits itself on its ability to work without any a priori knowledge about the worm attack.

***Discussion*** Nowadays, the number of scanning worms has largely decreased due to advances in OS development and security techniques [112]. Therefore, the usage of Moonwalk, which can only seek the propagation origin of the scanning worm, is largely limited. Moreover, a full collection of infection logs is hardly achieved in the real world. Finally, current computer viruses are normally distributed by Botnet [124]. Moonwalk, which can only seek single origin, may not be helpful in this scenario.

### 3.4.5 Four-metric Method

Seo et al. [94] proposed a *four-metric source estimator* to identify single source node in directed networks. They assume propagation follows the SI model. The sensor nodes who transited from susceptible states to infected states are regarded as positive sensors. Otherwise, they are considered as negative sensors.

***Method*** Seo et al. [94] use the intuition that the source node must be close to the positive sensor nodes, but far away from the negative sensor nodes. They propose four metrics to locate the source. First, they find out a set of nodes which are reachable to all positive sensors. Second, they filter the set of nodes by choosing the ones with the minimum sum of distances to all positive sensor nodes. Third, they further choose the

nodes that are reachable to the minimum number of negative sensor nodes. Finally, the node which satisfies all of the above three metrics and has the maximum sum of distances to all negative sensor nodes is considered as the source node.

***Analysis*** Seo et al. [94] studied and compared different methods of choosing sensors, such as randomly choosing (*Random*), choosing the nodes with high betweenness centrality values ($BC$), choosing the nodes with a large number of incoming edges ($NI$), and choosing the nodes which are at least d hops away from each other ($Dist$). Different sensor selection methods produce different sets of sensor nodes, and have different accuracies in source identification. They show that the $NI$ and $BC$ sensor selection methods outperform the others.

***Discussion*** For the four-metric source estimator, it needs to compute the shortest paths from the sensors to any potential source. Generally, the computational complexity is $O(N^3)$. It is too computationally expensive to use this method.

## 3.5 Comparative Study

In order to have a numerical understanding of the existing methods of source identification, we examine the methods under different experiment environments. Furthermore, we analyze potential impact factors on the accuracy of source identification. We test the methods on both synthetic and real-world networks. All the experiments were conducted on a desktop computer running Microsoft Windows7 with 2 CPUs and 4G memory. The implementation was done in Matlab2012.

For each category of observation, we examined one or two typical source identification methods. In total, five methods were examined. For complete observation, we tested the *rumor center method* and the *dynamic-age method*. We also tested the

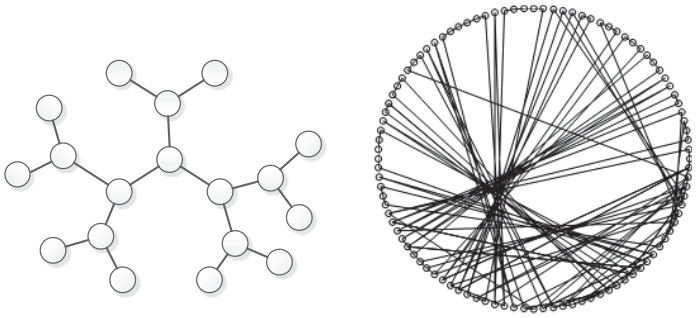
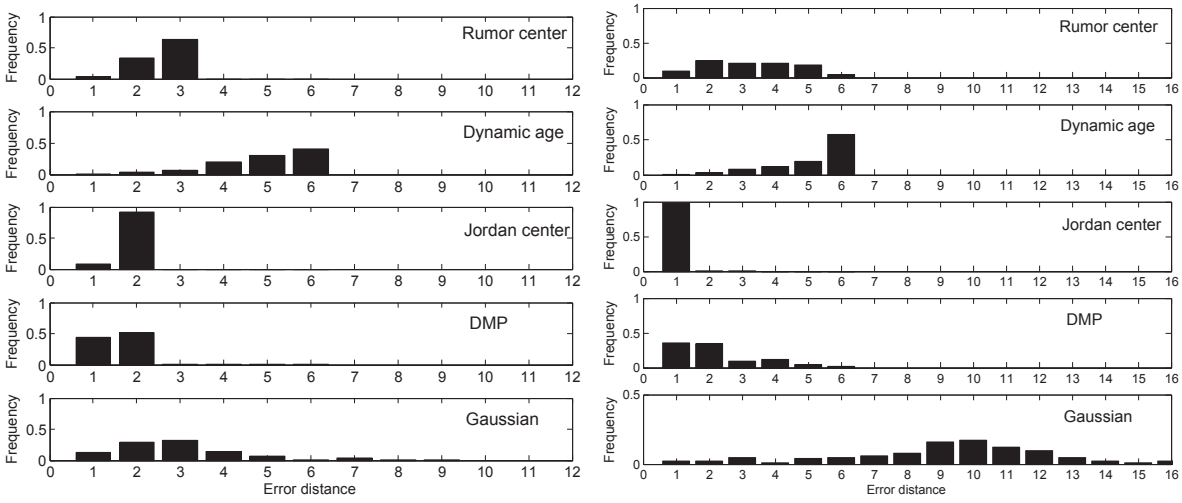



Figure 3.4: Sample topologies of two synthetic networks. (A) 3-regular tree; (B) small-world network.

*Jordan center method* and the *DMP method* for snapshots of networks. The *Gaussian source estimator* was examined for sensor observation. In the experiments, we typically choose infection probability ($q$) to be 0.75 and recovery probability ($p$) to be 0.5. We randomly choose a node as a source to initiate a propagation, and then average the *error distance* $\delta$ between the estimated sources and the true sources by 100 runs.

### 3.5.1 Comparison on Synthetic Networks

In this subsection, we first compare the performance of different source identification methods on *synthetic networks*. Then, we study three potential impact factors (network topology, propagation scheme and infection probability) on the accuracies of the methods.




(A) 4-regular tree

(B) Small-world network

Figure 3.5: Crosswise comparison of existing methods on two synthetic networks.

### Crosswise Comparison

We conducted experiments on two synthetic networks: *a regular tree* [95] and *a small-world network* [108]. Fig. 3.4(A) and Fig. 3.4(B) show example topologies of a 3-regular tree and a small-world network. Fig. 3.5(A) shows the frequency of error distances $\delta$ of different methods on a *4-regular tree*, respectively. We can see that, the sources estimated by the *DMP method* and the *Jordan center method* are the closest to the true sources, with an average of 1.5-2 hops away. The rumor center method and the Gaussian method estimate the sources with an average of 2-3 hops away from the true sources. The sources estimated using the dynamic age method were the farthest away from the true sources. Fig. 3.5(B) shows the performances of different methods on a *small-world network*. It is clear the *Jordan center method* outperforms the others, with estimated sources around 1 hop away from the true sources. The DMP method also exposes good performances by showing estimated

sources are an average of 1-2 hops away from the true sources. The dynamic age method and Gaussian method have the worst performance.

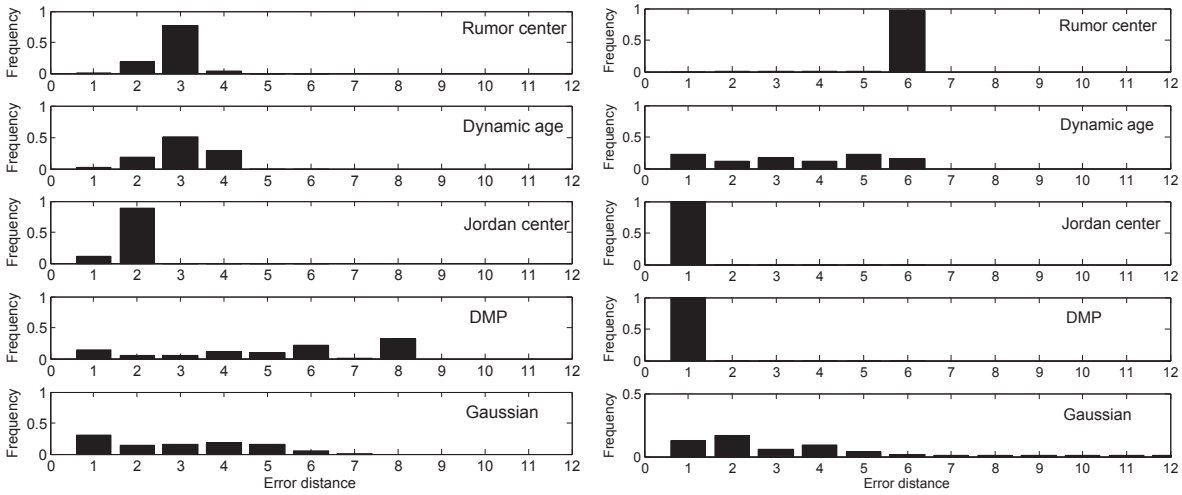
***Numerical Results***: From the experiment results on the regular tree and small-world network, we can see that the DMP method and the Jordan center method have better performance than the other methods.

### The Impact of Network Topology

In Sections 3.2-3.4, we know that some existing methods of source identification are considered on *tree-like networks*. In the previous subsection, we have shown the results of methods implemented on *regular trees* and *small-world networks*. In order to analyze the impact of network topology on the methods, we introduce another two different network topologies: *random trees* and *regular graphs*: We further conduct performance evaluation on these two topologies.

Fig. 3.6(A) shows the experiment results of methods on a random tree. It is clear the Jordan center method has the best performance, with estimated sources around 2 hops away from the true sources. The rumor center method and the dynamic age method show similar performance, with estimated sources around 3 hops away from the true sources. The DMP method and the Gaussian method have the worst performance. Fig. 3.6(B) shows the experiment results of methods on a regular graph. It shows that sources estimated by using the Jordan center method and the DMP method were the closest to the true sources. The sources estimated by the rumor center method were the farthest from true sources. The dynamic age method and the Gaussian method also show poor performance in this scenario.

***Numerical Results***: From the experiment results on the four different network




(A) Random tree (B) Small-world network

Figure 3.6: The impact of network topologies.

topologies, we can see the source identification methods are sensitive to network topology.

### The Impact of Propagation Scheme

From Sections 3.2-3.4, we know that some existing methods of source identification are based on the assumption that information propagates along the *BFS trees in networks*. This means propagation follows the *broadcast scheme*. However, in the real world, propagation may follow various propagation schemes. We focus on three most common propagation schemes: *snowball*, *random walk* and *contact process* [15]. Their definitions are given below.

- Random Walk: A node can deliver a message randomly to one of its neighbors.
- Contact Process: A node can deliver a message to a group of its neighbors that have expressed interest in receiving the message.

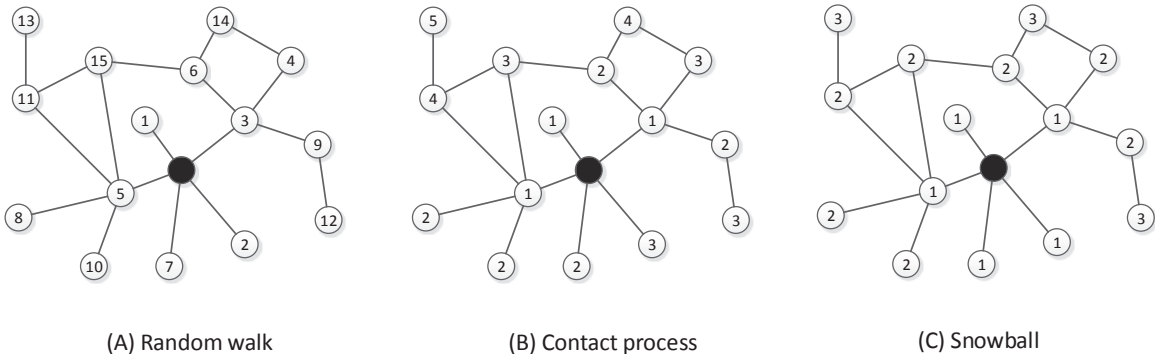
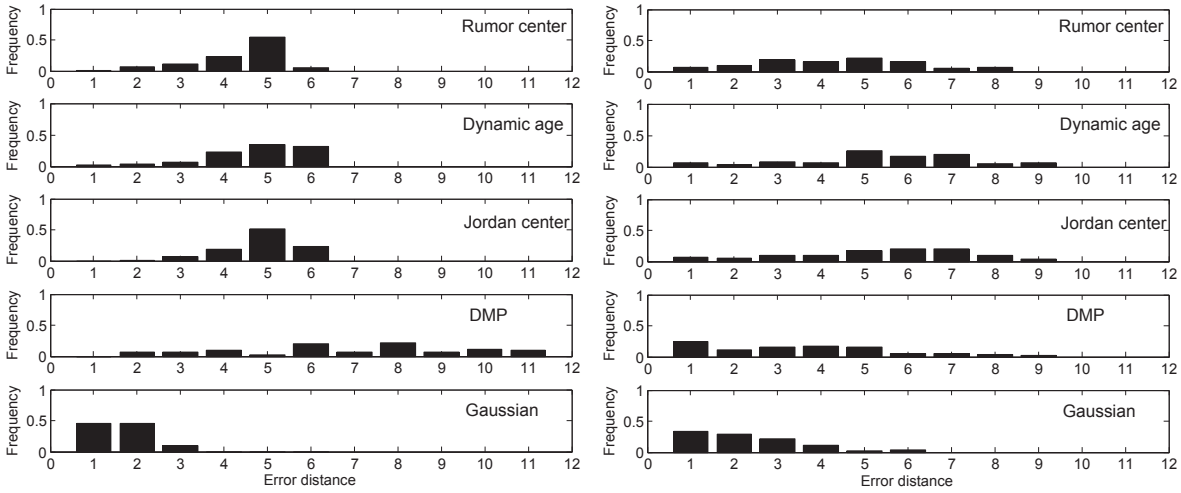



Figure 3.7: Illustration of different propagation schemes. The black node stands for the source. The numbers indicate the hierarchical sequence of nodes getting infected.

- Snowball Spreading: A node can deliver a message to all of its neighbors.

An illustration of these three propagation schemes is shown in Fig. 3.7. We examine different propagation schemes on both regular trees and small-world networks.

Fig. 3.8(A) shows the experiment results of the methods with propagation following the *random-walk propagation scheme* on a *4-regular tree*. It is clear the Gaussian source estimator outperforms the others, with estimated sources around 1-2 hops away from the true sources. The performances of the rumor center method, the dynamic age method and the Jordan center method are similar to each other, with estimated sources around 5 hops away from the true sources. The DMP method has the worst performance. Fig. 3.9(A) shows experiment results of the methods with propagation following the *contact-process propagation scheme* on a *4-regular tree*. It is clear the results in Fig. 3.8(A) and Fig. 3.9(A) are similar to each other. This means the methods have similar performances on both the random-walk and contact-process propagation schemes. Fig. 3.10(A) shows the experiment results of the methods with



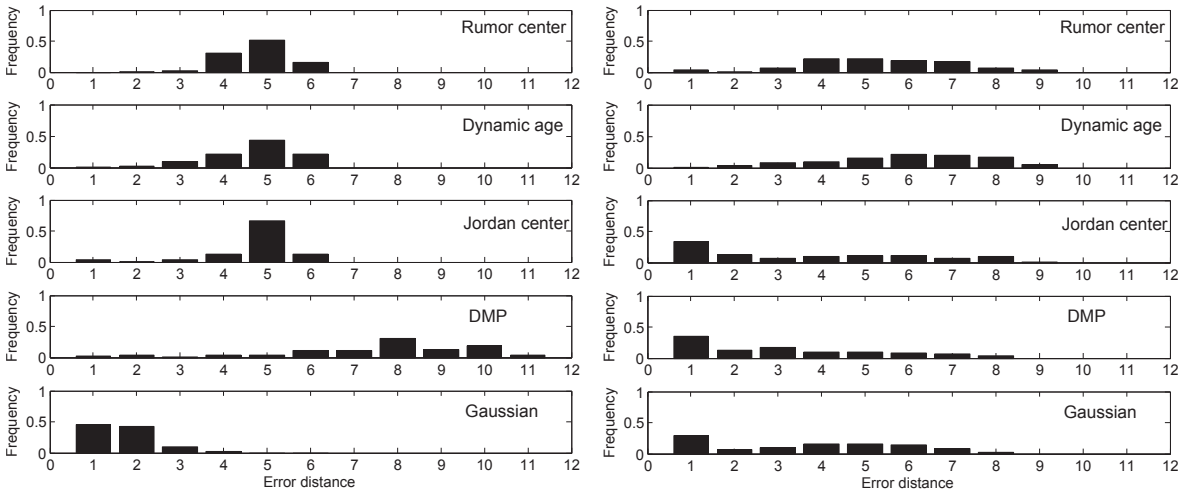

(A) 4-regular tree

(B) Small-world network

Figure 3.8: The impact of propagation schemes: random-walk scheme.

propagation following the *snowball propagation scheme* on a *4-regular tree*. The results show a big difference from the results of the previous two propagation schemes. The DMP method and the Jordan center method outperformed the others, with estimated sources around 1-2 hops away from the true sources. The rumor center method and the Gaussian method also showed good performances, with estimated sources around 2-3 hops away from the true sources. The dynamic age method had the worst performance.

The experiment results of the methods with propagation following different propagation schemes on a *small-world network* are shown in Fig. 3.8(B), Fig. 3.9(B) and Fig. 3.10(B). The results are dramatically different from the results on the 4-regular tree. From Fig. 3.9 we can see the Gaussian source estimator obtains the best performance, followed by the DMP method. The rumor center method, the dynamic age

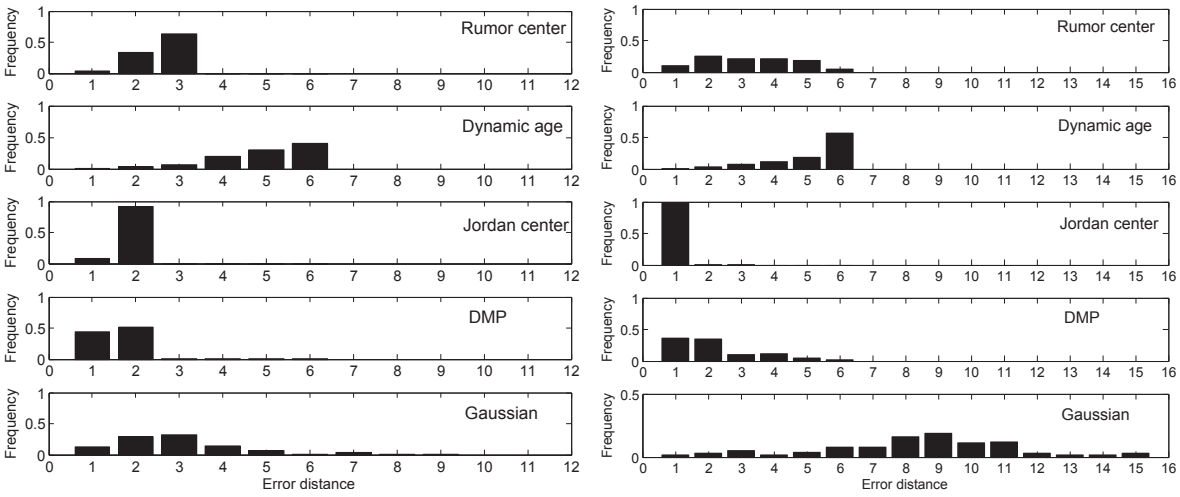



(A) 4-regular tree (B) Small-world network

Figure 3.9: The impact of propagation schemes: contact-process scheme.

method and the Jordan center method show identifying sources by randomly choosing. From Fig. 3.9(B), it is clear the Jordan center method, the DMP method and the Gaussian method show similar performances. These three methods outperform the others. From Fig. 3.10(B) we can see the Jordan center method outperforms the others, with estimated sources around 1 hop away from the true sources. The sources estimated using the DMP method are around 1-2 hops away from the true sources. The Gaussian source estimator has the worst performance.

***Numerical Results***: From the experiment results, we see the source identification methods are also sensitive to propagation schemes. The methods of source identification show better performance when propagation follows the snowball propagation scheme rather than the random-walk or contact-process propagation schemes.




(A) 4-regular tree (B) Small-world network

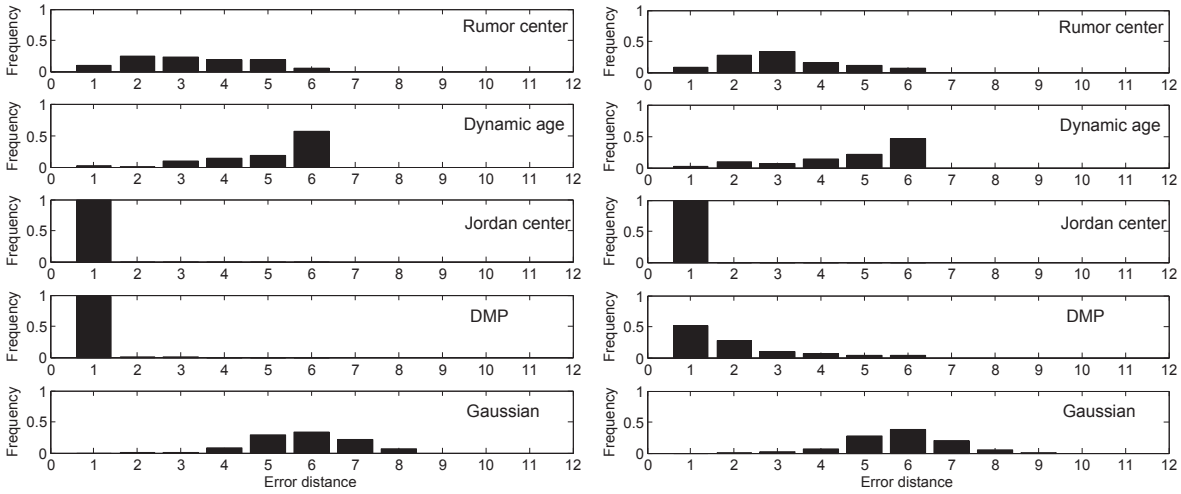
Figure 3.10: The impact of propagation schemes: snowball scheme.

### The Impact of Infection Probability

In this subsection, we will analyze the impact of *infection probability* on the accuracy of source identification. We set the infection probability from 0.5 to 0.95.

The experiment results are shown in Fig. 3.11(A) and Fig. 3.11(B). From these figures, we can see that the rumor center method have similar performances when we change the infection probability. The same phenomenon happens on the dynamic age method, the Jordan center method and the Gaussian methods. The DMP method performs best when infection probability q is equal to 0.5. The accuracy declines when q increases to 0.95. Among the experiment results, the Jordan center method and the DMP method outperform the other methods, with estimated sources around 1 hop away from the true sources. The dynamic age method and the Gaussian method have the worst performance.

***Numerical Results***: From the experiment results, we can see only the DMP




(A) $q = 0.5$ (B) $q = 0.95$

Figure 3.11: The impact of infection probability.

method is sensitive to the infection probability and performs better when the infection probability is lower. The other methods show slightly difference in their performance when applied with various infection probabilities.

### 3.5.2 Comparison on Real-world Networks

In this subsection, we examine the methods of source identification on *two real-world networks*. The first one is an *Enron email network* [41]. This network has 143 nodes and 1,246 edges. On average, each node has 8.71 edges. Therefore, the Enron email network is a dense network. The second is a *power grid network* [3]. This network has 4,941 nodes and 6,594 edges. On average, each node has 1.33 edges. Therefore, the power grid network is a sparse network. Sample topologies of these two real-world networks are shown in Fig. 3.12.

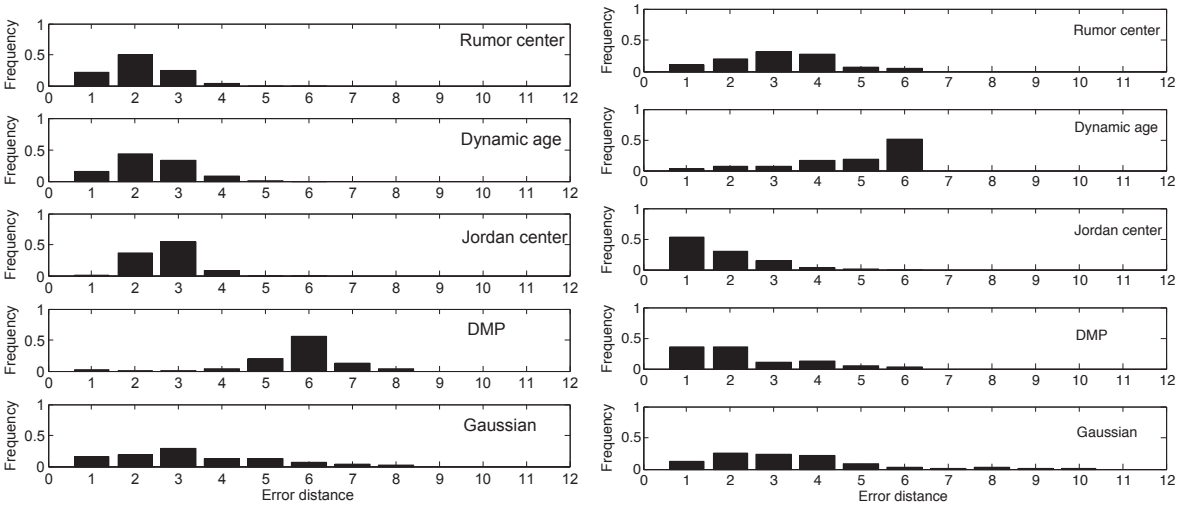
Fig. 3.13(A) shows the frequency of error distance $\delta$ of different methods on

(A) Eron email network (B) Power grid network

Figure 3.12: Sample topologies of two real-world networks.

the Enron email network. We can see the rumor center method, the Jordan center method and the dynamic age method outperform the others. The DMP method has the worst performance. The Enron email network is a small and dense network, complete observation of this network is reasonable and executable, and the identification accuracy is also acceptable. Fig. 3.13(B) shows the experiment results on the power grid network. It is clear the Jordan center method and the DMP method outperform the others, with estimated sources around 1-2 hops away from the true sources. The rumor center method and the Gaussian method show similar performance, with estimated sources around 2-4 hops away from the true sources. The dynamic age method has the worst performance.

***Numerical Results***: From the experiment results, we can see the accuracies of the methods are greatly different between these two real-world networks. For the Enron email network, the rumor center method and the dynamic age method outperform the other methods, while the DMP method has the worst performance. However, for the power grid network, the DMP method and the Jordan center have




(A) Enron email (B) Power grid

Figure 3.13: Source identification methods applied on real networks.

the best performance

### 3.5.3 Summary

We summarize the source identification methods in this subsection. Based on the content in Sections 3.2-3.4, it is clear that current methods rely on either the topological centrality measures or the measures of the distance between the observations and mathematical estimations of the propagation.

In Table 3.1, we collect seven features from the methods discussed in this article. A detailed summary on each feature is elaborated as follows:

1. Topology: As shown in Table 3.1, a significant part the focus for current methods is tree-like topology. These methods can deal with generic network topologies

by using the BFS technique to reconstruct generic networks into trees. According to comparative studies in Section 3.5, methods on different topologies show a great variety of accuracy in seeking origins.

2. Observation: Based on the analysis in Sections 3.2-3.5, the category of observation is not a deterministic factor on the accuracy of source identification. The accuracy of each method varies according to the different conditions and scenarios. In the real world, complete observation is generally difficult to achieve. Snapshot and sensor observation are normally more realistic.

3. Model: The majority of methods employ SI model to present the propagation dynamics of risks. The SI model only considers the susceptible and infected states of nodes regardless of the recovery process. The extension to SIR/SIS will increase the complexity of source identification methods. Jordan center and Monte Carlo method is based on SIR/SIS model. In particular, the Bayesian source estimator can be used in scenarios with various propagation models as the belief propagation approach can estimate the probabilities of node states under various conditions.

4. Source: Most methods focus on single source identification. The multi-rumor center method and eigenvector center method can be used to identify multiple sources. However, these two methods are too computationally expensive to be implemented. In the real world, risks are normally distributed from multiple sources. For example, attackers generally employ a botnet which contains thousands of victims to help spread the computer virus [8, 28]. For source identification, these victims are the propagation origins.

5. Probability: For simplicity, earlier methods consider the infection probabilities to be identical among the edges in networks. Later, most methods are extended to varied infection probabilities among different edges. Noticeably, this extension makes source identification methods more realistic.

6. Time Delay: Only the methods under sensor observations consider time delay for edges. Accurate time delay of risks is an important factor in the propagation [18]. It is important to consider the time delay in source identification techniques.

7. Complexity: Most current methods are too computationally expensive to quickly capture the sources of propagation. The complexity ranges from $O(N\log N/\varepsilon)$ to $O(N^k)$. In fact, the complexity of methods dominates the speed of seeking origins. Quickly identifying propagation sources in most cases is of great significance in the real world, such as capturing the culprits of rumors. Future work is needed to improve the identification speed.

Table 3.1: Summary of Current Source Identification Methods.

| | Topology | Observation | Model | Number of Sources | Infection Probability | Time Delay | Complexity |
|---|---|---|---|---|---|---|---|
| Single rumor center | Tree | Complete | SI | Single | HM/HT | Constant | $O(N^2)$ |
| Local rumor center | Tree | Complete | SI | Single | HM | Constant | $O(N^2)$ |
| Multi rumor centers | Tree | Complete | SI | Multiple | HM | Constant | $O(N^k)$ |
| Eigenvector center | Generic | Complete | SI | Multiple | HM | Constant | $O(N^3)$ |
| Jordan center | Tree | Snapshot | SI(R/S) | Single | HM/HT | Constant | $O(N^3)$ |
| DMP | Generic | Snapshot | SIR | Single | HT | Constant | $O(t_0N^2d)$ |
| Effective distance | Generic | Snapshot | SI | Single | HT | Constant | $O(N^3)$ |
| Gaussian | Tree | Sensor | SI | Single | HT | Variable | $O(N^3)$ |
| Monte Carlo | Generic | Sensor | SIR | Single | HT | Variable | $O(N\log N/\varepsilon^2)$ |
| Four-metrics | Generic | Sensor | SI | Single | HT | Variable | $O(N^3)$ |

HM and HT represent homogeneous and heterogeneous, respectively.

# Chapter 4

# Rumor Source Identification in Time-varying Networks

## 4.1 Introduction

This chapter focuses on identifying rumor sources in *time-varying networks*, particularly time-varying social networks. The proposed method can also be applied in other types of time-varying networks. Rumor spreading in social networks has long been a critical threat to our society [79]. Nowadays, with the development of mobile devices and wireless techniques, the *temporal characteristic* of social networks (*time-varying social networks*) has deeply influenced the dynamic information diffusion process occurring on top of them [87]. The ubiquity and easy access of time-varying social networks not only promote the efficiency of information diffusion but also dramatically accelerate the speed of *rumor spreading* [44, 104].

For either forensic or defensive purposes, it has always been a significant work to identify the source of rumors in time-varying social networks [21]. However, the existing techniques for rumor source identification generally require *firm connections*

*between individuals* (*i.e.*, *static networks*), so that administrators can trace back along the determined connections to reach the diffusion sources. For example, many methods rely on identifying *spanning trees* in networks [58, 95, 107], then the roots of the spanning trees are regarded as the rumor sources. The firm connections between users are the premise of constructing spanning trees in these methods. Some other methods detect rumor sources by measuring *node centralities*, such as degree, betweenness, closeness, and eigenvector centralities [82, 120]. The individual who has the maximum centrality value is considered as the rumor source. All of these centrality measures are based on static networks. Time-varying social networks, where the involved users and interactions always change, have led to *great challenges* to the traditional rumor source identification techniques.

In this chapter, a novel source identification method is proposed to overcome the challenges, which consists the following three steps.

(i) To *represent a time-varying social network*, we reduce it to a sequence of static networks, each aggregating all edges and nodes present in a time-integrating window. This is the case, for instance, of rumors spreading in Bluetooth networks, for which the fine-grained temporal resolution is not available, whose spreading can be studied through different integrating windows $\Delta t$ (*e.g.*, $\Delta t$ could be minutes, hours, days or even months). In each integrating window, if users did not activate the Bluetooth on their devices (*i.e.*, *offline*), they would not receive or spread the rumors. If they moved out the bluetooth coverage of their communities (*i.e.*, *physical mobility*), they would not receive or spread the rumors.

(ii) Similar to the detective routine in criminology, *a small set of suspects* will

be identified by adopting a reverse dissemination process to narrow down the scale of the source seeking area. The reverse dissemination process distributes copies of rumors reversely from the users whose states have been determined based on various observations upon the networks. The ones who can simultaneously receive all copies of rumors from the infected users are supposed to be the suspects of the real sources.

(iii) To *determine the real source from the suspects*, we employ a microscopic rumor spreading model to analytically estimate the probabilities of each user being in different states in each time window. Since this model allows the time-varying connections among users, it can feature the dynamics of each user. More specifically, assuming any suspect as the rumor source, we can obtain the probabilities of the observed users to be in their observed states. Then, for any suspect, we can calculate the maximum likelihood (ML) of obtaining the observation. The one who can provide the maximum ML will be considered as the real rumor source.

The major *contribution* of this chapter is three-fold.

- We adopt a reverse dissemination method to *narrow the scale* of the source seeking area. Compared with the previous methods which scan the entire network, our proposed method significantly promotes the *efficiency* of source identification.

- We introduce a novel ML-based algorithm that can *overcome the connection-always-changing challenge* through a novel rumor spreading model in time-varying social networks.

- Experiment results show *significant advantages* of our method in the identification of rumor sources, the estimation of spreading time, and the prediction of infection scale of rumors.

The rest of this chapter is organized as follows. We introduce the preliminary knowledge of source identification in Section 4.2. Section 4.3 presents the details of the reverse dissemination method. We elaborate upon the ML-based algorithm in Section 4.4, followed by Section 4.5 which shows a series of evaluations on our method. Section 4.6 concludes some remarks in this chapter.

## 4.2 Time-varying Social Networks

In this section, we introduce the primer for rumor source identification in *time-varying social networks*, including the features of time-varying social networks, the state transition of users when they hear a rumor, and the categorization of partial observations in time-varying social networks.

### 4.2.1 Time-varying Topology

The essence of social networks lies in its *time-varying nature*. For example, the neighborhood of individuals moving over a geographic space evolves over time (*i.e.*, ***physical mobility***), and the interaction between the individuals appears and disappears in online social networks (*i.e.*, ***online/offline***) [87]. Time-varying social networks are defined by an ordered stream of interactions between individuals. In other words, as time progresses, the interaction structure keeps changing. Examples can be found in both face-to-face interaction networks [13], and online social networks [104]. The

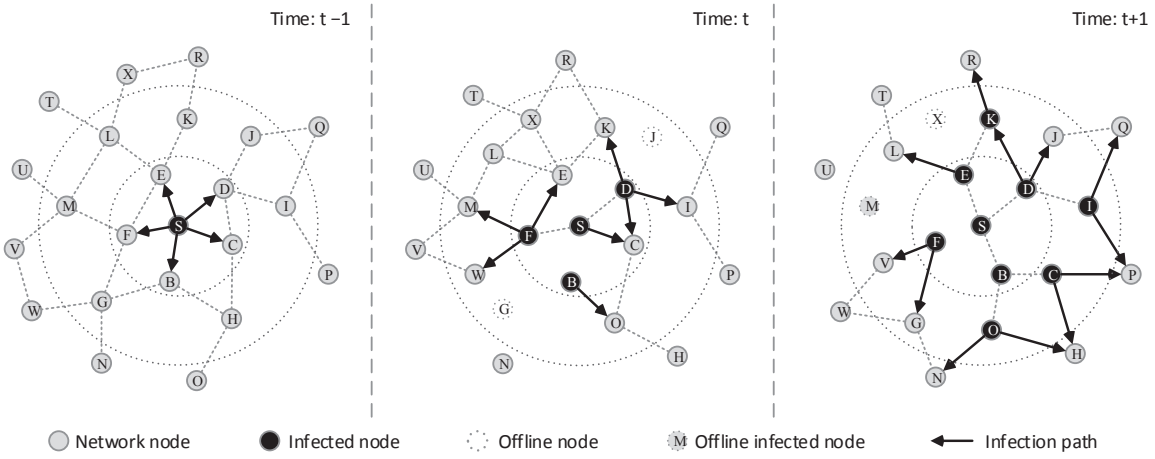



Figure 4.1: Example of a rumor spreading in a time-varying network. The random spread is located on the black node, and can travel on the links depicted as line arrows in the time windows. Dashed lines represent links that are present in the system in each time window.

temporal nature of such networks has a deep influence on information spreading on top of them. Indeed, the spreading of rumors is affected by duration, sequence, and concurrency of contacts among people.

In this work, we reduce time-varying networks to a series of static networks by introducing a time-integrating window. Each integrating window aggregates all edges and nodes present in the corresponding time duration. In Fig. 4.1, we show an example to illustrate the time-integrating windows. In the time window $t-1$ (or, at time $t-1$), a rumor started to spread from node $S$ who had interaction with 5 neighbors in this time window. In the next time window $t$, nodes $B$, $D$ and $F$ were successfully infected. In this time window, we notice that node $O$ moved next to $B$ (*i.e.*, *physical mobility*), and node $G$ had no interaction with its neighbors (*i.e.*, *offline*). Other examples of physical mobility or online/offline status of nodes can be found in the time window $t+1$.

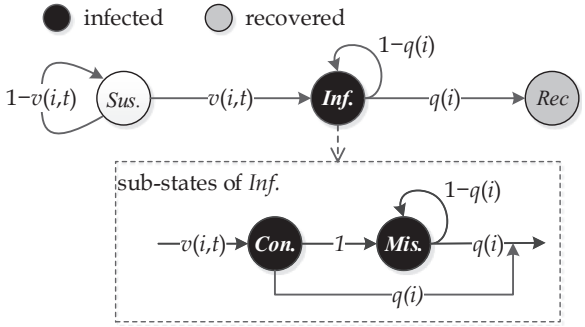



Figure 4.2: State transition of a node in rumor spreading model.

### 4.2.2 Security States of Individuals

For the convenience of description, we borrow the notions from *epidemiology* to describe the spreading of rumors in time-varying social networks [126]. We say a user is *infected* when he/she accepts the rumors, and an infected user is *recovered* if he/she abandons the rumors. In this chapter, we adopt the classic susceptible-infected-recovered (SIR) scheme to present the infection dynamics of each user. Fig. 4.2 shows the state transition graph of an arbitrary user in this model. Every user is initially susceptible (*Sus.*). They can be infected (*Inf.*) by their neighbors with probability $v(i,t)$, and then recover (*Rec.*) with probability $q(i)$. Rumors will be spread out from infected users to their social neighbors until they get recovered. There are also many other models of rumor propagation, including the SI, SIS, SIRS models [60, 67, 117]. In present work, we adopt the SIR model because it can reflect the state transition of users when they hear a rumor, from being susceptible to being recovered. Generally, people will not believe the rumor again after they know the truth. Therefore, recovered users will not transit their states any more. For other propagation models, readers can refer to Section 4.6 for further discussion.

To more precisely describe node states under different types of observations, we

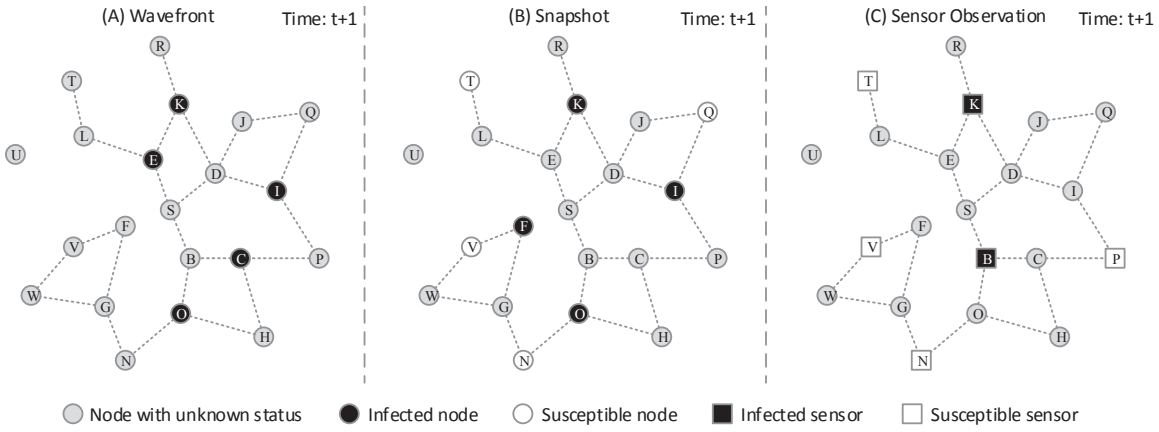



Figure 4.3: Three types of observations in regards to the rumor spreading in Fig. 4.1. (A) Wavefront; (B) Snapshot; (C) Sensor.

introduce two sub-states of nodes being infected: 'contagious' (*Con.*) and 'misled' (*Mis.*), see Fig. 4.2. An infected node first becomes contagious and then transit to being misled. The *Con.* state describes the state of nodes newly infected. More specifically, a node is *Con.* at time $t$ means this node is susceptible at time $t-1$ but becomes infected at time $t$. An misled node will stay being infected until it recovers. For instance, sensors can record the time at which they get infected, and the infection time is crucial in detecting rumor sources because it reflects the infection trend and speed of a rumor. Hence, the introduction of contagious and misled states is intrinsic to the rumor spreading framework.

### 4.2.3 Observations on Time-varying Social Netowrks

Prior knowledge for source identification is provided by various types of *partial observations* upon time-varying social networks. According to previous work on static networks, we collect three categories of partial observations: *wavefronts*, *snapshots*,

and *sensor observations*. We denote the set of observed nodes as $O = \{o_1, o_2, ..., o_n\}$. Following the rumor spreading in Fig. 4.1, we will explain each type of the partial observations as follows.

*Wavefront* [12]: Given a rumor spreading incident, a wavefront provides partial knowledge of the time-varying social network status. Only the users who are in the wavefront of the spreading can be observed (*i.e.*, all the contagious nodes in the latest time window are observed). Fig. 4.3(A) shows an example of the wavefront in the rumor spreading in Fig. 4.1. We see that nodes $C, E, I, K$ and $O$ are in the wavefront as they transit to being contagious at time $t + 1$.

*Snapshot* [58]: Given a rumor spreading incident, a snapshot also provides partial knowledge of the time-varying social network status. In this case, only a group of users can be observed in the latest time window when the snapshot is taken. The states of the observed users can be susceptible, infected or recovered. We use $O_S$, $O_I$ and $O_R$ to denote the observed users who are susceptible, infected or recovered, respectively. This type of observations is the most common one in our daily life. Fig. 4.3(B) shows an example of the snapshot in the rumor spreading in Fig. 4.1 . We see that $O_S = \{N, Q, T, V\}$, $O_I = \{F, I, K, O\}$ and $O_R = \emptyset$.

*Sensor Observation* [82]: Sensors are a group of preselected users in time-varying social networks. The sensors can record the rumor spreading dynamics over them, including the security states and the time window when they get infected (more specifically, become contagious). We introduce $O_S$ and $O_I$ to denote the set of susceptible and infected sensors, respectively. For each $o_i \in O_I$, the infection time is denoted by $t_i$. This type of observation is usually obtained from sensor networks. Fig. 4.3(C) shows an example of the sensor observations in the rumor spreading in Fig. 4.1. In

this case, $O_S = \{N, P, T, V\}$, $O_I = \{K, B\}$, and the infection time of node $K$ is $t+1$, and node $B$ is infected at time $t$.

We can see that these three types of partial observations provide three different categories of partial knowledge of the time-varying social network status. Different types of observations are suitable for different circumstances in real-world applications. Readers can refer to [12, 82, 120] for further discussion on different types of partial observations. The partial knowledge together with the time-varying characteristics of social networks make the tracing back of rumor sources much more difficult.

## 4.3 Narrowing Down the Suspects

Current methods of source identification need to scan every node in the underlying network. This is a bottlenecks of identifying rumor sources: *scalability*. It is necessary to narrow down a set of suspects, especially in large-scale networks. In this section, we develop a *reverse dissemination method* to identify *a small set of suspects*. The details of the method are presented in Section 4.3.1, and its efficiency will be evaluated in Section 4.3.2.

### 4.3.1 Reverse Dissemination Method

In this subsection, we first present the rationale of the reverse dissemination method. Then, we show how to apply the reverse dissemination method into different types of partial observations on networks.

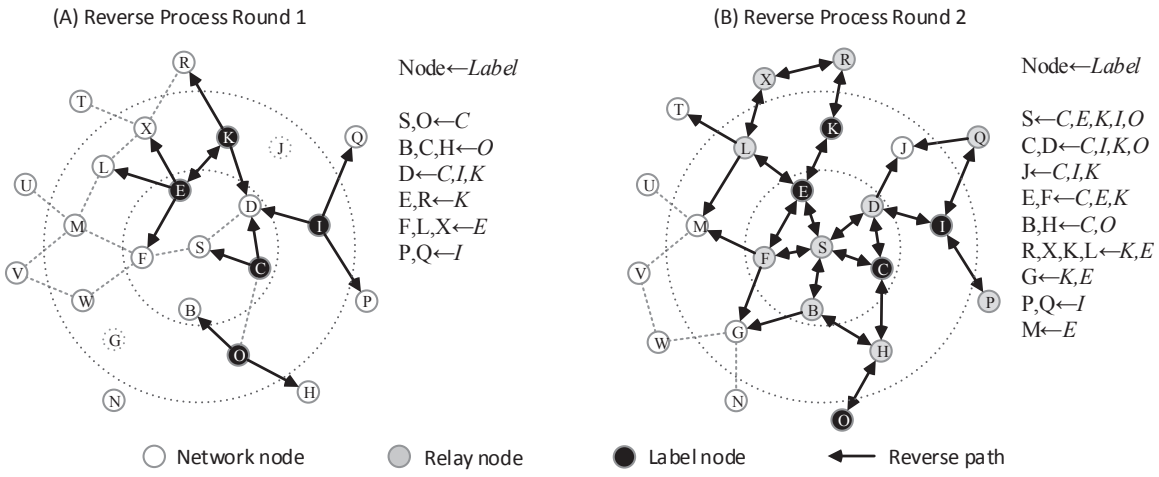



Figure 4.4: Illustration of the reverse dissemination process in regards to the wavefront observation in Fig. 4.3 (A). (A) The observed nodes broadcast labeled copies of rumors to their neighbors in time window $t$; (B) The neighbors who received labeled copies will relay them to their own neighbors in time window $t-1$.

### Rationale

The rationale of the reverse dissemination method is to send copies of rumors along the reversed dynamic connections from observed nodes to exhaust all possible spreading paths leading to the observation. The node from which all the paths, covering all the observed nodes' states, originated is more likely to be a suspect. The reverse dissemination method is inspired from the Jordan method [120]. The reverse dissemination method is different from the Jordan method, because our method is based on time-varying social networks (involving the physical mobility and online/offline status of users) rather than static networks. In Fig. 4.4, we show a simple example to illustrate the reverse dissemination process. This example follows the rumor spreading in Fig. 4.1 and the wavefront observation in Fig. 4.3(A). All wavefront nodes $O_I = \{E, C, I, K, O\}$ observed in time window $t+1$ are labeled as black in

Fig. 4.4 (A). The whole process is composed of two rounds of reverse dissemination. In round 1 (Fig. 4.4 (A)), all observed nodes broadcast labeled copies reversely to their neighbors in time window $t$. For example, nodes $S$ and $O$ received copies of node $C$ ($S, O \leftarrow C$), and node $D$ received copies of three observed nodes $C, I$ and $K$ ($D \leftarrow C, I, K$). In round 2 (Fig. 4.4 (B)), the neighbors who have received labeled copies will relay them to other neighbors in time window $t-1$. In each round, the labels will be recorded in each relay node. We can see from Fig. 4.4(B) that node $S$ has received all copies from all the observed nodes ($S \leftarrow C, E, K, I, O$). Then, node $S$ is chosen to be a suspect.

We notice that the starting time for each observed node starting their reverse dissemination processes varies in different types of observations. For a wavefront, since all the observed nodes are supposed to be contagious in the latest time window, all the observed nodes need to simultaneously start their reverse dissemination processes. For a snapshot, the observed nodes stay in their states in the latest time window. Therefore, the reverse dissemination processes will also simultaneously starts from all the observed nodes. However, for a sensor observation, because the infected sensors record their infection time, the starting time of reverse dissemination for each sensor will be determined by $t_i$. More specifically, the latest infected sensors first start their reverse dissemination processes, and then the sensors infected in the previous time window, until the very first infected sensors.

**Wavefront**

Given a reverse dissemination process starting from an observed node $o_i$, we use $P_C(u, t|o_i)$ to denote the probability of an arbitrary node $u$ to be contagious after

time $t$, where $t$ denotes the time span of the whole reverse dissemination process. Let all observed nodes $o_i$ start their reverse dissemination processes in the latest time window. To match the wavefront, it is expected a suspect $u$ can simultaneously receive rumor copies from all $o_i \in O$ (*i.e.*, the rumor copies sent from all observed nodes can make node $u$ become contagious simultaneously). Mathematically, we identify those nodes who can provide the maximum likelihood, $L(u,t)$, of being a suspect receiving copies from all the observed nodes, as in

$$L(u,t) = \sum_{o_i \in O} \ln(P_C(u,t|o_i)). \tag{4.3.1}$$

For the convenience of computation, we adopt logarithmic function $\ln(\cdot)$ in Eq. (4.3.1) to derive the maximum likelihood. We use $U$ to denote the set of suspects. The ones who provide larger values of $L(u,t)$ are recognized as a member of set $U$.

**Snapshot**

To match the snapshot observation (which includes susceptible, infected or recovered nodes), it is expected that a suspect $u$ needs to satisfy the following three principles at time $t$. First, copies of rumors disseminated from observed *susceptible* nodes $o_i \in O_S$ cannot reach node $u$ at time $t$ (*i.e.*, $u$ is still susceptible). Second, copies of rumors disseminated from observed *infected* nodes $o_j \in O_I$ can reach node $u$ at time $t$ (*i.e.*, $u$ becomes infected). Third, copies of rumors disseminated from observed *recovered* nodes $o_k \in O_R$ can arrive at node $u$ before time $t$ (*i.e.*, $u$ becomes recovered). Again, we employ maximum likelihood to capture this kind of nodes, as in

$$\begin{aligned} L(u,t) = &\sum_{o_i \in O_S} \ln(P_S(u,t|o_i)) + \sum_{o_j \in O_I} \ln(P_I(u,t|o_j)) \\ &+ \sum_{o_k \in O_R} \ln(P_R(u,t|o_k)), \end{aligned} \tag{4.3.2}$$

where $P_S(u,t|o_i)$, $P_I(u,t|o_i)$ and $P_R(u,t|o_i)$ denote the probabilities of $u$ to be susceptible, infected or recovered after time $t$, respectively, given that the reverse dissemination started from $o_i$.

**Sensor**

For sensor observations, according to our previous discussion, we let infected sensor $o_i \in O_I$ start to reversely disseminate copies of the rumor at time $\hat{t}_i = T - t_i$, where $T = \max\{t_i|o_i \in O_I\}$. We also let the susceptible sensors $o_j \in O_S$ start to reversely disseminate copies of rumors at time $t$=0. To match a sensor observation, it is expected a suspect $u$ needs to satisfy the following two principles at time $t$. First, copies of rumors disseminated from *susceptible* sensors $o_i \in O_S$ cannot reach node $u$ at time $t$ (*i.e.*, node $u$ is still susceptible). Second, copies of rumors disseminated from all *infected* sensors $o_j \in O_I$ can be received by node $u$ at time $t$ (*i.e.*, node $u$ becomes contagious). Mathematically, we determine the suspects by computing their maximum likelihood, as in

$$\begin{aligned} L(u,t) = &\sum_{o_i \in O_I} \ln(P_C(u, t+\hat{t}_i|o_i)) \\ &+ \sum_{o_j \in O_S} \ln(P_S(u,t|o_j)). \end{aligned} \tag{4.3.3}$$

The values of $P_S(u,t|o_i)$, $P_C(u,t|o_i)$, $P_I(u,t|o_i)$ and $P_R(u,t|o_i)$ will be calculated by the model introduced in Section 4.4.2. We summarize the reverse dissemination method in **Algorithm 1**.

**Algorithm 1:** Reverse dissemination

**Input**: A set of observed nodes $O = \{o_1, o_2, ..., o_n\}$, a set of infection times of the observed nodes $\{t_1, t_2, ..., t_n\}$, a threshold $\alpha$, and a threshold $t_{max}$.

**Initialize**: A set of suspects $U = \emptyset$, and $t_1 = ... = t_n = T$ if $O$ is a snapshot/wavefront, otherwise $T = \max\{t_1, t_2, ..., t_n\}$.

**for** *(t starts from 1 to a given maximum value $t_{max}$)* **do**

  **for** *($o_i$: i starts from 1 to n)* **do**

    **if** *($o_i$ has not started to disseminate the rumor)* **then**

      Start to propagate the rumor from user $o_i$ separately and independently at time $t + T - t_i$.

  **for** *(u: any node in the whole network)* **do**

    **if** *(user u received n separate rumors from O)* **then**

      Compute the maximum likelihood $L(u, t)$ for user $u$;

      Add user $u$ into the set $U$.

  **if** *($|U| \geq \alpha N$)* **then**

    Keep the first $\alpha N$ suspects with large maximum likelihoods in $U$, and delete all the other suspects.

    **Stop**.

**Output**: A set of suspects $U$.

### 4.3.2 Performance Evaluation

We evaluate the performance of the reverse dissemination method in real time-varying social networks. Similar to Lokhov et. al's work [51], we consider the infection probabilities and recovery probabilities to be uniformly distributed in (0,1), and the average infection and recovery probabilities are set to be 0.6 and 0.3. We also use $\alpha$ to denote the ratio of suspects over all nodes, $\alpha = |U|/N$, where $N$ is the number of all nodes in a time-varying social network. The value of $\alpha$ ranges from 5% to 100%. We randomly choose the real source in 100 runs of each experiment. The number of

100 comes from the wrok in [126].

We consider four real time-varying social networks in Table 4.1: The MIT reality [24] dataset captures communication from 97 subjects at MIT over the course of the 2004-2005 academic year. The Sigcom09 [81] dataset contains the traces of Bluetooth device proximity of 76 persons during SIGCOMM 2009 conference in Barcelona, Spain. The Enron Email [98] dataset contains record of email conversations from 143 users in 2001. The Facebook [105] dataset contains communications from 45,813 users during December 29th, 2008 and January 3rd, 2009. All of these datasets reflect the physical mobility and online/offline features of time-varying social networks. According to the study in [87], an appropriate temporal resolution $\Delta t$ is important to correctly characterize the dynamical processes on time-varying networks. Therefore, we need to be cautious when we choose the time interval of size $\Delta t$. Furthermore, many social networks have been shown small-world, *i.e.*, the average distance $l$ between any two nodes is small, generally $l \leq 6$. Previous extensive works show that rumors can spread quickly in social networks, generally after 6-10 time ticks of propagation (see [21]). Hence, we divided the social networks into 6-10 time windows. Therefore, for the datasets used in this chapter, we uniformly divide each into 6-10 discrete time windows [87]. For other division of temporal resolution, readers could

Table 4.1: Comparison of Data Collected in the Experiments.

| Dataset | MIT | Sigcom09 | Email | Facebook |
|---|---|---|---|---|
| Device | Phone | Phone | Laptop | Laptop |
| Network type | Bluetooth | Bluetooth | WiFi | WiFi |
| Duration (days) | 246 | 5 | 14 | 6 |
| # of devices | 97 | 76 | 143 | 45,813 |
| # of contacts | 54,667 | 69,189 | 1,246 | 264,004 |

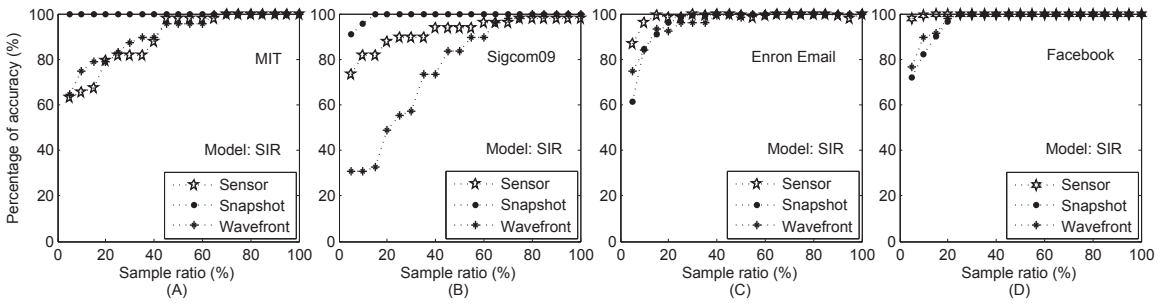



Figure 4.5: Accuracy of the reverse dissemination method in networks. (A) MIT; (B) Sigcom09; (C) Enron Email; (D) Facebook.

refer to [87] for further discussion.

Fig. 4.5 shows the experiment results in the four real datasets. We find the proposed method works quite well in reducing the number of suspects. Especially for snapshots, the searching scale can be narrowed to 5% of all users for the MIT dataset, 15% for the Sigcom09 dataset, and 20% for the Enron Email and Facebook datasets. The number of suspects can be reduced to 45% of all users in the MIT reality dataset under snapshot and wavefront observations. For the Enron Email and Facebook datasets, the number of suspects can be reduced to 20% of all users. The worst case occurred in the Sigcom09 dataset with wavefronts, but our method still achieved a reduction of 35% in the total number of users.

The experiment results on real time-varying social networks show that the proposed method is efficient in narrowing down the suspects. Real-world social networks usually have a large number of users. Our proposed method addresses the scalability in source identification, and therefore is of great significance.

## 4.4 Determining the Real Source

Another bottleneck of identifying rumor sources is to design a good measure to specify the real source. Most of the existing methods are based on *node centralities*, which ignore the propagation probabilities between nodes. Some other methods consider the *BFS trees* instead of the original networks. These violate the rumor spreading processes. In this section, we adopt an innovative *maximum-likelihood based method* to identify the real source from the suspects. A *novel rumor spreading model* will also be introduced to model rumor spreading in *time-varying social networks*.

### 4.4.1 A Maximum-likelihood (ML) Based Method

#### Rationale

The key idea of the ML-based method is to expose the suspect from set $U$ that provides the largest maximum likelihood to match the observation. It is expected that the real source will produce a rumor propagation which not only temporally but also spatially matches the observation more than other suspects. Given an observation $O = \{o_1, o_2, ..., o_n\}$ in a time-varying network, we let the spread of rumors start from an arbitrary suspect $u \in U$ from the time window that is $t_u$ before the latest time window. For an arbitrary observed node $o_i$, we use $P_S(o_i, t_u|u)$ to denote the probability of $o_i$ being susceptible at time $t_u$, given that the spread of rumors starts from suspect $u$. Similarly, we have $P_C(o_i, t_u|u)$, $P_I(o_i, t_u|u)$ and $P_R(o_i, t_u|u)$ representing the probabilities of $o_i$ being contagious, infected and recovered at time $t_u$, respectively. We use $\tilde{L}(t_u, u)$ to denote the maximum likelihood of obtaining the observation when the rumor started from suspect $u$. Among all the suspects in $U$, we can estimate the

real source by choosing the maximum value of the ML, as in

$$(u^*, t^*) = \arg\max_{u \in U} \tilde{L}(t_u, u). \tag{4.4.1}$$

The result of Eq. (4.4.1) suggests that suspect $u^*$ can provide a rumor propagation not only temporally but also spatially matches the observation better than other suspects. We also have an estimation of infection scale $I(t^*, u^*)$ as a byproduct, as in

$$I(t^*, u^*) = \sum_{i=1}^{N} P_I(i, t^* | u^*). \tag{4.4.2}$$

Later, we can justify the effectiveness of the ML-based method by examining the accuracy of $t^*$ and $I(t^*, u^*)$.

**Wavefront**

In a wavefront, all observed nodes are contagious in the time window when the wavefront is captured. Supposing suspect $u$ is the rumor source, the maximum likelihood $\tilde{L}(t_u, u)$ of obtaining the wavefront $O$ is the product of the probabilities of any observed node $o_i \in O$ being contagious after time $t_u$. We also adopt a logarithmic function to present the computation of the maximum likelihood. Then, we have $\tilde{L}(t_u, u)$ for a wavefront, as in

$$\tilde{L}(t_u, u) = \sum_{o_i \in O} \ln(P_C(o_i, t_u | u)). \tag{4.4.3}$$

**Snapshot**

In a snapshot, the observed nodes can be susceptible, infected or recovered in the time window when the snapshot is taken. Supposing suspect $u$ is the rumor source, the maximum likelihood of obtaining the snapshot is the product of the probabilities of

any observed node $o_i \in O$ being in its observed state. Then, we have the logarithmic form of the calculation for $\tilde{L}(t_u, u)$ in a snapshot, as in

$$\begin{aligned}\tilde{L}(t_u, u) = &\sum_{o_i \in O_S} \ln(P_S(o_i, t_u|u)) + \\ &\sum_{o_j \in O_I} \ln(P_I(o_j, t_u|u)) + \sum_{o_k \in O_R} \ln(P_R(o_k, t_u|u)).\end{aligned} \tag{4.4.4}$$

**Sensor**

In a sensor observation, each infected sensor $o_i \in O_I$ records its infection time $t_i$. Although the absolute time $t_i$ cannot directly suggest the spreading time of the rumor, we can derive the relative infection time of each sensor. Supposing suspect $u$ is the rumor source, for an arbitrary infected sensor $o_i$, its relative infection time is $\tilde{t}_i = t_i - \tilde{t} + t_u$ where $\tilde{t} = \min\{t_i | o_i \in O_I\}$, and $t_u$ is obtained from **Algorithm 1**. For suspect $u \in U$, the maximum likelihood $\tilde{L}(t_u, u)$ of obtaining the observation is the product of the probability of any sensor $o_i$ to be in its observed state at time $\tilde{t}_i$. Then, we have the logarithmic form of the calculation for $\tilde{L}(t_u, u)$ in a sensor observation, as in

$$\tilde{L}(t_u, u) = \sum_{o_i \in O_I} \ln(P_C(u, \tilde{t}_i|o_i)) + \sum_{o_j \in O_S} \ln(P_S(u, t_u|o_j)). \tag{4.4.5}$$

Note that, $P_S(u, t|o_i)$, $P_C(u, t|o_i)$, $P_I(u, t|o_i)$, and $P_R(\tilde{u}, t|o_i)$ can be calculated in the rumor spreading model in Section 4.4.2. We summarize the method of determining rumor sources in **Algorithm 2**.

### 4.4.2 Propagation Model

In this subsection, we introduce an *analytical model* to present the spreading dynamics of rumors in time-varying social networks. The state transition of each node follows

**Algorithm 2:** Targeting the suspect

**Input**: A set of suspects $U$, a set of observed nodes $O$, and a threshold $t_{max}$.

**Initialize**: $L_{max} = 0$, $u^* = \emptyset$, $t^* = 0$.

**for** *($\tilde{u}$: any node in set $U$)* **do**

  **for** *($t$ starts from 1 to a given maximum value $t_{max}$)* **do**

    Disseminate the rumor from suspect $\tilde{u}$.

    **if** *(We can obtain the observation $O$)* **then**

      Compute the maximum likelihood value $\tilde{L}(t, \tilde{u})$.

      **if** *($\tilde{L}(t, \tilde{u}) > L_{max}$)* **then**

        $L_{max} = \tilde{L}(t, \tilde{u})$;

        $u^* = \tilde{u}$;

        $t^* = t$.

      **if** *($\tilde{L}(t, \tilde{u}) < \tilde{L}(t-1, \tilde{u})$)* **then**

        **Stop**.

**Output**: The rumor source $u^*$ and propagation time $t^*$.

the SIR scheme introduced in Section 4.2.2. For rumor spreading processes among users, we use this model to calculate the probabilities of each user in various states.

In the modeling, every user is initially susceptible. We use $\eta_{ji}(t)$ to denote the spreading probability from user $j$ to user $i$ in time window $t$. Then, we can calculate the probability of a susceptible user being infected by his/her infected neighbors as in

$$v(i,t) = 1 - \prod_{j \in N_i} [1 - \eta_{ji}(t) \cdot P_I(j, t-1)], \tag{4.4.6}$$

where, $N_i$ denotes the set of neighbors of user $i$. Then, we can compute the probability of an arbitrary user to be susceptible at time $t$ as in

$$P_S(i,t) = [1 - v(i,t)] \cdot P_S(i, t-1). \tag{4.4.7}$$

Once a user gets infected, he/she becomes contagious. We then have the probability

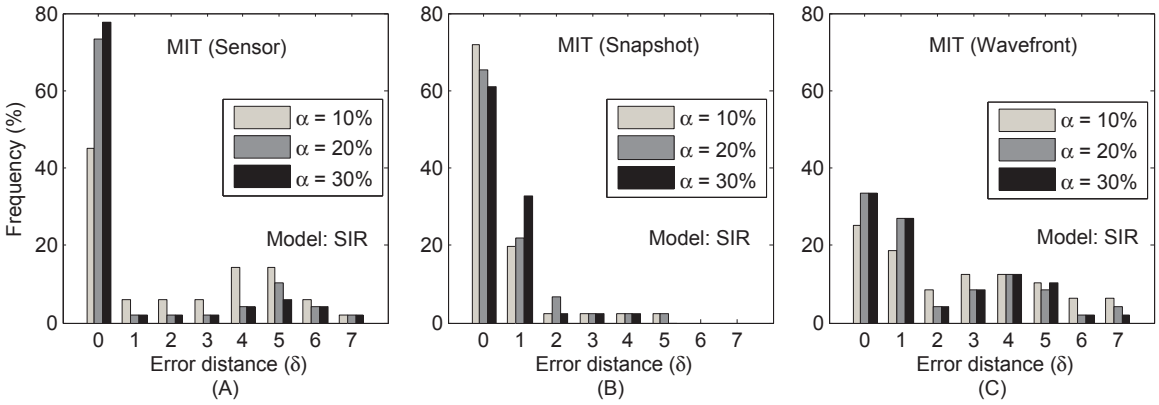



Figure 4.6: The distribution of error distance ($\delta$) in the MIT Reality dataset. (A) Sensor; (B) Snapshot; (C) Wavefront.

that an arbitrary user is contagious at time $t$ as in

$$P_C(i,t) = v(i,t) \cdot P_S(i,t-1). \tag{4.4.8}$$

Since an infected user can be either contagious or misled, we can obtain the value of $P_I(i,t)$ as in

$$P_I(i,t) = P_C(i,t) + (1 - q_i(t)) \cdot P_I(i,t-1). \tag{4.4.9}$$

Then, the value of the $P_R(i,t)$ can be derived from

$$P_R(i,t) = P_R(i,t-1) + q_i(t) \cdot P_I(i,t-1). \tag{4.4.10}$$

This model analytically derives the probabilities of each user in various states in an arbitrary time $t$. This in addition constitutes the maximum likelihood $L(u,t)$ of an arbitrary user $u$ being a suspect in time window $t$ in Section 4.3.1. This also supports the calculation of the maximum likelihood $\tilde{L}(t,u)$ to match the observation in time window $t$, given that the rumor source is the suspicious user $u$ in Section 4.4.1.

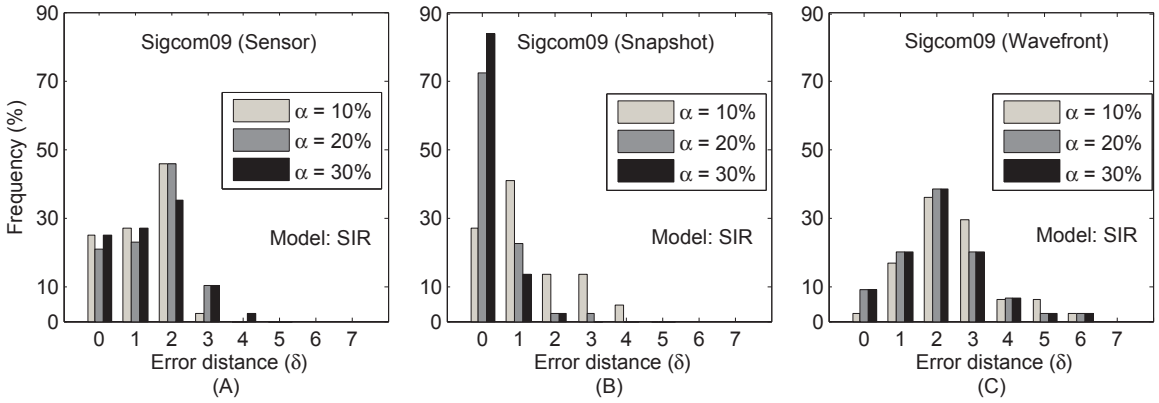



Figure 4.7: The distribution of error distance ($\delta$) in the Sigcom09 dataset. (A) Sensor; (B) Snapshot; (C) Wavefront.

## 4.5 Evaluation

In this section, we evaluate the efficiency of our source identification method. The experiment settings are the same as those presented in Section 4.3.2. Specifically, we let the sampling ratio $\alpha$ range from 10% to 30%, as the reverse dissemination method has already achieved a good performance with $\alpha$ dropping in this range.

### 4.5.1 Accuracy of Rumor Source Identification

We evaluate the accuracy of our method in this subsection. We use $\delta$ to denote the error distance between a real source and an estimated source. Ideally, we have $\delta = 0$ if our method accurately captures the real source. In practice, we expect that our method can accurately capture the real source or a user very close to the real source (i.e., $\delta$ is very small). As the user close to the real source usually has similar characteristics with the real source, quarantining or clarifying rumors at this user is also very significant to diminish the rumors [95].

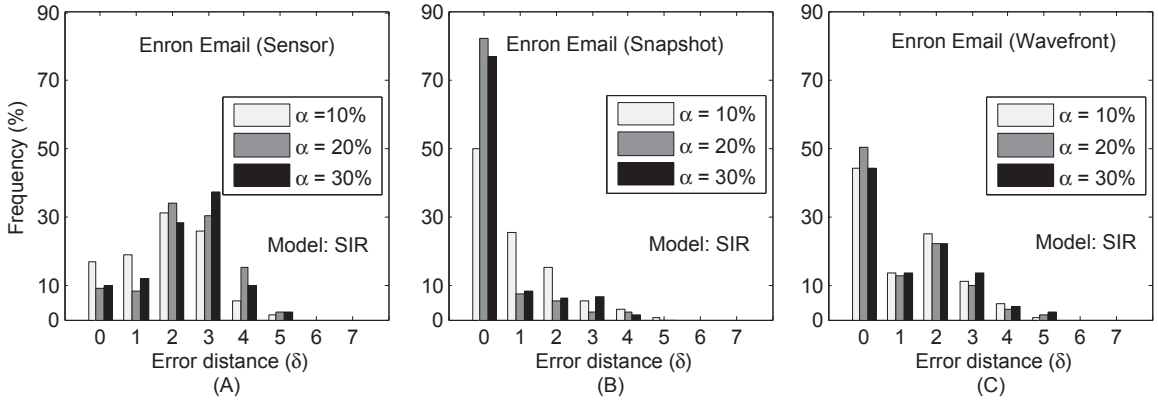



Figure 4.8: The distribution of error distance ($\delta$) in the Enron Email dataset. (A) Sensor; (B) Snapshot; (C) Wavefront.

Our method shows good performances in the four real time-varying social networks. Fig. 4.6 shows the frequency of the error distances ($\delta$) in the MIT reality dataset under different categories of observations. When the sampling ratio $\alpha \geq 20\%$, our method can identify the real sources with an accuracy of 78% for the sensor observations, more than 60% for the snapshots, and around 36% for the wavefronts. For the wavefronts, although our method cannot identify real sources with very high accuracy, the estimated sources are very close to the real sources, and are generally 0-2 hops away. Fig. 4.7 shows the frequency of the error distances $\delta$ in the Sigcom09 dataset. When the sampling ratio $\alpha \geq 20\%$, the proposed method can identify the real sources with an accuracy of more than 70% for the snapshots. For the other two categories of observations, although our method cannot identify real sources with very high accuracy, the estimated sources are very close to the real sources, with an average of 1-2 hops away in the sensor observations, and 1-3 hops away for the wavefronts. Fig. 4.8 shows the performance of our method in the Enron Email dataset. When the sampling ratio $\alpha \geq 20\%$, our method can identify the real sources with

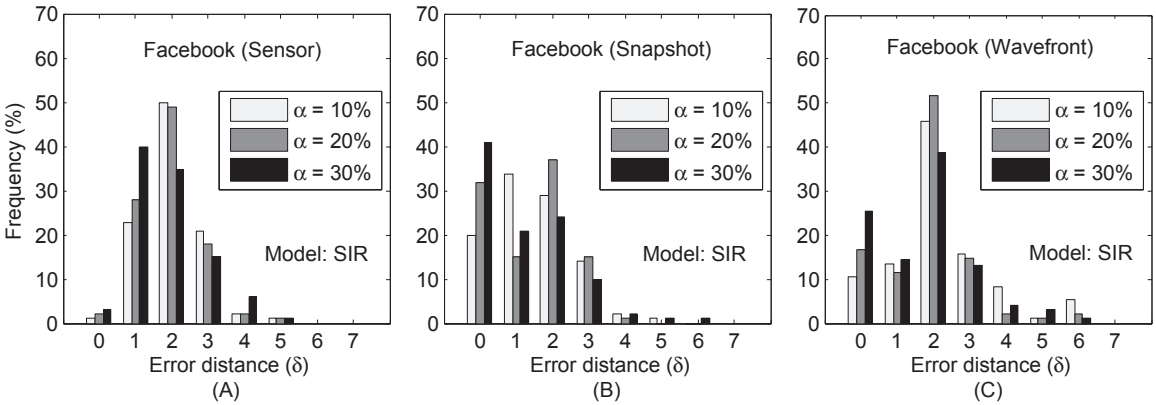



Figure 4.9: The distribution of error distance ($\delta$) in the Facebook dataset. (A) Sensor; (B) Snapshot; (C) Wavefront.

an accuracy of 80% for the snapshots, and more than 45% for the wavefronts. The estimated sources are very close to the real sources, with an average 1-3 hops away in the sensor observations. Fig. 4.9 shows the performance of our method in the Facebook dataset. Similarly, when the sampling ratio $\alpha \geq 20\%$, the proposed method can identify the real sources with an accuracy of around 40% for the snapshots. The estimated sources are very close to the real sources, with an average of 1-3 hops away from the real sources under the sensor and wavefront observations.

Compared with previous work, our proposed method is superior because our method can work in time-varying social networks rather than static networks. Our method can achieve around 80% of all experiment runs that accurately identify the real source or an individual very close to the real source. However, the previous work of [107] and [96] has theoretically proven their accuracy was at most 25% or 50% in tree-like networks, and their average error distance is 3-4 hops away.

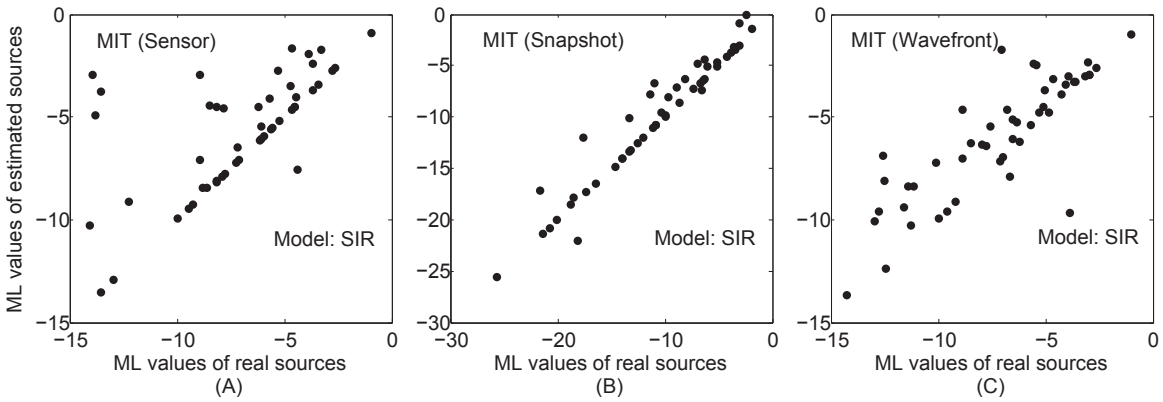



Figure 4.10: The correlation between the maximum likelihood of the real sources and that of the estimated sources in the MIT reality dataset. (A) Sensor observation; (B) Snapshot observation; (C) Wavefront observation.

### 4.5.2 Effectiveness Justification

We justify the effectiveness of our ML-based method from three aspects: the correlation between the ML of the real sources and that of the estimated sources, the accuracy of estimating rumor spreading time, and the accuracy of estimating rumor infection scale.

#### Correlation between real sources and estimated sources

We investigate the correlation between the real sources and the estimated sources by examining the correlation between their maximum likelihood values. For different types of observation, the maximum likelihood of an estimated source can be obtained from Eq. (4.4.3), Eq. (4.4.4) or Eq. (4.4.5), *i.e.*, $\tilde{L}(t^*, u^*)$. The maximum likelihood of a real source is obtained by replacing $u^*$ and $t^*$ as the real source and the real rumor spreading time, respectively. If the estimated source is in fact the real source, their maximum likelihood values should present high correlation.

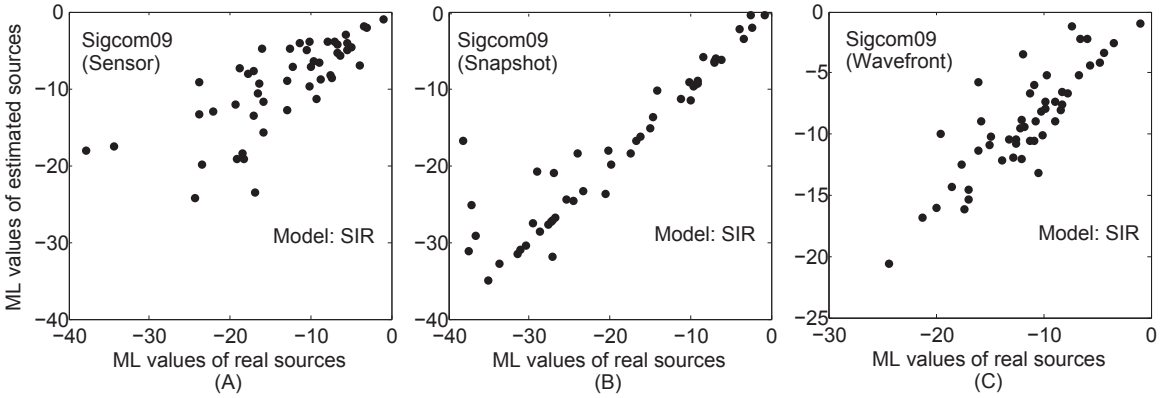



Figure 4.11: The correlation between the maximum likelihood of the real sources and that of the estimated sources in the Sigcom09 dataset. (A) Sensor observation; (B) Snapshot observation; (C) Wavefront observation.

The correlation results of the maximum likelihood values when $\alpha = 20\%$ in the four time-varying social networks are shown from Fig. 4.10 to Fig. 4.13. We see that the maximum likelihood values of the real sources and that of the estimated sources are highly correlated with each other. Their maximum likelihood values approximately form linear relationships to each other. Fig. 4.10 shows the results in the MIT reality dataset. We can see that the maximum likelihood values of the real sources and that of the estimated sources are highly correlated in both sensor and snapshot observations. The worst results occurred in wavefront observations, however the majority of the correlation results still tend to be clustered in a line. These exactly reflect the accuracy of identifying rumor sources in Fig. 4.6. The results in the Sigcom09 dataset are shown in Fig. 4.11. We see that the maximum likelihood values are highly correlated in both snapshot and wavefront observations. The worst results occurred in sensor observations, however the majority of the correlation results still tend to be clustered in a line. These exactly reflect the accuracy of identifying

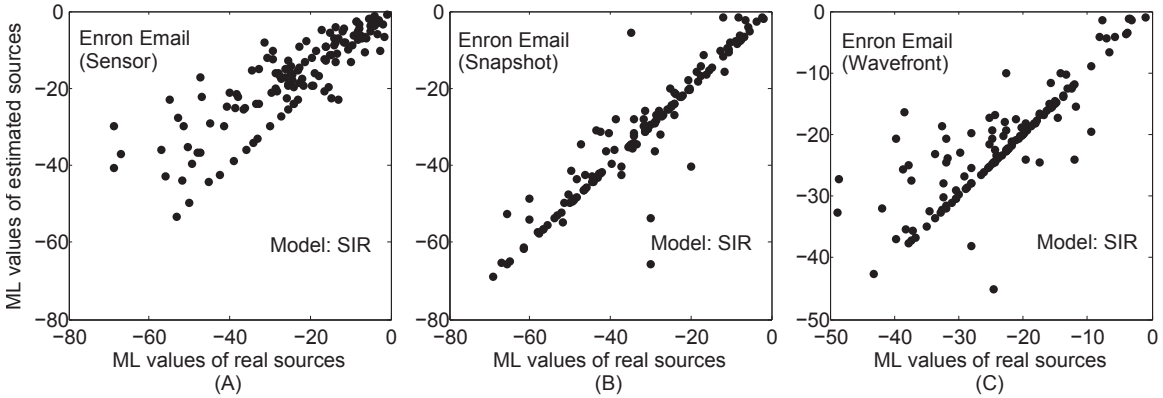



Figure 4.12: The correlation between the maximum likelihood of the real sources and that of the estimated sources in the Enron Email dataset. (A) Sensor observation; (B) Snapshot observation; (C) Wavefront observation.

rumor sources in Fig. 4.7. The results in the Enron Email dataset are shown in Fig. 4.12. We see that the maximum likelihood values are highly correlated in both snapshot and wavefront observations, and slightly correlated in sensor observations. These exactly reflect the accuracy of identifying rumor sources in Fig. 4.8. Similar results can be found in the Facebook dataset in Fig. 4.13, which precisely reflects the accuracy of identifying rumor sources in Fig. 4.9.

The strong correlation between the ML values of the real sources and that of the estimated sources in time-varying social networks reflects the effectiveness of our ML-based method.

### Estimation of spreading time

As a byproduct, our ML-based method can also estimate the spreading time (in Eq. (4.4.1)) of rumors. In order to justify the effectiveness of our proposed method, we further investigate the effectiveness of this byproduct. We expect the estimate can

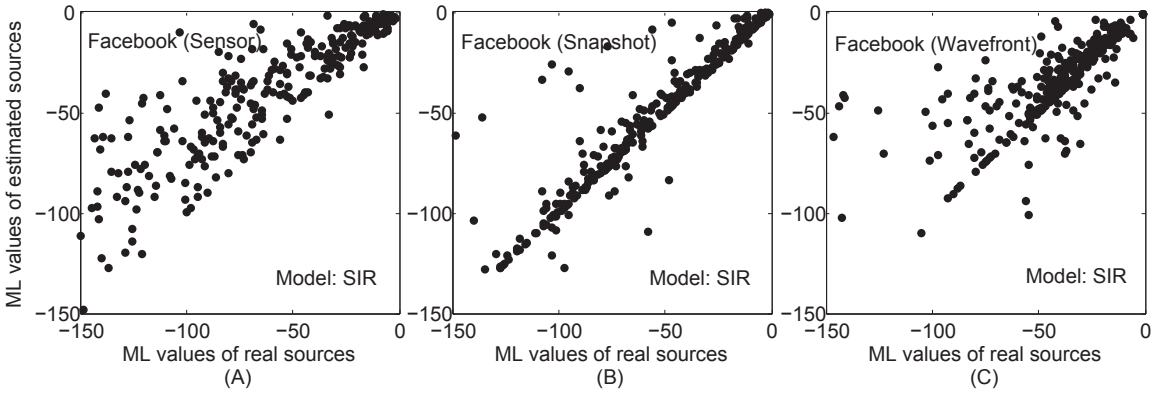



Figure 4.13: The correlation between the maximum likelihood of the real sources and that of the estimated sources in the Facebook dataset. (A) Sensor observation; (B) Snapshot observation; (C) Wavefront observation.

accurately expose the real spreading time of rumors. We let the real spreading time vary from 2 to 6 in four real time-varying social networks. The experiment results are shown in Table 4.2.

As shown in Table 4.2, we analyze the means and the standard deviations of the estimated spreading time. We see that the means of the estimated spreading time are very close to the real spreading time, and most results of the standard deviations are smaller than 1. Especially when the spreading time $T = 2$, our ML-based method in sensor observations and wavefront observations can accurately estimate the spreading time in the MIT reality, Sigcom09 and Enron Email datasets. The results are also quite accurate in the Facebook dataset. From Table 4.2, we can see that our method can estimate the spreading time with extremely high accuracy in wavefront observations, and relatively high accuracy in snapshot observations.

Both the means and standard deviations indicate that our method can estimate the real spreading time with high accuracy. The accurate estimate of the spreading

Table 4.2: Accuracy of Estimating Rumor Spreading Time.

| Environment settings | | Estimated spreading time | | | |
|---|---|---|---|---|---|
| Observation | $T$ | MIT | Sigcom09 | Email | Facebook |
| Sensor | 2 | 2±0 | 2±0 | 2±0 | 1.787±0.411 |
| | 4 | 4.145±0.545 | 3.936±0.384 | 4.152±0.503 | 3.690±0.486 |
| | 6 | 6.229±0.856 | 5.978±0.488 | 6.121±0.479 | 5.720±0.604 |
| Snapshot | 2 | 1.877±0.525 | 2.200±1.212 | 2.212±0.781 | 2.170±0.761 |
| | 4 | 3.918±0.862 | 3.920±0.723 | 3.893±0.733 | 4.050±0.716 |
| | 6 | 6.183±1.523 | 6.125±1.330 | 5.658±1.114 | 5.650±1.266 |
| Wavefront | 2 | 2±0 | 2±0 | 2±0 | 1.977±0.261 |
| | 4 | 4.117±0.686 | 4±0 | 3.984±0.590 | 4.072±0.652 |
| | 6 | 6±0 | 5.680±1.096 | 5.907±0.640 | 5.868±0.864 |

time indicates that our method is effective in rumor source identification.

### Estimation of infection scale

We further justify the effectiveness of our ML-based method by investigating its accuracy in estimating the infection scale of rumors provided by the second byproduct in Eq. (4.4.2). We expect that the ML-based method can accurately estimate the infection scale of each propagation incident. Particularly, we let the rumor spreading initiate from the node with largest degree in each full time-varying social network and spread for 6 time windows in experiments.

In Fig. 4.14, we show the real infection scales at each time tick, and also the estimated infection scales in different types of observations. We can see that the proposed method can provide a fairly accurate estimate of on the infection scales of rumors in the MIT reality dataset, the Sigcom09 dataset and the Facebook dataset in different types of observations. As shown in Fig. 4.14(C), the worst result occurred in the Enron Email dataset after time tick 4. According to our investigation, this was

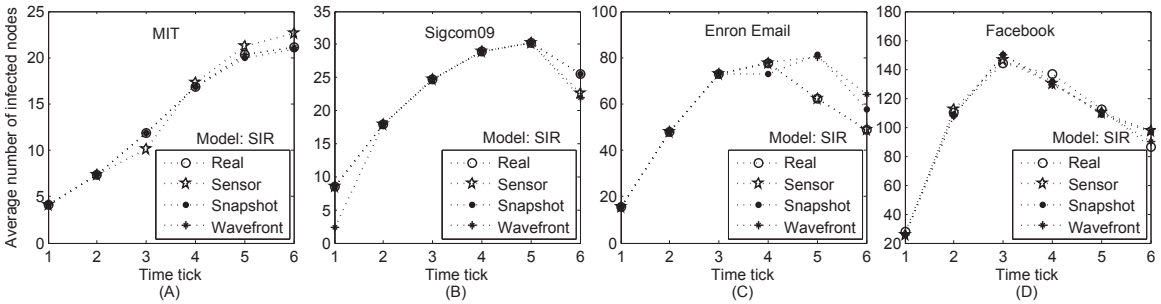



Figure 4.14: The accuracy of estimating infection scale in real networks. (A) MIT; (B) Sigcom09; (C) Enron Email; (D) Facebook.

caused by a great deal of infected nodes that tend to be in the recovered stage in the SIR scheme, which leads to a fairly large uncertainty in the estimate.

To summarize, all of the above evaluations reflect the effectiveness of our method from different aspects: the high correlation between the ML values of the real sources and that of the estimated sources, the high accuracy in estimating spreading time of rumors, and the high accuracy of the infection scale.

## 4.6 Conclusion and Discussion

In this chapter, we explore the problem of rumor source identification in time-varying social networks that can be reduced to a series of static networks by introducing a time-integrating window. In order to address the challenges posted by time-varying social networks, we adopted two innovative methods. First, we utilized a novel reverse dissemination method which can sharply narrow down the scale of suspicious sources. This addresses the scalability issue in this research area and therefore dramatically promotes the efficiency of rumor source identification. Then, we introduced an analytical model for rumor spreading in time-varying social networks. Based on this

model, we calculated the maximum likelihood of each suspect to determine the real source from the suspects. We conduct a series of experiments to evaluate the efficiency of our method. The experiment results indicate that our methods are efficient in identifying rumor sources in different types of real time-varying social networks.

There is some future work can be done in identifying rumor sources in time-varying networks. There are also many other models of rumor propagation, such as the models in [60, 67, 117]. These models can be basically divided into two categories: the macroscopic models and the microscopic models. The macroscopic models, which are based on differential equations, only provide the overall infection trend of rumor propagation, such as the total number of infected nodes [126]. The microscopic models, which are based on difference equations, not only provide the overall infection status of rumor propagation, but they also can estimate the probability of an arbitrary node being in an arbitrary state [82]. In the field of identifying propagation sources, researchers generally choose microscopic models, because it requires to estimate which specific node is the first one getting infected. As far as we know, so far there is no work that is based on the macroscopic models to identify rumor sources in social networks. Future work may also investigate combining microscopic and macroscopic models, or even adopting the mesoscopic models [61, 64], to estimate both the rumour sources and the trend of the propagation. There are also many other microscopic models other than the SIR model adopted in this chapter, such as the SI, SIS, and SIRS models [82, 120]. As we discussed in Section 4.2.2, people generally will not believe the rumor again after they know the truth, i.e., after they get recovered, they will not transit to other states. Thus, the SIR model can reflect the state transition of people when they hear a rumor. We also evaluate the performance of the proposed method

on the SI model. Since the performance of our method on the SI model is similar to that on the SIR model, we only present the results on the SIR model in this chapter.

# Chapter 5

# Identifying Multiple Rumor Sources

## 5.1 Introduction

In this chapter, we aim at addressing the issue of identifying *multiple rumor sources* in complex networks. In the real world, rumors often emerge from multiple sources and spread incredibly fast in complex networks [29, 63, 78, 103]. After the initial outbreak of rumor diffusion, the following *three issues* often attract people's attention: (i) How many sources are there? (ii) Where did the diffusion emerge? and (iii) When did the diffusion breakout?

In the past few years, researchers have proposed a series of methods to identify rumor diffusion sources in networks. However, due to the extreme complexity of the spatiotemporal rumor propagation process and the underlying network structure, **few of existing methods are proposed for identifying multiple diffusion sources**. Luo et al. [57] proposed a multi-rumor-center method to identify multiple rumor sources in tree-structured networks. The computational complexity of this method is $O(n^k)$, where $n$ is the number of infected nodes and $k$ is the number of sources. It

is too computationally expensive to be applied in large-scale networks with multiple diffusion sources. Chen et al. [14] extended the Jordan-center method from single source detection to the identification of multiple sources in tree networks. However, the topologies of real-world networks are far more complex than trees. Fioriti et al. [25] introduced a dynamic age method to identify multiple diffusion sources in general networks. They claimed that the 'oldest' nodes, which were associated to those with largest eigenvalues of the adjacency matrix, were the sources of the diffusion. Similar work to this technique can be found in [84]. However, an essential prerequisite is that we need to know the number of sources in advance.

In this chapter, we propose a novel method, ***K-center method***, to identify multiple rumor sources in general networks. In the real world, the rumor diffusion processes in networks are spatiotemporally complex because of the combined multi-scale nature and intrinsic heterogeneity of the networks. To have a clear understanding of the complex diffusion processes, we adopt a measure, *effective distance*, recently proposed by Brockmann and Helbing [12]. The concept of effective distance reflects the idea that a small propagation probability between neighboring nodes is effectively equivalent to a large distance between them, and vice versa. By using effective distance, the complex spatiotemporal diffusion processes can be reduced to homogeneous wave propagation patterns [12]. Moreover, the relative arrival time of diffusion arriving at a node is independent of diffusion parameters but linear with the effective distance between the source and the node of interest. For multi-source diffusion, we obtain the same linear correlation between the relative arrival time and the effective distance of any infected node. Thereby, supposing that any node can be infected very quickly, an arbitrary node is more likely to be infected by its closest source in terms of effective

distance. Therefore, to identify multiple diffusion sources, we need to partition the infection graph so as to minimize the sum of effective distances between any infected node and the corresponding partition center. The final partition centers are viewed as diffusion sources.

The *contribution* of this part of work is three-fold corresponding to the *three key issues* of rumor source identification problem:

- We propose a *fast method to identify multiple rumor diffusion sources*. Based on this method, we can determine *where the diffusion emerged*. We prove that the proposed method is convergent and the computational complexity is $O(mn\log\alpha)$, where $\alpha = \alpha(m, n)$ is the slowly growing inverse-Ackermann function, $n$ is the number of infected nodes, and $m$ is the number of edges connecting them.

- According to the topological positions of the detected rumor sources, we derive an efficient algorithm to estimate the *spreading time of the diffusion*.

- When the number of sources is unknown, we develop an intuitive and effective approach that can estimate *the number of diffusion sources* with high accuracy.

The rest of this chapter is organized as follows. Preliminary knowledge is introduced in Section 5.2. The problem formulation of multi-source identification is presented in Section 5.3. Section 5.4 presents the K-center method for identifying multiple rumor diffusion sources, followed by Section 5.5 which evaluates the proposed methods in real network topologies. Section 5.6 concludes some remarks in this chapter.

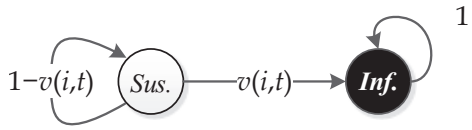



Figure 5.1: The state transition graph of a node in the SI model.

## 5.2 Preliminaries

In this section, we introduce preliminary knowledge used in this chapter, including the *analytic epidemic model* [127] and the concept of *effective distance* [12]. For convenience, we borrow notions from the area of epidemics to represent the states of nodes in a network [110]. A node being infected stands for a person getting infected by a disease, viruses having compromised a computer, or a user believing a rumor. Readers can derive analogous meanings for a node being susceptible or recovered.

### 5.2.1 The Epidemic Model

We adopt the classic *susceptible-infected* (*SI*) model to present the diffusion dynamics of each node. Fig. 5.1 shows the state transition graph of a node in this model.

As shown in Fig. 5.1, every node is initially susceptible (*Sus.*). An arbitrary susceptible node $i$ can be infected (*Inf.*) by its already-infected neighbors with probability $v(i,t)$ at time $t$. Therefore, we can compute the probability of node $i$ to be susceptible at time $t$ as in

$$P_S(i,t) = [1 - v(i,t)] \cdot P_S(i,t-1). \tag{5.2.1}$$

Then, we can obtain the probability of node $i$ to be infected at time $t$ as in

$$P_I(i,t) = v(i,t) \cdot P_S(i,t-1) + P_I(i,t-1). \tag{5.2.2}$$

We use $\eta_{ji}$ to denote the propagation probability from node $j$ to its neighboring node $i$. Then, we can calculate the probability of node $i$ being infected by its neighbors as in

$$v(i,t) = 1 - \Pi_{j \in N_i}[1 - \eta_{ji} \cdot P_I(j, t-1)], \tag{5.2.3}$$

where $N_i$ denotes the set of neighbors of node $i$. This model analytically derives the probability of each node in various states at an arbitrary time. To address real problems, the length of each time tick relies on the real environment. It can be one minute, one hour or one day. We also need to set the propagation probability $\eta_{ij}$ between nodes properly.

### 5.2.2 The Effective Distance

Brockmann and Helbing [12] recently proposed a new measure, *effective distance*, which can disclose the hidden pattern geometry of complex diffusion. The effective distance from a node $i$ to a neighboring node $j$ is defined as

$$e(i,j) = 1 - \log\eta_{ij}, \tag{5.2.4}$$

where $\eta_{ij}$ is again the propagation probability from $i$ to $j$. This concept reflects the idea that a small propagation probability from $i$ to $j$ is effectively equivalent to a large distance between them, and vice versa. To illustrate this measure, a simple example is shown in Fig. 5.2. For instance, the propagation probability is 0.8 between node $S$ and $A$, and is only 0.1 between $S$ and $B$ (see Fig. 5.2(A)). Correspondingly, the effective distance between $S$ and $A$ is 1.22 which is much less than that between $S$ and $B$ (see Fig. 5.2(B)).

Based on the effective distances between neighboring nodes, the length $\lambda(\Gamma)$ of a path $\Gamma = \{u_1, ..., u_L\}$ is defined as the sum of effective lengths along the edges of the

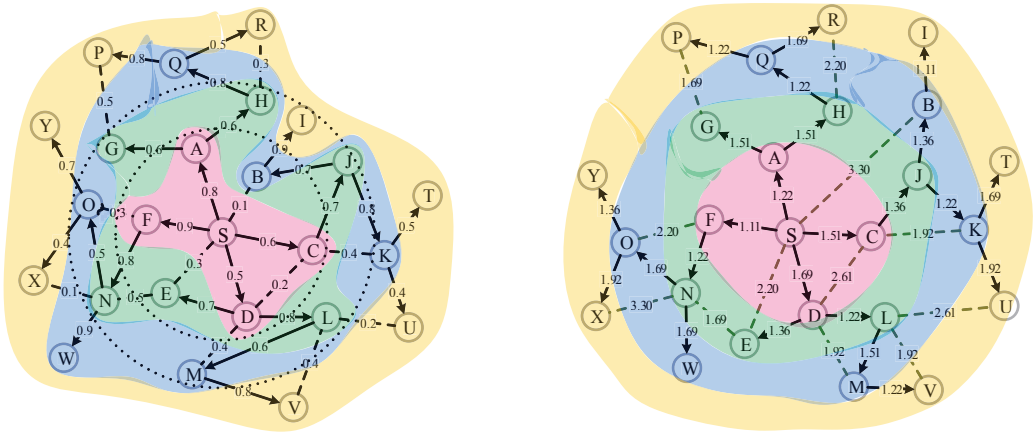



Figure 5.2: An example of altering an infection graph using effective distance. (A) An example infection graph with source $S$. The weight on each edge is the propagation probability. The two dot circles represent the first-order and second-order neighbors of source $S$. The colors indicate the infection order of nodes, *e.g.*, nodes $A$, $C$, $D$ and $F$ are infected after the first time tick. Notice that the diffusion process is spatiotemporally complex. (B) The altered infection graph. The weight on each edge is the effective distance between the corresponding end nodes. Notice that the effective distances from source S to the infected nodes can accurately reflect their infection orders.

path. Moreover, the effective distance from an arbitrary node $i$ to another node $j$ is defined by the length of the shortest path in terms of effective distance from node $i$ to node $j$, *i.e.*,

$$d(i,j) = \min_{\Gamma} \lambda(\Gamma), \tag{5.2.5}$$

From the perspective of diffusion source $s$, the set of shortest paths in terms of effective distance to all the other nodes constitutes a shortest path tree $\Psi_s$. Brockmann and Helbing obtain that the diffusion process initiated from node $s$ on the original network can be represented as wave patterns on the shortest path tree $\Psi_s$.

In addition, they conclude that the relative arrival time of the diffusion arriving at a node is independent of diffusion parameters and is linear with the effective distance from the source to the node of interest.

In this chapter, we will alter the original network by utilizing effective distance through converting the propagation probability on each edge to the corresponding effective distance. Then, by using the linear relationship between the relative arrival time and the effective distance of any infected node, we derive a novel method to identify multiple diffusion sources.

## 5.3 Problem Formulation

Before we present the problem formulation derived in this chapter, we firstly show an *alternate expression* of an arbitrary infection graph by using effective distance (see again Fig. 5.2). Fig. 5.2(A) shows an example of an infection graph with diffusion source $S$. The colors indicate the infection order of nodes (e.g., nodes $A$, $C$, $D$ and $F$ were infected after the first time tick $T = 1$, similarly for the other nodes). Notice that the diffusion process is spatiotemporally complex, because the first-order neighbors of source $S$ can be infected after the second time tick (e.g., node $E$) or even the third time tick (e.g., node $B$), similarly for the second-order and third-order neighbors. We then alter the infection graph by replacing the weight on each edge with the effective distance between the corresponding end nodes (see Fig. 5.2(B)). We notice that the effective distances from source $S$ to all the infected nodes can accurately reflect the infection order of them. This exactly shows that the relative arrival time of an arbitrary node getting infected is linear with the effective distance between the source and the node of interest.

Suppose that at time $T = 0$, there are $k(\geq 1)$ sources, $S^* = \{s_1, ..., s_k\}$, starting the diffusion simultaneously [25, 58]. Several time ticks after the diffusion started, we got $n$ infected nodes. These nodes form a connected infection graph $G_n$, and each source $s_i$ has its infection region $C_i(\subseteq G_n)$. Let $\mathcal{C}^* = \cup_{i=1}^{k} C_i$ be a partition of the infection graph such that $C_i \cap C_j = \emptyset$. for $i \neq j$. Each partition $C_i$ is a connected subgraph in $G_n$ and consists of the nodes whose infection can be traced back to the source node $s_i$. For an arbitrary infected node $v_j \in C_i$, suppose it can be infected in the shortest time, then according to our previous analysis, it will have shorter effective distance to source $s_i$ than to any other source. Therefore, we need to divide the infection graph $G_n$ into $k$ partitions so that each infected node belongs to the partition with the shortest effective distance to the partition center. The final partition centers are considered as the diffusion sources.

Given an infection graph $G_n$, from the above analysis, we know that our goal is to identify a set of diffusion sources $S^*$ and the corresponding partition $\mathcal{C}^*$ of the infection graph $G_n$. To be precise, we aim to find a partition $\mathcal{C}^*$ of $G_n$, to minimize the following objective function,

$$\min_{\mathcal{C}^*} f = \sum_{i=1}^{k} \sum_{v_j \in C_i} d(v_j, s_i), \tag{5.3.1}$$

where node $v_j$ belongs to partition $C_i$ associated with source $s_i$, and $d(v_j, s_i)$ is the shortest-path distance in terms of effective distance between $v_j$ and $s_i$.

Eq. (5.3.1) is the proposed formulation for multi-source identification problem. Since, we need to find out the $k$ centers of the diffusion from Eq. (5.3.1), we name the proposed method of solving the multi-source identification problem as the *K-center method*, which we will detail in the following section.

## 5.4 The K-center Method

In this section, we propose a *K-center method* to identify multiple diffusion sources and the corresponding infection regions in general networks. We *firstly* introduce a method for network partition. Then, we derive the K-center method. *Secondly*, according to the estimated sources, we derive an algorithm to predict the spreading time of the diffusion. *Finally*, we present a heuristic algorithm to estimate the diffusion sources when the number of sources is unknown.

### 5.4.1 Network Partitioning with Multiple Sources

Given an infection network $G_n$ and a set of sources $S^* = \{s_1, ..., s_k\}$, network partition refers to the division of a network into $k$ partitions with $s_i (i \in \{1, 2, ..., k\})$ as the partition centers. According to our previous analysis in Section 5.3, an arbitrary node $v_j \in G_n$ should be classified into partition $C_i$ associated with source $s_i$, such that

$$d(v_j, s_i) = \min_{s_l \in S} d(v_j, s_l). \tag{5.4.1}$$

In essence, for an arbitrary node $v_j \in G_n$, it needs to be associated to source $s_i$ that is the nearest source to $v_j$. This is similar to the Capacity Constrained Network-Voronoi Diagram (CCNVD) problem [115]. Given a graph and a set of service centers, the CCNVD partitions the graph into a set of contiguous service areas that meet service center capacities and minimize the sum of the distances (min-sum) from graph nodes to allotted service centers. The CCNVD problem is important for critical societal applications such as assigning evacuees to shelters and assigning patients to hospitals.

In this chapter, to satisfy Eq. (5.4.1), we utilize the Voronoi strategy to partition the altered infection graph obtained from Section 5.3. The detailed Voronoi partition

**Algorithm 3:** Network partition

**Input**: A set of partition centers $S = \{s_i | i = 1, ..., k\}$ in an infection graph $G_n$.

**Initialize**: Initialize $k$ partitions: $C_1 = \{s1\}$, ...,$C_k = \{s_k\}$.

**for** *(j starts from 1 to n)* **do**

Find the nearest source to node $v_j$ as follows,

$$s_i = \text{argmin}_{s_l \in S} d(x_j, s_l). \tag{5.4.2}$$

Put node $v_j$ into partition $C_i$.

**Output**: A partition of $G_n$: $\mathcal{C}^* = \cup_{i=1}^{k} C_i$.

process is shown in **Algorithm 1**. Future work may use community structure for network partition. Current methods for detecting community structure include strategies based on betweenness [32], information theory [88], and modularity optimization [68].

### 5.4.2 Identifying Diffusion Sources and Regions

In this subsection, we present the K-center method to identify multiple diffusion sources. According to the objective function in Eq. (5.3.1), we need to find a partition $\mathcal{C}^*$ of the altered infection graph $G_n$, which can minimize the sum of the effective distances between each infected node and its corresponding partition center. From the previous subsection, if we randomly choose a set of sources $S$, Voronoi partition can split the network into subnets such that each node is associated with its nearest source. Thus, Voronoi partition can find a local optimal partition of $G_n$ with a fixed set of sources $S$. However, to optimize the partition $\mathcal{C}^*$, we need to adjust the center of each partition so as to minimize the objective function in Eq. (5.3.1). In this chapter, we adjust the center of each partition by choosing a new center as the

**Algorithm 4:** K-center to identify multiple sources

**Input**: An infection graph $G_n$ and the number of sources $k$.

**Initialize**: Initialize an positive integer $L$, and randomly choose a set of sources $S^{(0)} = \{s_1^{(0)}, ..., s_k^{(0)}\} \subseteq G_n$.

**for** *(l starts from 1 to a given maximum value L)* **do**

Use **Algorithm 1** to partition $G_n$ with partition centers $S^{(0)}$, and obtain a partition:

$$C^{(l)} = \cup_{i=1}^{k} C_i^{(l)}. \tag{5.4.3}$$

Find the new center in each partition $C_i^{(l)}$ as follows,

$$s_i^{(l)} = \text{argmin}_{v_j \in C_i^{(l)}} \sum_{v_x \in C_i^{(l)}} d(v_j, v_x), i = 1, ..., k. \tag{5.4.4}$$

**if** *($S^{(l)} = \{s_1^{(l)}, ..., s_k^{(l)}\}$ is the same as $S^{(l-1)}$)* **then**

**Stop**.

**Output**: A set of estimated sources $S^{(l)} = \{s_1^{(l)}, ..., s_k^{(l)}\}$.

node that has the minimum sum of effective distances to all the other nodes in the partition. Therefore, we call this method as the K-center method. This is similar to the rumor-center method and the Jordan-center method that consider rumor centers or Jordan centers as the diffusion sources. As the name suggests, the K-center method is more specific to the multi-source identification. The detailed process of the K-center method is shown in **Algorithm 2**.

The following two theorems show the convergence of the proposed K-center method and its computational complexity.

*Theorem* 1. The objective function in Eq. (5.3.1) is monotonically decreasing in iterations. Therefore, the K-center method is convergent.

Proof. Suppose that at iteration $t$, $S_t = \{s_1^t, ..., s_k^t\}$ are the estimated sources. We then use **Algorithm 1** to partition the infection graph $G_n$ as $\mathcal{C}^t = \cup_{i=1}^{k} C_i^k$. Thus,

the objective function at iteration $t$ becomes

$$f^t = \sum_{i=1}^{k} \sum_{v_j \in C_i^t} d(v_j, s_i^t). \tag{5.4.5}$$

At the next iteration $t+1$, according to the K-center method, we recalculate the center of each partition $C_i^t$ and obtain $S^{t+1} = \{s_1^{t+1}, ..., s_k^{t+1}\}$, such that

$$\sum_{v_j \in C_i^t} d(v_j, s_i^{t+1}) \leq \sum_{v_j \in C_i^t} d(v_j, s_i^t). \tag{5.4.6}$$

Then, the objective function becomes

$$\tilde{f}^t = \sum_{i=1}^{k} \sum_{v_j \in C_i^t} d(v_j, s_i^{t+1}). \tag{5.4.7}$$

From Eqs. (5.4.5) and (5.4.6), we notice that

$$\tilde{f}^t \leq f^t. \tag{5.4.8}$$

We then repartition the infection graph $G_n$ with centers $S^{t+1} = \{s_1^{t+1}, ..., s_k^{t+1}\}$ such that each infected node $v_j \in G_n$ will be associated to a nearest center $s_i^{t+1}$, and obtain a new partition $\mathcal{C}^{t+1} = \cup_{i=1}^{k} C_i^{t+1}$ of $G_n$. Thus, the objective function at iteration $t+1$ becomes

$$f^{t+1} = \sum_{i=1}^{k} \sum_{v_j \in C_i^{t+1}} d(v_j, s_i^{t+1}). \tag{5.4.9}$$

Since each node is classified to the nearest $s_i^{t+1}$, we see that

$$f^{t+1} \leq \tilde{f}^t. \tag{5.4.10}$$

From Eqs. (5.4.8) and (5.4.10), we have

$$f^{t+1} \leq \tilde{f}^t \leq f^t. \tag{5.4.11}$$

Therefore, the objective function in Eq. (5.3.1) is monotonically decreasing, *i.e.*, the K-center method is convergent. □

**Algorithm 5:** K-center identification with unknown number of sources

**Input**: An infection graph $G_n$.

**Initialize**: Initialize the number of sources k = 1, and set $T^{(0)} = 0$.

**while** *(1)* **do**

Use **Algorithm 2** to identify a set of $k$ sources in $G_n$: $S^{(k)} = \{s_1, ..., s_k\}$.

Calculate the spreading time $T^{(k)}$ in Eq. (5.4.13).

**if** $(T^{(k)} = T^{(k-1)})$ **then**

**Stop**.

Update $k = k + 1$.

**Output**: A set of $k$ estimated sources $S^{(k)} = \{s_1, ..., s_k\}$.

*Theorem* 2. Given a infection graph $G_n$ with $n$ nodes and $m$ edges, the computational complexity of the K-center method is $O(mn\log\alpha)$, where $\alpha = \alpha(m, n)$ is the very slowly growing inverse-Ackermann function [80].

Proof. From **Algorithm 2**, we know that the main difficulty of the K-center method stems from the calculation of the shortest paths between node pairs in the altered infection graph $G_n$. Other computation in this algorithm can be treated as a constant. In this chapter, we adopt the Pettie-Ramachandran algorithm [80], to compute all-pairs shortest paths in $G_n$. The computational complexity of the algorithm is $O(mn\log\alpha)$, where $\alpha = \alpha(m, n)$ is the very slowly growing inverse-Ackermann function [80]. Therefore, we have proved the theorem. □

According to Theorem 1, we know that the proposed K-center method is well defined. We notice that the rationale of the K-center method is similar to that of the K-means algorithm in the data-mining field. Similar to the K-means algorithm, there is no guarantee that a global minimum in the objective function will be reached. From Theorem 2, We see that the computational complexity of the K-center method is much less than the method in [58] with $O(n^k)$, and much less than the method in [25] with

$O(n^3)$. In addition, the proposed method can be applied in general networks, whereas the other methods mainly focus on trees. Comparatively, the proposed method is more efficient and practical in identifying multiple diffusion sources in large networks.

### 5.4.3 Predicting Spreading Time

Given an infection graph $G_n$, we can obtain a partition $C^* = \cup_{i=1}^{k} C_i$ of $G_n$ and the corresponding partition centers $S^*$ by using the proposed K-center method. According to the SI model in Section 5.2.1, the spreading time of diffusion can be estimated by the total number of time ticks of the diffusion. Then, we can predict the spreading time based on the hops between the source and the infected nodes in each partition. For an arbitrary source $s_i$ associated with partition $C_i$ and an arbitrary node $v_j \in C_i$, we introduce $h(s_i, v_j)$ to denote the minimum number of hops between $s_i$ and $v_j$. Therefore, the spreading time in each partition can be estimated as in

$$t_i = \max\{h(s_i, v_j) | v_j \in C_i\}, i \in \{1, ..., k\}. \tag{5.4.12}$$

Then, the spreading time of the whole diffusion is as in

$$T = \max\{t_i | i = 1, ..., k\}. \tag{5.4.13}$$

The spreading time $T$ based on hops has simplified the modeling process. In the real world, the spreading time of different paths with the same number of hops may vary from each other. We have solved this temporal problem of the SI model in another chapter [110]. In this field, the majority of current modeling is based on spreading hops [106]. To be consistent with previous work, we adopt the simplified hop-based SI model to study the source identification problem.

### 5.4.4 Unknown Number of Diffusion Sources

In most practical applications, the number of diffusion sources is unknown. In this subsection, we present a heuristic algorithm that allows us to estimate the number of diffusion sources.

From Section 5.4.3, we know that if the number of sources $k$ is given, we can estimate the spreading time $T^{(k)}$ using Eq. (5.4.13). To estimate the number of diffusion sources, we let $k$ start from 1 and compute the spreading time $T^{(1)}$. Then, we increase the number of sources $k$ by 1 in each iteration and compute the corresponding spreading time $T^{(k)}$ until we find $T^{(k)} = T^{(k+1)}$, *i.e.*, the spreading time of the diffusion stays the same when the number of sources increases from $k$ to $k+1$. That is to say when the number of sources increases from $k$ to $k+1$, they can lead to the same infection graph $G_n$. We then choose the number of diffusion sources as $k$ (or $k+1$). The detailed process of the K-center method with unknown number of sources is shown in **Algorithm 3**. We evaluate this heuristic algorithm in Section 5.5.2.

To address real problems, we firstly need to get the underlying network over which the real diffusion spreads. Secondly, we need to measure the propagation probability on each edge in the network. Thirdly, according to the SI model in Section 5.2.1, we need to specify the length of one time tick of the diffusion properly. All of this information is crucial to source identification. However it requires big effort to obtain.

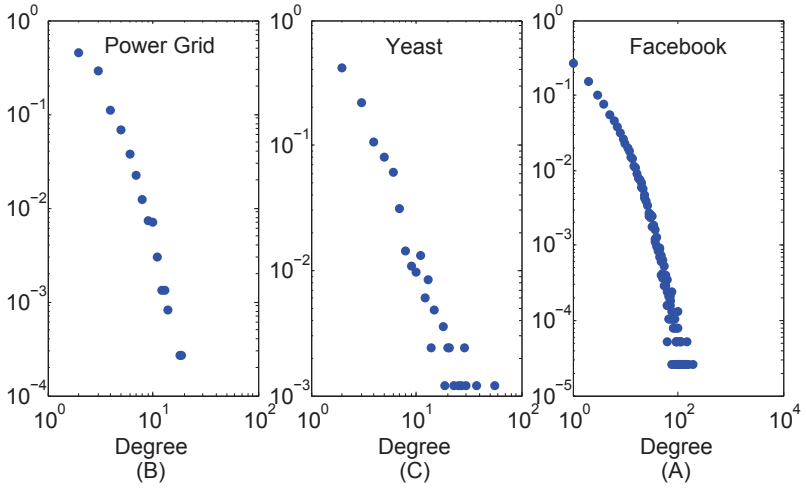



Figure 5.3: Degree distribution. (A) Power Grid; (B) Yeast; (C) Facebook

## 5.5 Evaluation

In this section, we evaluate the proposed K-center method in three real network topologies: the North American Power Grid [108], the Yeast protein-protein interaction network [39], and the Facebook network [105]. The Facebook network topology is crawled from December 29th, 2008 to January 3rd, 2009. The basic statistics of these networks are shown in Table 5.1, and their degree distributions are shown in Fig. 5.3. We adopt the classic SI model, and suppose all infections are independent of each other. In simulations, we typically set the propagation probability on each

Table 5.1: Statistics of the Datasets Collected in Experiments.

| Dataset | Power Grid | Yeast | Facebook |
|---|---|---|---|
| # nodes | 4,941 | 2,361 | 45,813 |
| # edges | 13,188 | 13,554 | 370,532 |
| Average degree | 2.67 | 5.74 | 8.09 |
| Maximum degree | 19 | 64 | 223 |

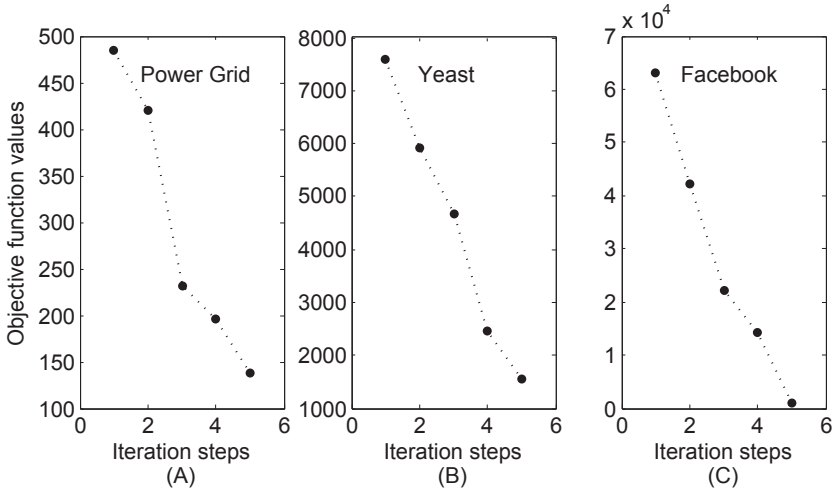



Figure 5.4: The monotonically decreasing of the objective functions.

edge, $\eta_{ij}$, uniformly distributed in (0, 1). As previous work [110, 127] has proven that the distribution of propagation probability will not affect the accuracy of the SI model, uniform distribution is enough to evaluate the performance of the proposed method. Similar propagation probability setting can be found in [59, 122] and [14]. We randomly choose a set of sources $S^*$, and let the number of diffusion sources $|S^*|$ range from 2 to 5. For each type of network and each number of diffusion sources, we perform 100 runs. The number of 100 comes from the discussion in the previous work of [127]. The implementation is in C++ and Matlab 2012b.

We firstly show the convergence of the proposed method. Fig. 5.4 shows the objective function values in iterations when the number of sources is 2 in the three real network topologies. It can be seen that the objective function is monotonically decreasing in iterations. Similar results can be found when we choose different number of sources. This, therefore, justifies Theorem 1 in Section 5.4.2.

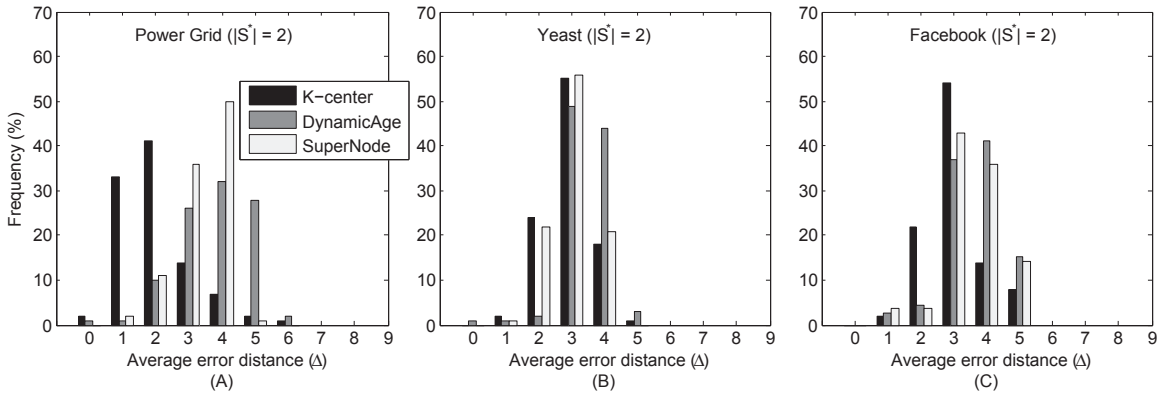



Figure 5.5: Histogram of the average error distances ($\Delta$) in various networks when $S^* = 2$. (A) Power Grid; (B) Yeast; (C) Facebook.

### 5.5.1 Accuracy of Identifying Rumor Sources

We compare the performance of the proposed K-center method with two competing methods: the dynamic age method [25] (see Section 3.2.5) and the multi-rumor-center (also called SuperNode) method [58] (see Section 3.2.3). To quantify the performance of each method, we firstly match the estimated sources $\hat{S} = \{\hat{s}_1, ..., \hat{s}_k\}$ with the real sources $S^* = \{s_1, ..., s_k\}$ so that the sum of the error distances between each estimated source and its match is minimized [58, 95]. The average error distance is then given by

$$\Delta = \frac{1}{|S^*|} \sum_{i=1}^{|S^*|} h(s_i, \hat{s}_i). \tag{5.5.1}$$

We expect that our method can accurately capture the real sources or at least a set of sources very close to the real sources (i.e., $\Delta$ is as small as possible).

The average error distances for the three real network topologies are provided in Table 5.2. From this table we can see that the proposed method outperforms the other two methods, that the estimated sources are closer to the real sources. To have a clearer comparison between our proposed method and the other two methods,

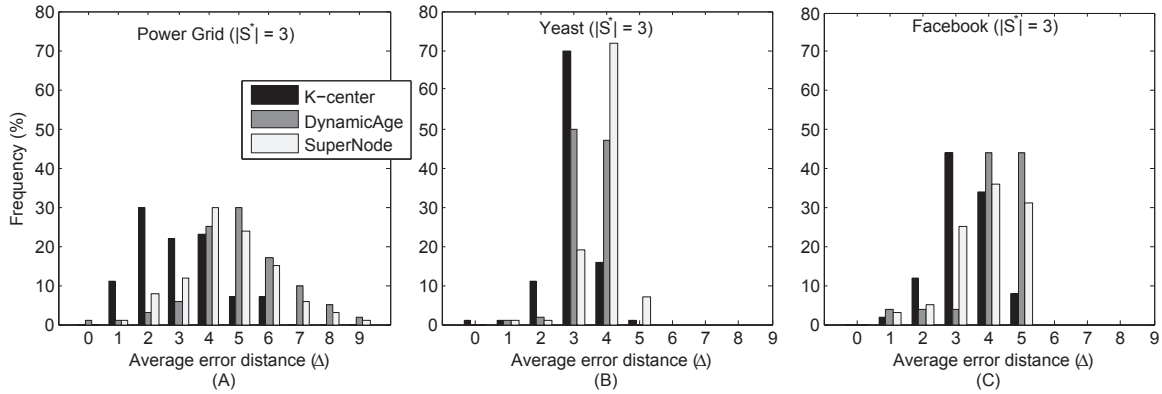



Figure 5.6: Histogram of the average error distances ($\Delta$) in various networks when $S^* = 3$. (A) Power Grid; (B) Yeast; (C) Facebook.

we show the histogram of the average error distances ($\Delta$) in Fig. 5.5 and Fig. 5.6, when $|S^*| = 2$ or 3, respectively. We can see that the proposed K-center method outperforms the others. When $|S^*| = 2$, the estimated sources are very close to

Table 5.2: Accuracy of Multi-source Identification.

| Experiment settings | | Average error distance $\Delta$ | | | Infection Percentage % |
|---|---|---|---|---|---|
| Network | $\lvert S^*\rvert$ | MRC | Dynamic age | K-center | |
| Power Grid | 2 | 3.135 | 3.610 | 1.750 | 96.290 |
| | 3 | 4.246 | 4.726 | 2.670 | 83.237 |
| | 4 | 5.331 | 6.027 | 3.240 | 78.322 |
| | 5 | 6.388 | 7.117 | 3.418 | 72.903 |
| Yeast | 2 | 2.700 | 3.175 | 2.680 | 89.606 |
| | 3 | 3.520 | 3.146 | 2.733 | 74.762 |
| | 4 | 3.525 | 3.077 | 2.962 | 70.599 |
| | 5 | 3.474 | 3.050 | 2.874 | 68.563 |
| Facebook | 2 | 3.433 | 3.950 | 3.215 | 81.776 |
| | 3 | 4.667 | 4.763 | 4.073 | 76.654 |
| | 4 | 5.120 | 5.762 | 4.137 | 69.762 |
| | 5 | 5.832 | 6.701 | 4.290 | 63.723 |

the real sources in the Power Grid, with the average error distances are generally 1-2 hops. However, the average error distances are around 3-4 hops when using the multi-rumor-center method, and around 3-5 hops when using the dynamic age method. For the Yeast network, the diffusion sources estimated by the proposed method are with an average of 2-3 hops away from the real sources. However, the sources estimated by using the multi-rumor-center method are averagely 2-4 hops away from the real sources, and averagely 3-4 hops away when using the dynamic age method. For the Facebook network, the proposed method can estimate the diffusion sources with an average of 2-3 hops away from the real sources. However, the estimated sources are averagely 3-4 hops away from the real sources when using the other two methods. Similarly, when $|S^*| = 3$, the diffusion sources estimated by the proposed method are much closer to the real sources in these real networks.

We have compared the performance of our method with two competeting methods. From the experiment results (Fig. 5.5 and Fig. 5.6, and Table 5.2), we see that our proposed method is superior to previous work. Around 80% of all experiment runs identify the nodes averagely 2-3 hops away from the real sources when there are two diffusion sources. Moreover, when there are three diffusion sources, there are also

Table 5.3: Accuracy of Spreading Time Estimation.

| Experiment settings | | Estimated spreading time | | |
|---|---|---|---|---|
| Network | $\|S^*\|$ | $T = 4$ | $T = 5$ | $T = 6$ |
| Power Grid | 2 | 4.020 ± 0.910 | 5.050 ± 1.256 | 5.627 ± 1.212 |
| | 3 | 4.085 ± 0.805 | 5.051 ± 0.934 | 6.006 ± 1.123 |
| Yeast | 2 | 4.600 ± 0.710 | 5.130 ± 0.469 | 5.494 ± 0.578 |
| | 3 | 4.534 ± 0.427 | 5.050 ± 0.408 | 5.447 ± 0.396 |
| Facebook | 2 | 4.380 ± 1.170 | 5.246 ± 0.517 | 5.853 ± 0.645 |
| | 3 | 4.417 ± 0.736 | 5.378 ± 0.645 | 5.738 ± 0.467 |

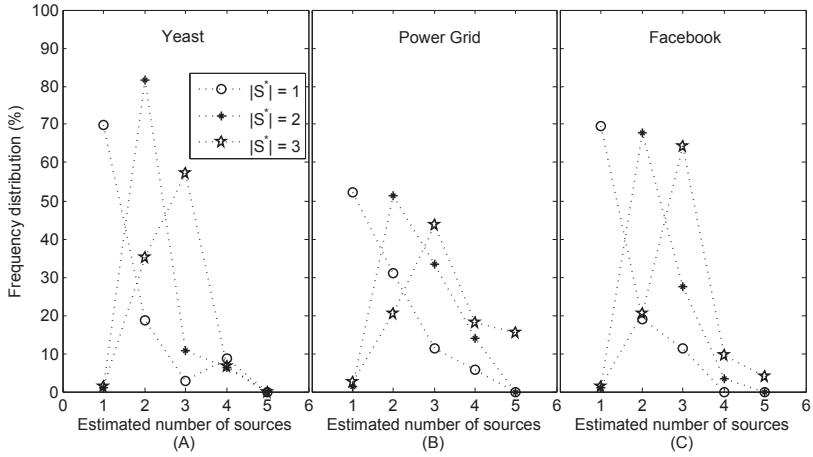



Figure 5.7: Estimate of the number of sources. (A) Yeast; (B) Power Grid; (C) Facebook.

around 80% of all experiment runs identifying the nodes averagely 3 hops away from the real sources.

### 5.5.2 Estimation of Source Number and Spreading Time

In this subsection, we evaluate the performance of the proposed method in estimating the number of sources and predicting diffusion spreading time.

Table 5.3 shows the means and the standard deviations of the time estimation in the three real networks when we vary the real spreading time $T$ from 4 to 6 and the number of sources $|S^*|$ from 2 to 3. Notice that the means of the estimated time are very close to the real spreading time under different experiment settings, and most results of the standard deviations are smaller than 1. This indicates that our method can estimate the real spreading time with high accuracy.

Fig. 5.7 shows the results in estimating the number of diffusion sources in different networks. We let the number of sources, $|S^*|$, range from 1 to 3. In Fig. 5.7, the horizontal axis indicates the estimated number of sources and the vertical axis

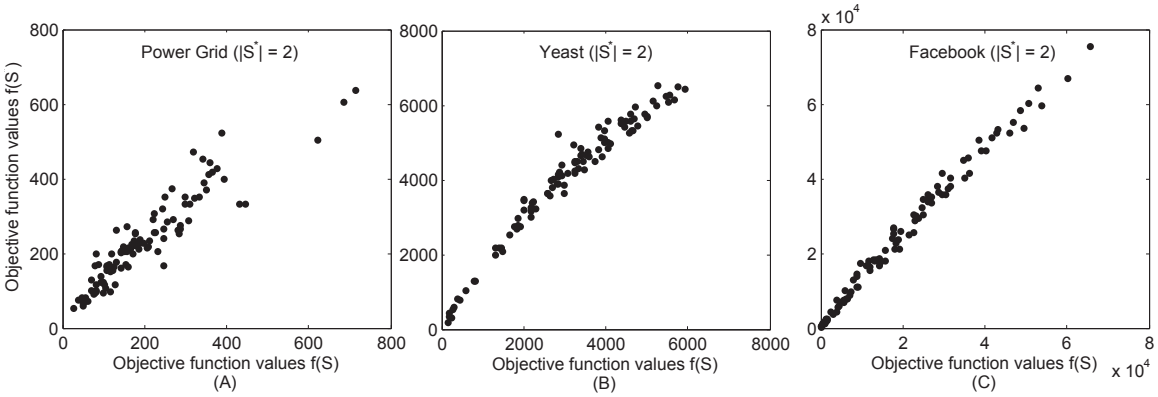



Figure 5.8: The correlation between the objective function of the estimated sources and that of the real sources when $S^* = 2$. (A) Power Grid; (B) Yeast; (C) Facebook.

indicates the percentage of experiment runs estimating the corresponding the number sources. For the Yeast network, we see that 70% experiment runs can accurately estimate the number of sources when $|S^*| = 1$. More than 80% of experiment runs can accurately estimate the number of sources when $|S^*| = 2$, and around 60% when $|S^*| = 3$. For the Power Grid network, it can be seen that around 50% of the total experiment runs can accurately detect the number of sources when $|S^*|$ ranges from 1 to 3. The accuracy is about 68% on Facebook when $|S^*|$ ranges from 1 to 3.

The high accuracy in estimating both the spreading time and the number of diffusion sources reflects the efficiency of our method from different angles.

### 5.5.3 Effectiveness Justification

We justify the effectiveness of the proposed K-center method from two different aspects. Firstly, we examine the correlation between the objective function values in Eq. (5.3.1) of the estimated sources and those of the real sources. If they are highly correlated with each other, the objective function in Eq. (5.3.1) will accurately describe

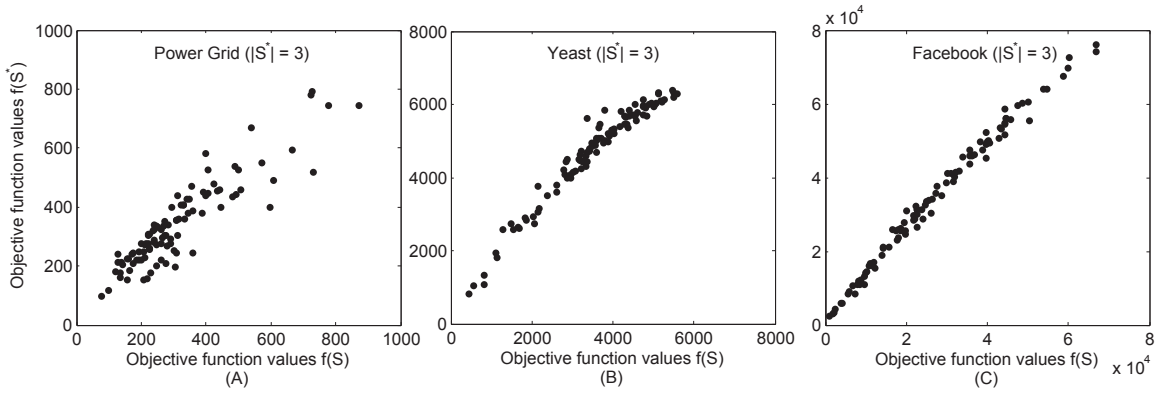



Figure 5.9: The correlation between the objective function of the estimated sources and that of the real sources when $S^* = 3$. (A) Power Grid; (B) Yeast; (C) Facebook.

the multi-source identification problem. Secondly, at each time tick, we examine the average effective distances from the newly infected nodes to their corresponding diffusion sources. The linear correlation between the average effective distances and the spreading time will justify the effectiveness of using effective distance in estimating multiple diffusion sources.

### Correlation Between Real Sources and Estimated Sources

We investigate the correlation between the estimated sources and the real sources by examining the correlation of their objective function values in Eq. (5.3.1). If the estimated sources are exactly the real sources, their objective function values $f$ should present high correlations.

Fig. 5.8 and Fig. 5.9 show the correlation results of the objective function values when $|S^*|$ is 2 or 3, respectively. We can see that their objective function values approximately form linear relationships. This means that the real sources and the estimated sources are highly correlated with each other. The worst results occur in

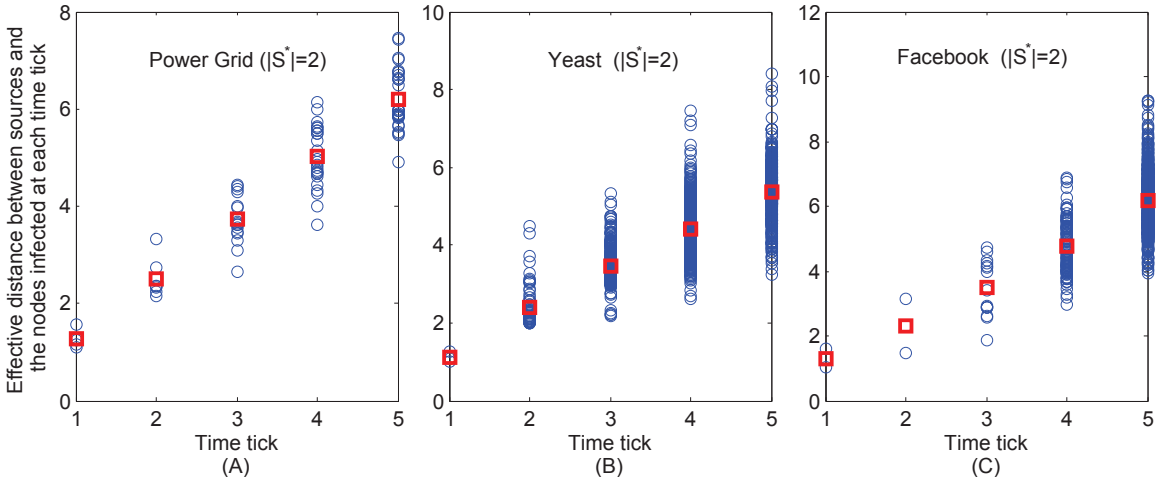



Figure 5.10: The effective distances between the nodes infected at each time tick and their corresponding sources when $S^* = 2$. (A) Power Grid; (B) Yeast; (C) Facebook.

Fig. 5.8(A) and Fig. 5.9(A) in the Power Grid network. However, the majority of the correlation results in these two figures still tend to be clustered in a line. The strong correlation between the real sources and estimated sources reflects the effectiveness of the proposed method.

### Average Effective Distance at Each Time Tick

We further investigate the correlation between the relative arrival time of nodes getting infected and the average effective distance from them to their corresponding sources. The experiment results in different networks when $|S^*|$ is 2 or 3 are shown in Fig. 5.10 and Fig. 5.11, respectively.

As shown in Fig. 5.10 and Fig. 5.11, at each time tick, the effective distance from the nodes infected at this time tick to their corresponding sources are indicated as blue circles. Their average effective distance to the corresponding sources at each time tick is indicated as red square. It can be seen that the average effective distance

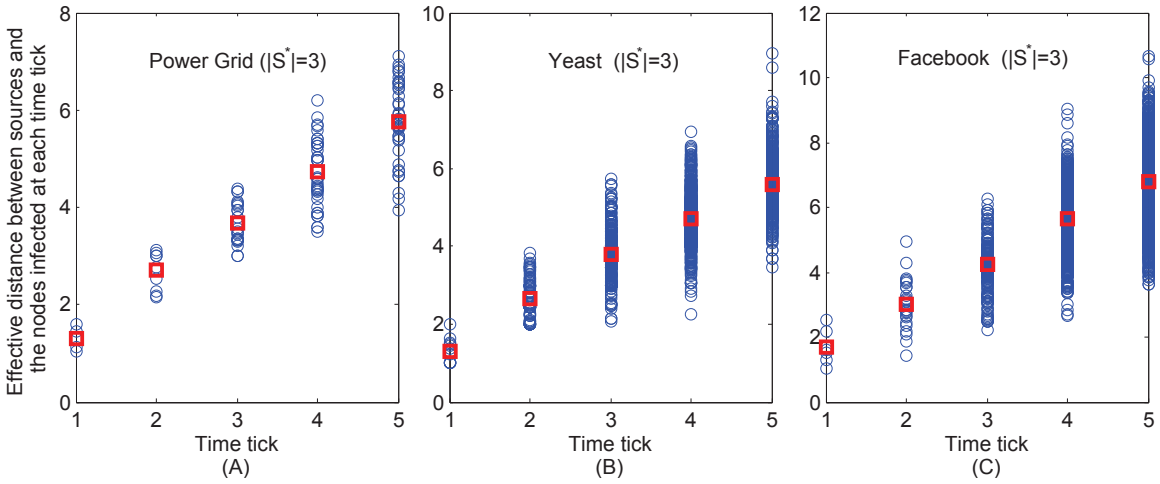



Figure 5.11: The effective distances between the nodes infected at each time tick and their corresponding sources when $S^* = 3$. (A) Power Grid; (B) Yeast; (C) Facebook.

is linear with the relative arrival time. This therefore justifies that the proposed K-center method is well-developed.

## 5.6 Conclusion and Discussion

In this chapter, we studied the problem of identifying multiple rumor sources in complex networks. Few of current techniques can detect multiple sources in complex networks. We used effective distance to transform the original network in order to have a clear understanding of the complex diffusion pattern. Based on the altered network, we derive a succinct formulation for the problem of identifying multiple rumor sources. Then we proposed a novel method that can detect *the positions of the multiple rumor sources*, estimate *the number of sources*, and predict *the spreading time* of the diffusion. Experiment results in various real network topologies show the outperformance of the proposed method than other competing methods, which justify

the effectiveness and efficiency of our method.

The identification of multiple rumor sources is a significant but difficult task. In this chapter, we have adopted the SI model with the knowledge of which nodes are infected and their connections. There are also some other models, such as the SIS and SIR. These models may conceal the infection history of the nodes that have been recovered. Therefore, the proposed method requiring a complete observation of network will not work in the SIS or SIR model. According to our study, we may need other techniques, *e.g.*, the network completion, to cope with the SIS and SIR models. Future research includes the use of different models. Moreover, in the real world, we may only obtain partial observations of a network. Thus, future work includes multi-source identification with partial observations. We may also need to take community structures into account to more accurately identify multiple diffusion sources.

# Chapter 6

# Identifying Rumor Sources in Large-scale Networks

## 6.1 Introduction

In this chapter, we develop a *community structure based approach* to efficiently identify diffusion sources in *large-scale networks*. With rapid urbanization progress worldwide, the world is becoming increasingly interconnected. This brings us great convenience in daily communication but also enables rumors to diffuse all around the world. The *huge scale* of the underlying networks and the *complex spatiotemporal rumor diffusions* make it difficult to develop effective strategies to quickly and accurately identify rumor sources and therefore eliminate the socio-economic impact of dangerous rumors.

In the past few years, researchers have proposed a series of methods to identify diffusion sources in networks. However, those methods are either with *high computational complexity* or with relative lower complexity but for *particularly structured networks* (*e.g.*, trees and regular networks). For example, the initial methods of rumor source identification are designed for *tree networks*, including the rumor center

method [96] and the Jordan center method [120]. Later, the constraints on trees were relaxed but with complete or snapshot observations through heuristic strategies, including Bayesian inference [5, 82], spectral techniques [25], and centralities methods [15]. Most of them are based on *scanning the whole network*. However, real networks are far more complex than tree networks and it is impractical to scan the whole network to locate the diffusion source, especially for large networks. Recently, Pinto et al. [82] proposed to identify rumor sources based on *sensor observations*. The proposed Gaussian method chooses sensors randomly or set up sensors on high degree nodes. In fact, the selection of sensors is crucial in identifying rumor sources since well chosen sensors can reflect the spreading direction and speed of the diffusion. Seo et al. [94] compared different strategies of choosing sensors, and concluded that high betweenness or degree sensors are more efficient in identifying rumor sources. They proposed a *Four-metric source estimator* which is also based on scanning the whole network and view the diffusion source as the node which not only can reach the infected sensors with the minimum sum of distances but also is the furthest away from the non-infected sensors. In a nutshell, current methods are not suitable for large-scale networks due to the expensive computational complexity and the large scale of real networks. Readers could refer to [40] for a detailed survey in this area.

In this chapter, we propose a *community structure based approach* to identify diffusion sources in *large-scale networks*. It not only addresses the *scalability* issue in this area, but also shows significant advantages. Firstly, to effectively set up sparse sensors, we detect the community structure of a network and choose the community bridge nodes as sensors. According to the earliest infected bridge sensors, we can easily determine the very first infected community where the diffusion started and spread

out to the rest of the network. Consequently, this narrows the suspicious sources down to the very first infected community. Therefore, this overcomes the scalability issue of current methods. According to a fundamental property of communities that links inside are much denser than those connecting outside nodes, bridge sensors will be very sparse. Secondly, to accurately locate the diffusion source from the first infected community, we use the intrinsic property of the diffusion source that the relative infection time of any node is linear with its effective distance from the source. The effective distance between any pair of nodes is based on not only the number of hops but also the propagation probabilities along the paths between them [12]. It reflects the idea that a small propagation probability between nodes is effectively equivalent to a large distance between them, and vice versa. Finally, we use correlation coefficient to measure the degree of linear dependence between the relative infection time and effective distances for each suspect, and consider the one that has the largest correlation coefficient as the diffusion source.

The main *contribution* of this chapter is three-fold.

- We address the scalability issue in source identification problems. Instead of randomly choosing sensors or setting up high centrality nodes as sensors in previous methods, we assign sensors on community bridges. According to the infection time of bridge sensors, we can easily narrow the suspicious sources down to the very first infected community.

- We propose a novel method which can efficiently locate diffusion sources from the suspects. Here, we use the intrinsic property of the real diffusion source that the effective distance to any node is linear with the relative infection time of that node. The effective distance makes full use of the propagation probability

and the number of hops between node pairs, which dramatically enhances the effectiveness of our method.

- We evaluate our method in two large networks collected from Twitter. The experiment results show significant advantages of our method in identifying diffusion sources in large networks. Especially, when the average size of communities shrinks, the accuracy of our method increases dramatically.

The rest of this chapter is organized as follows. Preliminary knowledge about community structure is introduced in Section 6.2. Section 6.3 presents the proposed community structure based approach. In Section 6.4, we evaluate the proposed approach in large networks and we compare it to many competeting methods in Section 6.4.4. Section 6.5 concludes some remarks in this chapter.

## 6.2 Community Structure

In general, *communities* are groups of nodes sharing common properties or corresponding to functional units within a networked system. Many networks of interest, including social networks, computer networks, and transportation networks, are found to divide naturally into communities, where the links inside are much denser than those connecting this set and the rest of the network [32]. Recent research results show that community structure can dramatically affect the behavior of dynamical processes of complex networks [68].

Past work on methods for discovering communities in networks divides into two principal lines of research, both with long histories. The first, generally called ***hard-partition***, assumes that communities of complex networks are *disjoint*, placing each

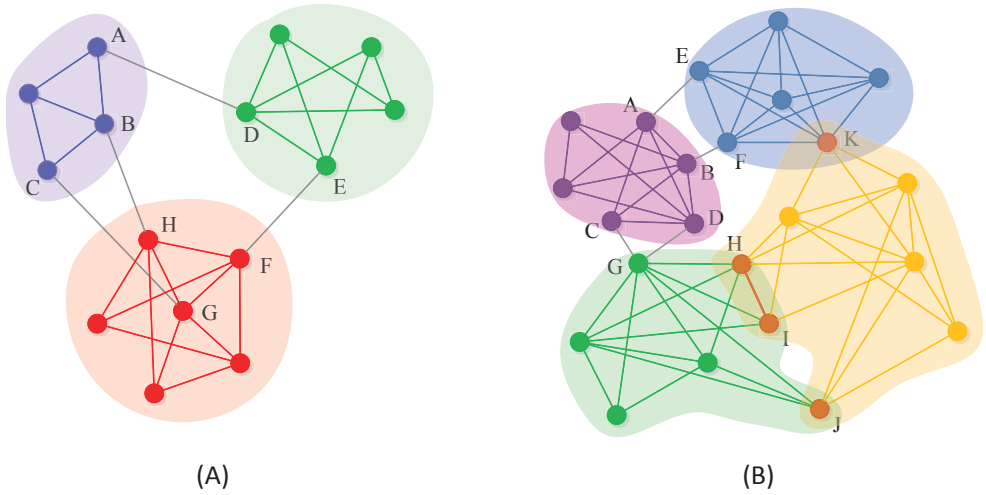



Figure 6.1: Illustration of network communities and community bridges. (A) Separated communities. Community bridges are the nodes associated with between-community edges, *e.g.*, nodes $A$ and $D$ connecting the blue community and the green community. (B) Overlapping communities. Community bridges are not only the nodes associated with between-community edges but also the nodes shared by different communities, *e.g.*, nodes $H$, $I$ and $J$ shared by the green community and the yellow community.

node in only one community and all communities are non-overlapped. Algorithms include division based on betweenness [32], information theory [88], modularity optimization [68], and some others [90]. However, many real networks are characterized by well-defined statistics of ***overlapping communities*** [76]. For example, in collaboration networks an author might work with researchers in many groups, and in biological networks a protein might interact with many groups of proteins. Algorithms in detecting overlapping communities include techniques based on $k$-clique [76], link clustering [2], and some others [75]. Fig. 6.1 shows two examples to illustrate separated communities and overlapping communities. To demonstrate the robustness of the results across different types of communities, we will apply both separated and

overlapping community detection methods on various real networks− InfoMap [89] and Link Clustering [2].

In this chapter, we will use community structures of networks to effectively assign sensors. More specifically, we set sensors on *community bridges*. Community bridges are nodes shared by two or more different communities or associated with inter-community links (See Fig. 6.1). This is fundamentally different from previous methods which choose high centrality nodes as sensors or even randomly set up sensors.

## 6.3 Community-based Method

In this section, we first introduce an effective strategy to set up sensors. Then, we derive an efficient method to detect sources according to sensors' sparse observations. Finally, we analyze the computational complexity of our method and compare to that of current methods.

### 6.3.1 Assigning Sensors

To identify diffusion sources under sensor observations, it is very critical to assign sensors properly. To effectively set up sensors in a network, we need to choose the nodes that are very important in diffusion processes. From Section 6.2, we see that community bridges play a crucial role in transmitting information from one community to another. They can reflect the spreading direction and speed of diffusion. Thus, we choose community bridges as sensors.

To assign sensors on community bridges, we first need to detect the community

structure of a network. According to Section 6.2, community structures can generally be divided into two categories: separated communities and overlapping communities. For separated communities, community bridges are the nodes associating with the inter-community edges. For example in Fig. 6.1 (A), the green community and red community are connected by bridges $E$ and $F$, and the bridges $B$, $H$, $C$ and $G$ connect the blue community and the red community. For overlapping communities, community bridges correspond not only to the nodes associated with the inter-community edges, but also the nodes shared by different communities. For example in Fig. 6.1 (B), the green community and the yellow community are connected by shared bridges $H$, $I$ and $J$, and bridges $G$, $C$ and $D$ connect the green community and the purple community.

When we assign sensors on community bridges, we need to pay attention to the number of sensors. The more sensors we set up, the more information we will collect from them. However, in the real world, setting up more sensors will require more money to buy equipments and more labor to maintain them. Generally, we can control the number of bridges by regulating the average size of communities. The larger the average size of communities, the smaller the number of community bridges, and vice versa. Here are two extreme examples to explain this. (i) If we divide a network into two communities, and choose one node as the first community and all the remaining nodes in the second community, the number of bridge nodes will be $d + 1$, where $d$ is the degree of the node in the first community. (ii) If we set every single node as a community, the number of bridges will be the number of nodes in the whole network. Furthermore, the number of bridges will be very small because of the intrinsic property of communities that the links between communities is much

sparser than those within communities. In Section 6.4.2, we will analyze in detail the influence of the average size of communities in detecting diffusion sources.

Compared with the existing sensor selection methods, which randomly choose sensors or select high centrality nodes as sensors, the proposed community structure based sensor selection method can additionally reflect the diffusion direction and speed. We will compare different sensor-selection methods in various real networks in Section 6.4.4.

### 6.3.2 Community Structure Based Approach

The proposed community structure based approach consists of two steps. In ***the first step***, we determine the very first infected communities. Given a diffusion process running for some time in a network, we obtain sparse observations from the sensors assigned by the scheme in the previous subsection. Assume there are $k$ sensors having been infected, denoted as $O = \{o_1, ..., o_k\}$, and $\{t_1, ..., t_k\}$ represents the time at which the infection arrives at these sensors. Then, according to the first infected sensor(s), we can determine which community started the diffusion since the diffusion has to go through community bridges to infect other communities. For example in Fig. 6.1 (A), if sensors $\{H, F, E, B\}$ are observed as infected and node $H$ is the first infected one, we can determine that the diffusion started from the red community. In Fig. 6.1 (B), if sensors $\{K, F, H, J, G\}$ are observed infected and node $K$ is the first infected one, we can determine that the diffusion could have started from the blue community or the yellow community. We denote the set of nodes in the first infected communities as

$$U = \{u_1, u_2, ..., u_m\}. \tag{6.3.1}$$

Since we do not have an absolute time reference, we have knowledge only about the relative infection time. Choosing an arbitrary infected sensor, say $o_1$, as the reference node, we can obtain the relative infection time of all the infected sensors as in

$$\tau = \{0, t_2 - t_1, ..., t_k - t_1\}. \tag{6.3.2}$$

In ***the second step***, we investigate each suspect in the set $U$ and identify the real diffusion source. According to the properties of effective distance in , we know that the relative infection time of any infected node is linear with its effective distance from the real diffusion source. Therefore, to identify the diffusion source, we aim to find the suspect with the best linear correlation between sensors' relative infection time and their effective distances from this suspect. Here, we use the correlation coefficient, which is widely used as a measure of the degree of linear dependence between two variables [50]. The correlation coefficient between two vectors $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ is defined as,

$$e = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})}}, \tag{6.3.3}$$

where $\bar{x}$ and $\bar{y}$ are the means of $x_i$ and $y_i$, respectively. The correlation coefficient ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between $\mathbf{x}$ and $\mathbf{y}$ perfectly, with all data points lying on a line for which $\mathbf{y}$ increases as $\mathbf{x}$ increases. A value of -1 implies that all data points lie on a line for which $\mathbf{y}$ decreases as $\mathbf{x}$ increases. A value of 0 implies that there is no linear correlation between the variables. Therefore, we need to find a suspect with the maximum correlation coefficient. More precisely, we aim to find a suspect in $U$ to maximize Eq. (6.3.3) in terms of the relative infection time of sensors and their effective distance from the suspect. The detailed process of the proposed approach is given in

Algorithm 1.

Compared with current methods of identifying diffusion sources, the proposed approach is superior, as many of the existing methods ignore the propagation probabilities [40]. The proposed method utilizes the effective distance between nodes which precisely reflects not only the propagation probability but also the number of hops between nodes (See the definition of effective distance in Section 5.2.2). This makes our algorithm more accurate and effective. The comparison of our method to many competeting methods is shown in Section 6.4.4.

### 6.3.3 Computational Complexity

In this subsection, we analyze the computation complexity of the proposed method and compare it with other existing methods of identifying diffusion sources based on sensor observations, including the Gaussian method [82], the Monte Carlo method [1], and the Four-metric method [94].

From Algorithm 1, we see that the computation of our method is dominated by Step 2 of calculating the correlation coefficient $e$ for each suspect $u_i$ in the very first infected community $U$. More specifically, the majority of computation is in the calculation of effective distance between $u_i$ and any infected sensor $o_j$ ($\in \{o_1, o_2, ..., o_k\}$). Here, we use Dijkstra's algorithm [30] to compute the shortest paths (*i.e.*, the effective distances) to all infected sensors from each $u_i$. Dijkstra's algorithm requires $O(M + N\log N)$ computations to find the shortest paths from one node to every other node in a network, where $M$ is the number of edges and $N$ is the number of nodes in the network. However, in Algorithm 1, we only need to calculate the effective distance between each suspect $u_i$ and any infected sensor $o_j$, *i.e.*,

$[D(u_i, o_1), D(u_i, o_2), ..., D(u_i, o_k)]$ in Eq. (6.5.1). Therefore, the complexity will be far less than $O(M + N\log N)$. Suppose the average size of communities in the network is $m$ and the number of infected sensors is $k$. Then, the computational complexity of the proposed method is far less than $O(L(M + N\log N))$, where $L = \min\{k, m\}$. Thus, if the average community size is smaller, it requires less time to identify the diffusion source.

Current methods are far more complex than the proposed method. They need to scan the whole network, and calculate the shortest path from each sensor to any other node. For example, the computational complexity of the Gaussian method [82] is $O(N^3)$ since it requires constructing the BFS tree rooted at each node in a network, and it also needs to calculate the inverse of the covariance matrix for each BFS tree. The computational complexity of the Monte Carlo method [1] is $O(k(M+N\log N)/\epsilon^2)$, where $k$ is the number of infected sensors. The majority of the computation is in calculating the shortest paths from an arbitrary node $i$ to all the sensors in order to sample the infection time of all sensors assuming that node $i$ is the diffusion source. By the central limit theorem, $O(1/\epsilon^2)$ samples are needed to achieve an error $o(\epsilon)$. For the Four-metric method [94], the majority of the computation is also in computing the lengths of the shortest paths from each node to all the sensors, both infected and non-infected. Thus, the computational complexity of this method is $O(n(M + N\log N))$, where $n$ is the number of sensors.

Compared with these existing methods of identifying diffusion sources based on sensor observations, we see that the computational complexity of the proposed method is much less than that of current methods. Furthermore, the proposed method takes advantages of the relative infection time of sensors, the propagation probabilities and

the number of hops between nodes. However, the existing methods either require the generation of the infection time of each sensor (*e.g.*, the Gaussian and Monte Carlo methods) or ignore the propagation probabilities (*e.g.*, the Four-metric method) between nodes. Thus, the proposed method is superior and is able to work in large networks.

## 6.4 Evaluation

The proposed community structure based approach is evaluated in two real-world large networks, the Retweet network and the Mention network collected from Twitter, which were also used in the work of [111]. These two networks were constructed from the tweets collected by using the Twitter streaming API during Mar 24 and Apr 25, 2012. The basic statistics of these two networks are listed in Table 6.1. In these two networks, only reciprocal communications are kept as network edges, as bi-directional communications reflect more stable and reliable social connections. Fig. 6.2 shows the degree distribution of these two networks. We can see that node degrees of these two networks follow the power law distribution. The number of contacts (retweets or mentions) between any two neighbors is set as the weight of the edge between them. Based on the number of contacts between any two neighbors, we also generate the

Table 6.1: Statistics of Two Large Networks in Experiments.

| Dataset | Mention | Retweet |
|---|---|---|
| # nodes | 300,197 | 374,829 |
| # edges | 1,048,818 | 598,487 |
| Average degree | 3.49 | 1.60 |
| Maximum degree | 124 | 178 |

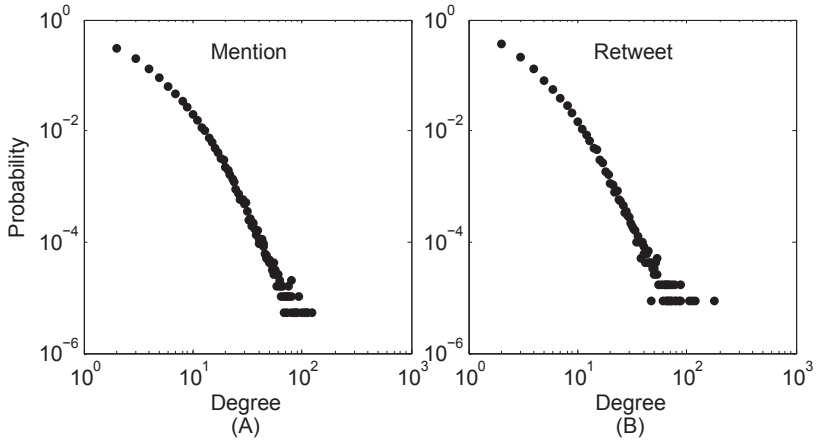



Figure 6.2: Degree distribution of the two large networks. (A) The Mention network; (B) The Retweet network.

propagation probability between them, as in

$$p_{ij} = \min\left\{1, \frac{z}{2 * \mu}\right\}, \tag{6.4.1}$$

where $z$ is the number of contacts between node $i$ and $j$, and $\mu$ is the median of contacts between neighbors.

To demonstrate the robustness of the proposed method across different types of community structure, we apply separated (InfoMap [89]) and overlapping (Link Clustering [76]) community detection methods on these two networks. The Infomap method shows communities of a network in a hierarchical structure from which we can choose different levels of communities. In each level, the number of communities will be different. The deeper the level is, the more communities we will obtain and the smaller the average size of each community will be. In our experiments, we typically choose the second, third and fourth-level communities, denoted by $\beta = 2$, 3 and 4. On the other hand, we can adjust the parameter $\alpha$ in the Link Clustering method to regulate the number of communities of a network. The larger $\alpha$ is, the

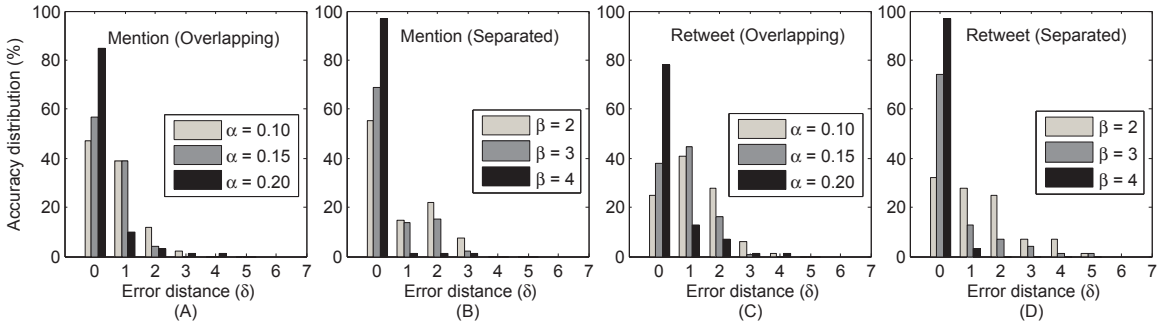



Figure 6.3: The accuracy of the proposed method in identifying diffusion sources in two real large networks. (A) and (C) show the accuracy of our method in the Mention network and the Retweet network having overlapping-community structure with parameter $\alpha \in \{0.10, 0.15, 0.20\}$. (B) and (D) show the accuracy of our method in these networks having separated-community structure with parameter $\beta \in \{2, 3, 4\}$.

more communities we obtain, and similar to the previous method, the smaller the average size of the communities will be. We typically set $\alpha = 0.10$, 0.15 and 0.20 in our experiments. In each experiment, we randomly choose a diffusion source in each run over 100 runs. The number of 100 comes from the discussion in the previous work of [126]. The implementation is in conducted in C++.

### 6.4.1 Identifying Diffusion Sources in Large Networks

We show the accuracy of the proposed method in identifying diffusion sources in this subsection. We use $\delta$ to denote the error distance (*i.e.*, the number of hops) between a real diffusion source and an estimated source. Ideally, we have $\delta = 0$ if our method accurately captures the real source. In practice, we expect that our method can accurately capture the real source or a node very close to the real source.

Fig. 6.3 shows the accuracy of our method in the Mention network and the Retweet

network associated with overlapping-community structure and separated-community structure. Overall, we see that the proposed method performs very well in these two large networks with the majority of the experiments able to precisely identify the diffusion sources. Especially with a large $\alpha$ or $\beta$, the proposed method performs better in identifying diffusion sources, as the number of communities becomes larger, and the average size of communities becomes smaller. Fig. 6.3 (A) shows the experiment results in the Mention network with overlapping-community structure. When $\alpha$ is 0.10, around 48% of the experiment runs can accurately identify the diffusion sources. When $\alpha$ increases to 0.15 (equivalently, the average size of communities becomes small), the accuracy of our method increases to about 57%. When $\alpha$ increases to 0.20, more than 83% of the experiment runs can accurately identify the real sources. Fig. 6.3 (B) shows the experiment results in the Mention network with separated-community structure. Similar to the results in the overlapping-community structure, when $\beta$ is 2, around 52% of the experiment runs can precisely identify the real diffusion sources. When $\beta$ increases to 3 (*i.e.*, the average size of communities becomes small), the accuracy of our method increases to around 70%. When $\beta$ increases to 4, our method achieves an accuracy of around 98%, which means only a few runs could not identify the real sources. Similar results can be found in the Retweet network in Fig. 6.3(C) and Fig. 6.3(D).

Furthermore, we notice that the average distance between the estimated sources and the real sources is very small. For both networks, from Fig. 6.3 we see that the average error distance is within 1-2 hops. That is to say, even when the proposed method does not accurately identify the real source, it is on average within a radius of 1-2 hops from the estimated source. In addition, from Fig. 6.3 we see that the

maximum error distance is also very small (on average 5 hops). Compared with the existing methods, which have low accuracy and expensive computational complexity [40], the proposed method shows significantly higher performance in identifying diffusion sources in large networks.

To summarize, our method performs very well in large networks associated with either overlapping or separated community structures. Especially, when a network is associated with a small average community size, our method can accurately identify diffusion sources.

### 6.4.2 Influence of the Average Community Size

From the previous subsection, we notice that the accuracy of the proposed method increases when the parameter $\alpha$ or $\beta$ becomes large. Equivalently, the performance of the proposed method improves when the average size of communities becomes small. In order to analyze the influence of the average size of communities in the accuracy of our method, we investigate the number of communities, bridges and suspects when we change the parameters in the separated-community and overlapping-community detection methods. More specifically, we let the parameter $\beta$ range from 2 to 4 for the Infomap method of detecting separated community structure, and we let the parameter $\alpha$ range from 0.10 to 0.20 for the Link Clustering method of detecting overlapping community structure.

The distribution of the community sizes of the previous two networks under different parameter settings is shown in Fig. 6.4. Overall, we can see that the community size shows power law distribution, *i.e.*, a few communities are of a significantly larger

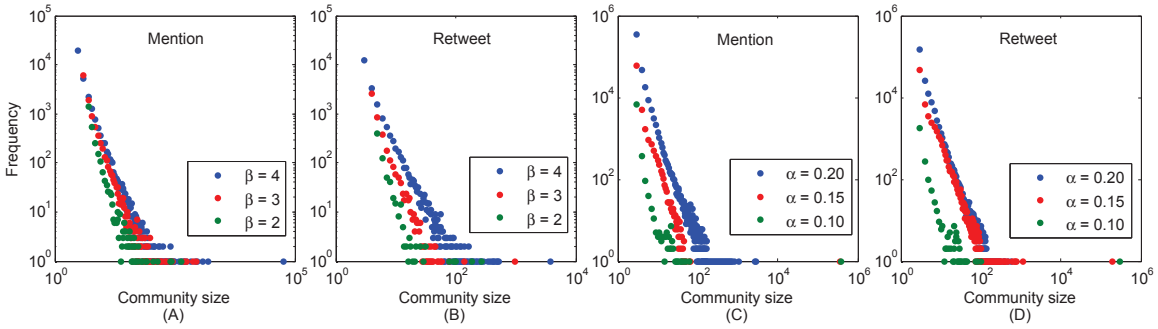



Figure 6.4: Community size distribution under different parameter setting. (A) and (B) show the community size distribution in the Mention network and the Retweet network having separated community structure with $\beta \in \{2, 3, 4\}$; (C) and (D) show the community size distribution in the Mention network and the Retweet network having separated community structure with $\alpha \in \{0.10, 0.15, 0.20\}$.

size but the majority of the communities are of a smaller size. Furthermore, the number of communities decreases when the parameter $\alpha$ or $\beta$ becomes smaller (compare the density of blue and green dots in Fig. 6.4). The detailed statistics of the community structures of these two networks derived by setting different parameters are shown in Table 6.2. For the Retweet network, when $\beta = 2$, there are 852 communities, 8,422 bridge nodes, an average of 2,158 suspects, and the average error distance between the estimated sources and the real sources is 1.77. When $\beta$ increases to 3, the average error distance decreases to 1.36 and the number of suspects shrinks to 925, while the number of communities and bridges increases. When $\beta$ increase to 4, the average error distance decreases to 0.48 and the number of suspects shrinks to 153, while the number of communities bridges becomes larger. We notice that when the parameter $\beta$ becomes large, the number of communities rises, which leads to a decrease in the average size of communities. Consequently, more bridges are needed

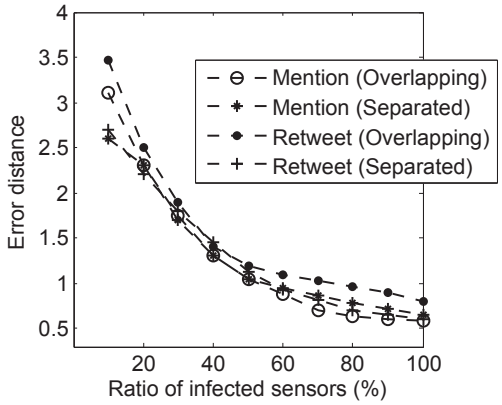



Figure 6.5: The influence of the ratio of infected sensors in the accuracy of our method in the two real networks.

to connect communities. We then can obtain more information from bridge sensors. Thus, we see that the average error distance between the real sources and the estimated sources becomes smaller. Similar results can be found in both networks with overlapping-community structure. When the parameter $\alpha$ increases, the number of bridges increases, and therefore, the average error distance decreases.

Table 6.2: Statistics of Network Communities and Accuracy of Our Method.

| Experiment settings | | Infomap | | | Link clustering | | |
|---|---|---|---|---|---|---|---|
| | | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\alpha = 0.10$ | $\alpha = 0.15$ | $\alpha = 0.20$ |
| Retweet | # communities | 852 | 2,684 | 21,300 | 2,332 | 7,166 | 34,559 |
| | # bridges | 8,422 | 13,078 | 36,558 | 7,470 | 12,281 | 76,537 |
| | # suspects | 2,158 | 925 | 153 | 5,256 | 1,380 | 157 |
| | Error distance | 1.77 | 1.36 | 0.48 | 1.83 | 1.18 | 0.96 |
| Mention | # communities | 588 | 5,001 | 32,355 | 2,754 | 7,812 | 21,187 |
| | # bridges | 10,165 | 18,974 | 57,335 | 8,306 | 14,072 | 81,287 |
| | # suspects | 3,525 | 1,169 | 318 | 6,132 | 1,247 | 297 |
| | Error distance | 1.37 | 0.81 | 0.50 | 1.26 | 0.69 | 0.58 |

In the real world, it requires a lot of money and energy to set up sensors and maintain them. Hence, we need to choose as few sensors as possible and start to identify diffusion sources when partial sensors get infected. Here, we select a moderate-size set of sensors and then analyze the accuracy of our method when only a small ratio of sensors are infected (see Fig. 6.5). More specifically, we choose $\beta = 2$ for the Infomap method and $\alpha = 0.10$ for the Link Clustering method. Fig. 6.5 shows the average error distance between the real sources and the estimated sources when the ratio of infected sensors ranges from 10% to 100%. We see that when more than 30% of sensors are infected, our method can identify a node on average less than 2 hops away from the real source. When there are more than 50% of sensors are infected, the average error distance between the real source and the estimated source is approximately 1. Therefore, the proposed method can identify diffusion sources with high accuracy even if only a small ratio of sensors are infected.

From Figs. 6.4, 6.5 and Table 6.2, we see that the performance of the proposed method improves when the average community size becomes smaller. Even if the average community size is large and only a small ratio of sensors are infected, our method can still accurately identify the real diffusion source or a node very close to the real diffusion source.

### 6.4.3 Effectiveness Justification

In the second step of the proposed method, we utilize the linear correlation between the relative infection time of any sensor and its effective distance from the diffusion source. The suspect with the highest correlation coefficient is considered as the diffusion source. In order to justify the effectiveness of the proposed method, we examine

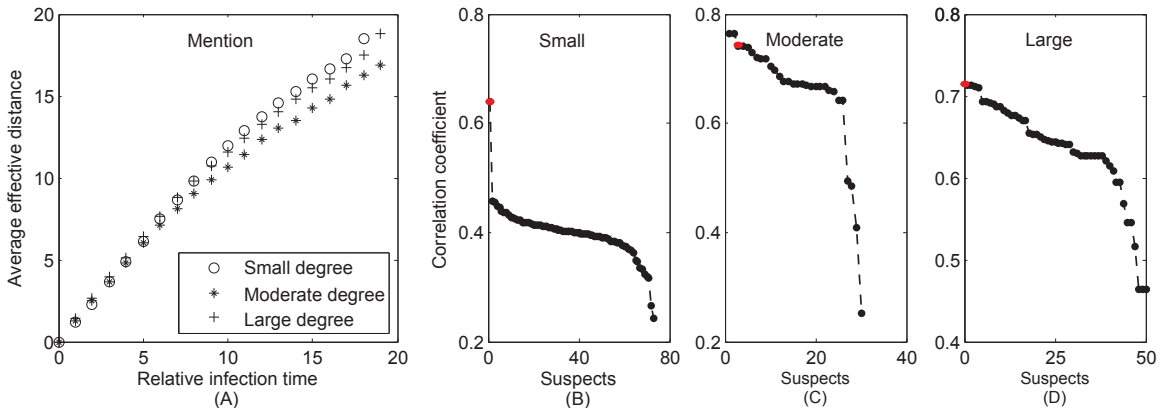



Figure 6.6: Justification of our method on the Mention network. (A) Linear correlation between the relative infection time of sensors and their average effective distance from the diffusion source. Specifically, we let the diffusion start from sources with different degrees: small degree, moderate degree and large degree. (B), (C) and (D) show the correlation coefficient value for each suspect.

the relationship between the relative infection time of any infected node and its effective distance from the diffusion source, especially when the diffusion starts from sources of different degrees.

In the previous two networks, we let the diffusion start from a small, moderate and large degree source respectively, and compare the correlation coefficient of the real source and that of all the suspects. Fig. 6.6 shows the experiment results on the Mention network. From Fig. 6.6 (A), we can see that, with the diffusion starting from sources of different degrees, the relative infection time of infected nodes is linear with their average effective distance from the diffusion source. We notice that when the diffusion starts from a large degree source, the scatter plot begins to curve beginning at time tick 15. According to our investigation, almost all of the nodes have been infected by time tick 15. Then, in the remaining time, only the nodes which refused

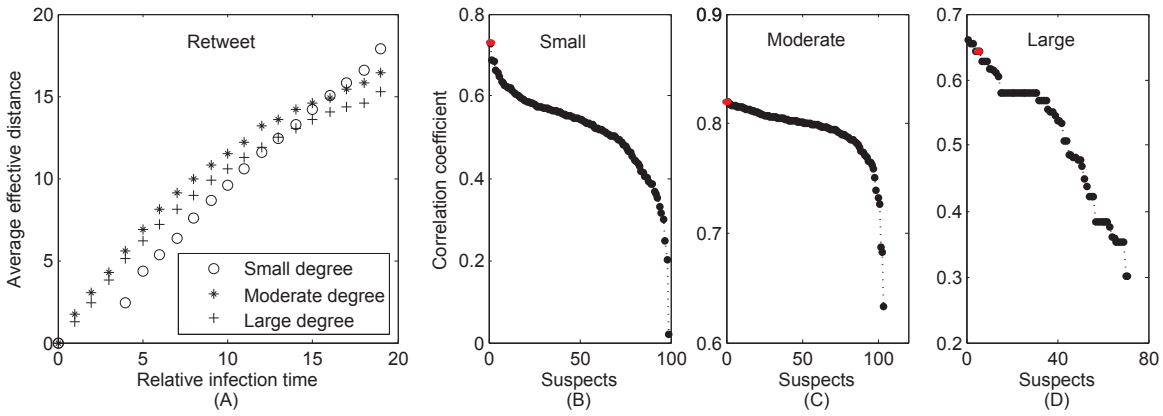



Figure 6.7: Justification of our method on the Retweet network. (A) Linear correlation between the relative infection time of sensors and their average effective distance from the diffusion source. (B), (C) and (D) show the correlation coefficient value for each suspect.

to be infected before time tick 15 can get infected. However, their smallest number of hops from the diffusion source is fixed. Therefore, according to Eq. (5.2.5), its effective distance from the diffusion source will be relatively short. In Fig. 6.6 (B), (C) and (D), we show the correlation coefficient of all suspects. It can be seen that the real sources (see the red dots) have a high correlation coefficient whenever the diffusion starts from a source of small, moderate, or large degree. Fig. 6.7 shows the experiment results on the Retweet network. Similar to the results on the Mention network, the relative infection time of any infected node is linear to its effective distance from the diffusion sources of different degrees. When the diffusion starts from a large degree source, their relation starts to curve towards the end. This is because almost all of the nodes have been infected by time tick 13. Fig. 6.7 (B), (C) and (D) show the correlation coefficient of all suspects. We can see that the real sources all have high correlation coefficient.

In summary, the linear correlation between relative infection time of nodes and

their average effective distance from diffusion source justifies the effectiveness of our proposed method.

### 6.4.4 Comparison with Current Methods

In this section, we compare the proposed community structure based approach with three competeting methods of identifying diffusion sources in networks based on sensor observations. They include:

- The Gaussian method [82],
- The Monte Carlo method [1], and
- The Four-metric method [94].

Readers can refer to Section 3.2 for more details. However, according to Section 6.1, these methods are all susceptible to the ***scalability issue*** because they need to scan every node in a network, which leads to very high computational complexity (see Section 6.3.3). Especially, for the Gaussian method, which is designed for tree networks, it needs to construct the BFS trees rooted at each node in a general network and the inverse of the covariance matrix for each BFS tree. Therefore, these methods are too computationally expensive to be applied in large networks. In addition, among these methods only the Four-metric method investigated and compared different sensor selection methods. Both the Gaussian method and the Monte Carlo method set up sensors on high degree nodes or even randomly choose nodes as sensors.

In the following, we first choose four relatively small networks to compare the performance of the proposed method to that of the three methods. Then, we introduce

two well studied methods to select sensors for the three methods. Finally, we present the detailed comparison results.

**Four Relatively Small Networks**

In order to compare with the three competeting methods, we choose four relatively small networks:

- The Western U.S. Power Grid network [108],
- The Yeast protein protein interaction (PPI) network [39],
- Mention: the network of political communication between Twitter users [16].
- Retweet: the network of political communication between Twitter users [16].

The political communication dataset describes two networks of political communication between users of the Twitter social media platform (mention, and retweet) in the six weeks prior to the 2010 U.S. Congressional midterm elections. We denote the network of political retweets as *Political Retweet*, and the network of political mentions as *Political Mention*. Statistics of these networks are given in Table 6.3.

Table 6.3: Statistics of Four Relative Small Networks in Experiments.

| Dataset | Mention | Retweet | Power grid | Yeast |
|---|---|---|---|---|
| # nodes | 7,175 | 18,470 | 4,941 | 2,361 |
| # edges | 28,473 | 121,043 | 13,188 | 13,554 |
| Average degree | 3.97 | 6.55 | 2.67 | 5.74 |
| Maximum degree | 425 | 1,017 | 19 | 64 |
| # communities ($\beta = 2$) | 636 | 1,462 | 37 | 2,271 |
| # bridges ($\beta = 2$) | 2,340 | 5,207 | 228 | 798 |
| # communities ($\alpha = 0.10$) | 1,343 | 2,528 | 322 | 297 |
| # bridges ($\alpha = 0.10$) | 2,245 | 4,966 | 689 | 513 |

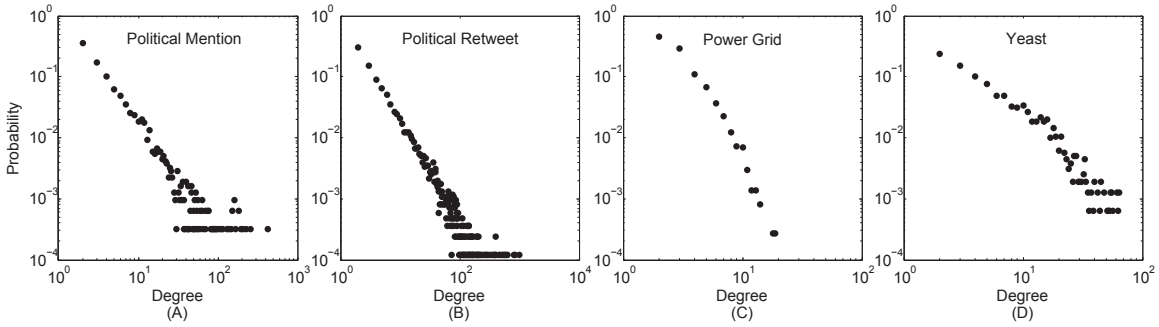



Figure 6.8: Degree distribution of the four networks. (A) Political Mention; (B) Political Retweet; (C) Power Grid; (D) Yeast PPI network.

### Sensor-selection Methods

Researchers in the work of [94] investigated various strategies to select sensors. They conclude that high degree or high betweenness sensors are more efficient in identifying rumor sources than other selected sensors. Therefore, we utilize these two strategies to set up sensors for the three competeting methods.

- High-degree sensors [15]: we sort the nodes according their degree and choose the high-degree nodes as sensors.
- High-betweenness sensors [68]: we sort the nodes according to their betweenness centrality value, an choose the high-betweenness nodes as sensors.

Fig. 6.8 shows the degree distributions of the four networks. We can see that the two political communication networks show power law degree distribution, and the Power Grid and the Yeast PPI network tend to be exponential. The betweenness distributions of these networks are shown in Fig. 6.9. Correspondingly, the betweenness distributions also show the scale-free or exponential phenomenon of these networks.

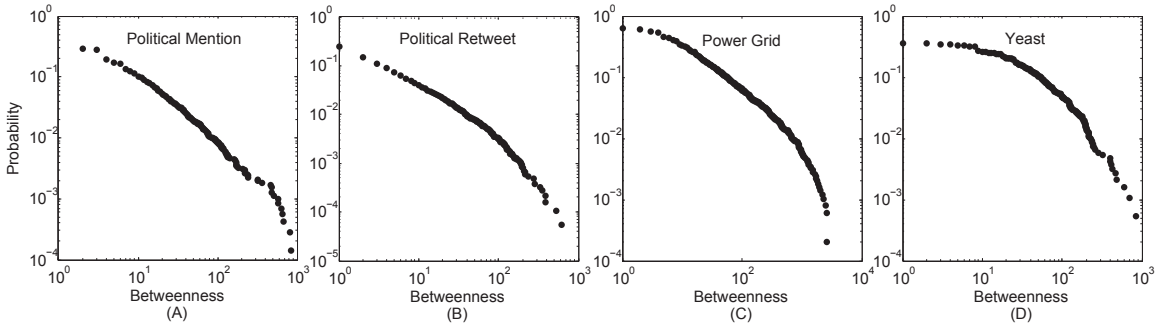



Figure 6.9: Betweenness distribution of the four networks. (A) Political Mention; (B) Political Retweet; (C) Power Grid; (D) Yeast PPI network.

We use the above high degree or betweenness strategies to set up sensors for the existing methods. We let the number of sensors account for no more than 50% of the total number of nodes in each network. For the proposed community structure based method, in order to select fewer sensors, we typically set $\alpha = 0.10$ for the Link clustering method in detecting overlapping community structures, and set $\beta = 2$ for the Infomap method in detecting separated community structures. The number of communities and bridges of these four networks under different experiment settings are shown in Table 6.3. We see that the number of communities is very small and the number of sensors account for less than 30% of the number of nodes in each network.

### Comparison Results

In the experiments, the diffusion probability is chosen uniformly from (0,1), and the diffusion process propagates $t$ time steps where $t$ is uniformly chosen from [8,10]. We use detection rate to measure the accuracy of identifying diffusion sources. The detection rate is defined as the fraction of experiments that accurately identify the real diffusion sources. The higher detection rate, the better performance.

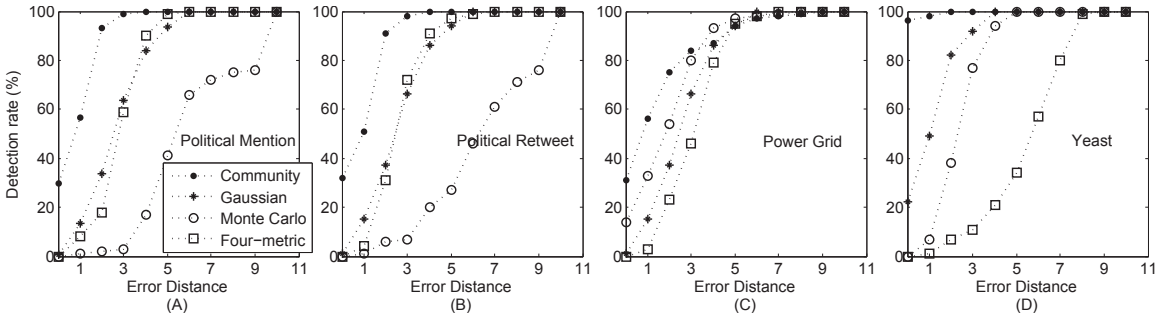



Figure 6.10: Comparison of the proposed method with other methods in the accuracy of identifying diffusion sources when setting sensors at high-degree nodes in four moderate-scale networks. (A) Political Mention; (B) Political Retweet; (C) Power Grid; (D) Yeast.

We first compare the proposed method to the existing methods associated with high degree sensors. More specifically, we utilized the Infomap method [89] to detect separated community structure of each network in this group of experiments. The experiment results are shown in Fig. 6.10. We can see that the detection rate of the proposed method is higher than that of the existing methods in each of the networks. For the two political networks (see Fig. 6.10 (A) and (B)), 30% of experiment runs can accurately identify the diffusion sources. More than 90% of experiment runs can identify a node within 2 hops away from the real source. Nearly 100% that the real source is within 3 hops around the estimated source. Furthermore, the average error distance from the real sources to the estimated sources is very small. However, for the existing methods, only few experiment runs can accurately identify the real diffusion sources. Similar results can be found in the Yeast PPI network (see Fig. 6.10 (D)). More than 90% of the experiment runs can accurately identify the real sources by using the proposed method, while few experiment runs can accurately identify the real sources by using the existing methods. The average error distance is much larger

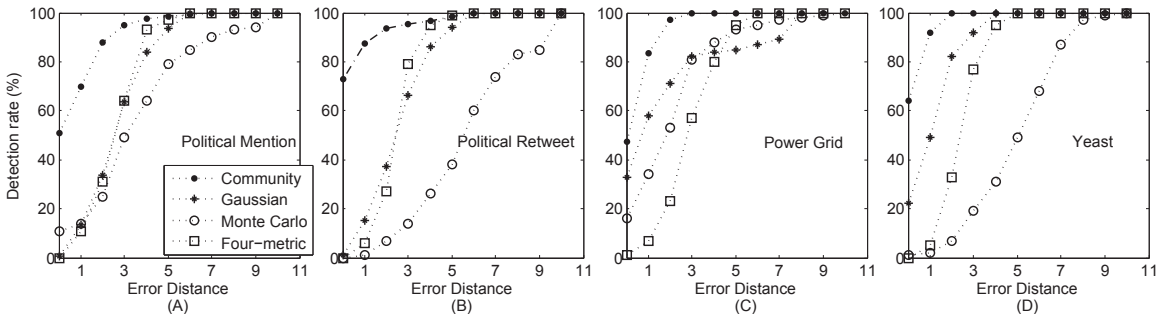



Figure 6.11: Comparison of the proposed method with other methods in the accuracy of identifying diffusion sources when setting sensors at high-betweenness nodes in four moderate-scale networks. (A) Political Mention; (B) Political Retweet; (C) Power Grid; (D) Yeast.

compared with that of the proposed method. The proposed method also outperforms the existing methods in the Power Grid network (see Fig. 6.10 (C)).

We then compare the proposed method to the existing methods associated with high-betweenness sensors. In this group of comparisons, we utilized the Link Clustering method [77] in detecting the overlapping community structure of each network. The experiment results on the four networks are shown in Fig. 6.11. Similar to the results in Fig. 6.10, the detection rate of the proposed method is higher than that of the existing methods in each of the four networks. For the two political networks, more than 50% of experiment runs accurately identified the real diffusion sources in the political Mention network, and more than 70% of experiment runs accurately identified the real diffusion sources in the political Retweet network. However, for the existing methods, few of the experiment runs accurately identified the diffusion sources. Furthermore, the average error distance is larger compared with that of the proposed method. Similar results can be found in the Power Grid network and the

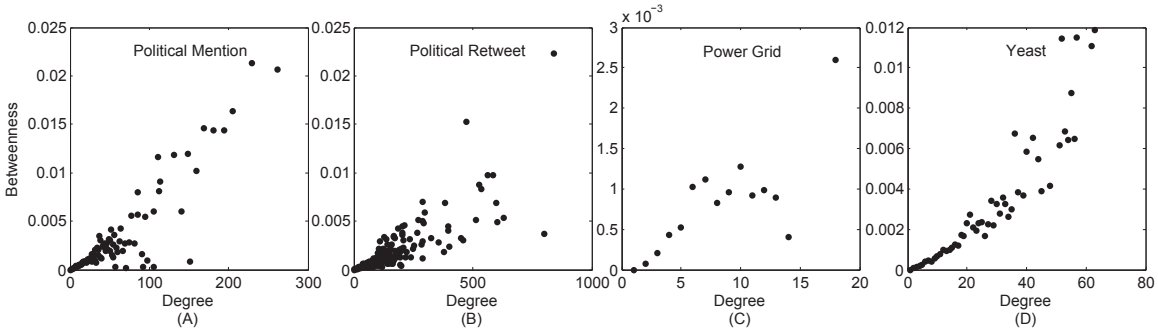



Figure 6.12: The relationship between degree and the average betweenness at each degree of the four networks. (A) Political Mention; (B) Political Retweet; (C) Power Grid; (D) Yeast PPI network.

Yeast PPI network. By using the proposed method, more than 45% of experiment runs accurately identified the diffusion sources in the Power Grid network, and more than 60% for the Yeast PPI network. However, for the existing methods, few of the experiment runs accurately identified the diffusion sources. Furthermore, the average error distance from the estimated sources and the real sources are larger compared that of the proposed method.

From Figs. 6.10 and 6.11, we see that the existing methods show different performances in the two different sensor selection methods. For example in Fig. 6.10 (D), the Monte Carlo method outperforms the Four-metric method with high degree sensor selection method. However, in Fig. 6.11 (D), the Four-metric method outperforms the Monte Carlo method with the high betweenness sensor selection method, and the Gaussian method shows similar performances. In order to see the impact of using different sensor selection methods on the existing methods, we show in Fig. 6.12 the correlation between nodes' degree and their average betweenness of the four

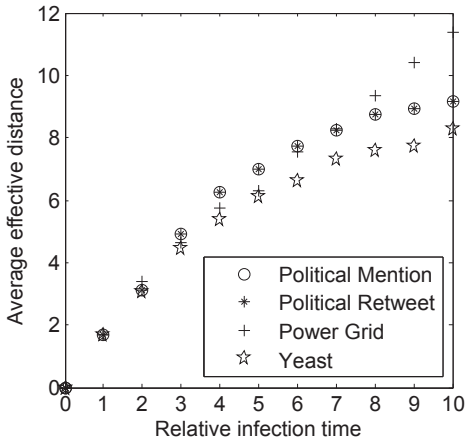



Figure 6.13: Linear correlation between relative infection time and average effective distance for the four relative small networks.

networks. As we can see, many nodes with the high degree tend to have high betweenness. However, this is not always the case, as there are also some nodes with low betweenness, especially for the political Retweet network and the Power Grid network. This explains why the existing methods show different performances in different sensor selection methods in Figs. 6.10 and 6.11.

Fig. 6.13 shows the linear relation between relative infection time of nodes and their average effective distance from the diffusion sources in the four relatively small networks. We can see that relative infection time is linear with the average effective distance in all these networks. Similar to the results in Figs. 6.6 (A) and 6.7 (A), the scatter plot curves towards the end because almost all of the nodes have been infected by then. The linear correlation in these networks again justifies the effectiveness of the proposed method.

In summary, we see that the proposed community structure based method outperforms the existing methods in identifying diffusion sources based on sensor observations in various networks. The majority of the experiment runs can accurately identify the real diffusion source or a node that is close to the real source. However, the existing methods show low performance, and the average error distance between the estimated sources and the real diffusion sources is very large.

## 6.5 Conclusion and Discussion

In this chapter, we proposed an efficient method to identify diffusion sources based on community structures in large-scale networks. To address the scalability issue in the source identification problems, we first detect community structure of the network and find bridges, which we assign as sensors. According to the infection time of the sensors, we can easily determine from which community the diffusion broke out. This method dramatically narrows down the scale of the search for diffusion sources and therefore address the scalability issue in this area. Then we proposed a novel method to locate the real diffusion source from the first infected community, and considered the one with the highest correlation coefficient as the real source. This method allows us to consider only sources inside the suspicious communities, rather than the whole network, which means the method can be applied just as efficiently to large networks as small ones. Experiments on large networks and comparison with many competitive methods show significant advantages of the proposed method.

From the present work, we see that a larger community size decreases the number of sensors, resulting in a less cost for implementation and maintenance, but also increases the number of nodes per community, and hence the number of suspects we

find, meaning results might not be accurate, and vice versa.

For future work, we may find a better way of assigning sensors to balance cost and accuracy through altering size or other aspects of communities. Furthermore, most existing methods for detecting source only detect a single source, however it may be the case that there are multiple sources in a single diffusion event. Finding multiple sources by adapting current methods is an interesting topic that can be investigated further. In addition, real-world networks are often associated with time-varying structure, such as collaboration networks. Future work could also take large networks with time-varying structure into account.

**Algorithm 6:** Community structure based approach

**Assigning Sensors**: Detect community structure of a given network, and then set the community bridges as sensors.

**Input**: A set of infected sensors $O = \{o_1, o_2, ..., o_k\}$, and their infection times $T = \{t_1, t_2, ..., t_k\}$.

**Initialization**: The optimal diffusion source $s^* = \emptyset$, and the optimal correlation coefficient $e^* = -\infty$.

**Step 1**: Find the earliest infected sensor. Without loss of generality, we assume $o_1$ is the first infected sensor. Choose $o_1$ as the reference, and calculate the relative infection time of all the infected sensors, denoted as

$$\tau = \{0, t_2 - t_1, t_3 - t_1, ..., t_k - t_1\}.$$

Find the communities that contain sensor $o_1$, and combine the nodes in these communities, denoted as

$$U = \{u_1, u_2, ..., u_m\}.$$

**Step 2**: Calculate the correlation coefficient for each node in $U$ and find the one which has the largest correlation coefficient as follows.

**for** *(each $u_i$ in $U$)* **do**

Compute the effective distance between $u_i$ and any infected sensor $o_j$, denoted as

$$\gamma = [D(u_i, o_1), D(u_i, o_2), ..., D(u_i, o_k)]. \tag{6.5.1}$$

Compute the correlation coefficient between $\tau$ and $\gamma$,

$$e = \frac{\sum_{j=1}^{k}(\tau_j - \bar{\tau})(\gamma - \bar{\gamma})}{\sqrt{\sum_{j=1}^{k}(\tau_j - \bar{\tau})^2}\sqrt{\sum_{j=1}^{k}(\gamma - \bar{\gamma})}}, \tag{6.5.2}$$

where $\bar{\tau}$ is the mean of $\tau$, and $\bar{\gamma}$ is the mean of $\gamma$.

**if** *($e > e^*$)* **then**

Set $e^* = e$, and $s^* = u_i$.

**Output**: The estimated optimal diffusion source $s^*$.

# Chapter 7

# Summary and Future Work

The research presented in this thesis consists of three main parts on the topic of *rumor source identification in complex networks*: the first part focuses on identifying rumor source in time-varying networks; the second part focuses on identifying multiple rumor sources; and the last part concentrates on efficiently identifying rumor sources in large-scale networks. The proposed methods aim to effectively detect rumor sources and eventually quarantine the wide spread of rumors. This chapter summarizes the research results and the main contributions of this thesis. Several open issues in rumor source identification and future research directions have also been identified.

## 7.1 Summary of Contributions

Theoretical and experimental results have led to the conclusions and main contributions of this thesis. They are:

- We developed an effective method to identify rumor sources in *time-varying*

*networks*. Traditional methods of rumor source identification assume firm connections between individuals, which leads to the overfitting problem of the underlying network. In this thesis, we reduce the time-varying networks into a series of static network windows. Based on the reduced network windows, we proposed a reverse dissemination strategy to narrow down the suspicious rumor sources. We also adopted the maximum likelihood estimation to pinpoint the true rumor sources from the suspects with a high accuracy. Experiment results justify the effectiveness of the proposed method in real-world time-varying networks.

- We proposed a fast method to identify *multiple rumor sources*. Few of current techniques can detect multiple rumor sources in complex networks and they all suffer from expensive computational complexity. In this thesis, we analyzed the diffusion patterns of multi-rumor spreading. Through combining K-means from data mining and effective distance from epidemic propagation, we formulated an optimization problem for multi-source identification and develop a fast method to solve the optimization problem. Theoretical analysis proves the efficiency of the proposed method, and the experiment results demonstrate the effectiveness of the proposed method in real-world networks.

- We addressed the *scalability issue* of identifying rumor sources in *large-scale networks*. Traditional methods are too computationally expensive to be able to quickly and accurately identify rumor sources in large-scale networks. In this thesis, we explored the intrinsic phenomenon that rumor diffusion is a network-driven process. In particular, we focused on community structure of networks.

Based on the community structure of networks, we successfully dramatically decreased the work of identifying rumor sources in the entire network to scanning a small community of the network. Theoretical analysis proves the efficiency of the proposed method, and the experiment results verify the significant advantages of the proposed method in large-scale networks.

## 7.2 Future Work

Although the proposed methods in this thesis have addressed three critical issues in identifying rumor sources in complex networks, there are still problems that need to be addressed. Some open issues are listed in this section as extensions of the presented work in the thesis. All the proposed future work is in the domain of rumor source identification in complex network. The following issues are ordered by the operational difficulties, from the easiest to the hardest.

### 7.2.1 Continuous Time-varying Networks

In Chapter 4, we developed a novel method to identify rumor sources in time-varying networks by utilizing discrete time-integrating windows to express time-varying networks. The size of the time window could be minutes, hours, days or even months. This may lead to new ideas of identifying rumor sources in continuous time windows.

In the real world, many complex networks − human contact network, online social networks, transportation network, computer networks, to just name a few − present continuous time-varying topologies. For example, in online social network websites, users continuous publish posts and commenting on posts, which is an essential part of

many social networking websites and forums. In many cases the data are recorded on a continuous time scale. The approach proposed in this thesis analyses discrete time windows, by dividing the entire time duration into several even intervals. This does greatly simplify time-varying networks but also lose some latent features of continuous time windows. The designing of detecting rumor sources in continuous time windows is a new direction for future research.

### 7.2.2 Multiple Rumors on the Same Topic

In Chapter 5, we proposed an efficient method to identify multiple rumors in complex networks. We considered multiple sources spreading the same rumor. In the real world, however, there often exist several different rumors on the same event spreading simultaneously in networks. These rumors may enhance the mass spreading of the same event. Therefore, identifying multiple sources of multiple rumors is of great significance.

Current research on rumor source identification only considers one rumor diffusion. However, real-world events generally are more complicated. For example, some rumor starting from March 2008 saying Obama was born in Kenya before bing flown to Hawaii were spread on social network websites. Some other rumor circulated on social network websites about his religion. These would disqualify Obama from the presidency. The rumors about the same event sometimes support each other, thus enlarge and extract more and more attentions from the general public, and finally mislead people. Therefore, how to identify sources of multiple rumors in complex networks is a good topic for future research.

### 7.2.3 Interconnected Networks

Current research on rumor source identification only considers rumor spreading in a single network. However, real-world networks are often interconnected or even interdependent. For example, in online social networks, a user could have a Facebook account and also have a Twitter account. After the user received a rumor on Facebook, he/she could also post the rumor on his/her Twitter account. Thus, the rumor will successfully spread from Facebook to Twitter. However, detecting rumor sources in interconnected networks is still an open issue. Therefore, identifying rumor sources in interconnected networks is much more realistic than methods considered in a single network.

# Bibliography


[1] A. Agaskar and Y. M. Lu. A fast monte carlo algorithm for source localization on graphs. In *SPIE Optical Engineering and Applications*. International Society for Optics and Photonics, 2013.

[2] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[3] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the north american power grid. *Physical review E*, 69(2):025103, 2004.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[5] F. Altarelli, A. Braunstein, L. DallAsta, A. Lage-Castellanos, and R. Zecchina. Bayesian inference of epidemics on networks via belief propagation. *Physical review letters*, 112(11):118701, 2014.

[6] R. M. Anderson, R. M. May, and B. Anderson. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

[7] N. T. Bailey et al. *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[8] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: Quantifying influence on twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 65–74, 2011.

[9] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[11] P. Bonacich. Power and centrality: A family of measures. *American journal of sociology*, pages 1170–1182, 1987.

[12] D. Brockmann and D. Helbing. The hidden geometry of complex, network-driven contagion phenomena. *Science*, 342(6164):1337–1342, 2013.

[13] C. Cattuto, W. Van den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PloS one*, 5(7):e11596, 2010.

[14] Z. Chen, K. Zhu, and L. Ying. Detecting multiple information sources in networks under the sir model. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–4. IEEE, 2014.

[15] C. H. Comin and L. da Fontoura Costa. Identifying the starting point of a spreading process in complex networks. *Phys. Rev. E*, 84:056105, Nov 2011.

[16] M. Conover, J. Ratkiewicz, M. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. Political polarization on twitter. In *ICWSM*, 2011.

[17] K. L. Cooke and P. Van Den Driessche. Analysis of an seirs epidemic model with two delays. *Journal of Mathematical Biology*, 35(2):240–260, 1996.

[18] D. Dagon, C. C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *NDSS*, volume 6, pages 2–13, 2006.

[19] D. J. Daley and D. G. Kendall. Epidemics and rumours. 1964.

[20] A. H. Dekker. Centrality in social networks: Theoretical and simulation approaches. *Proceedings of SimTecT 2008*, pages 12–15, 2008.

[21] B. Doerr, M. Fouz, and T. Friedrich. Why rumors spread so quickly in social networks. *Commun. ACM*, 55(6):70–75, June 2012.

[22] W. Dong, W. Zhang, and C. W. Tan. Rooting out the rumor culprit from suspects. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2671–2675. IEEE, 2013.

[23] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical review letters*, 85(21):4633, 2000.

[24] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[25] V. Fioriti, M. Chinnici, and J. Palomo. Predicting the sources of an outbreak with a spectral technique. *Applied Mathematical Sciences*, 8(135):6775–6782, 2014.

[26] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[27] S. Fortunato, A. Flammini, and F. Menczer. Scale-free network growth by ranking. *Physical review letters*, 96(21):218701, 2006.

[28] M. Fossi and J. Blackbird. Symantec internet security threat report 2010. Technical report, Symantec Corporation, March, 2011.

[29] C. Fraser, C. A. Donnelly, S. Cauchemez, W. P. Hanage, M. D. Van Kerkhove, T. D. Hollingsworth, J. Griffin, R. F. Baggaley, H. E. Jenkins, E. J. Lyons, et al. Pandemic potential of a strain of influenza a (h1n1): early findings. *science*, 324(5934):1557–1561, 2009.

[30] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[31] L. C. Freeman. A measure of betweenness centrality based on random walks. *Social networks*, 79:215–239, 1978.

[32] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[33] W. Goffman and V. Newill. Generalization of epidemic theory. *Nature*, 204(4955):225–228, 1964.

[34] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.

[35] P. Hage and F. Harary. Eccentricity and centrality in networks. *Social networks*, 17(1):57–63, 1995.

[36] S. L. Hakimi, M. L. Labbé, and E. Schmeichel. The voronoi partition of a network and its implications in location theory. *ORSA journal on computing*, 4(4):412–417, 1992.

[37] H. W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[38] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Physical Review E*, 65(5):056109, 2002.

[39] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, May 2001.

[40] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys and Tutorials*, accepted, in press.

[41] S. Jitesh and A. Jafar. The enron email dataset database schema and brief statistical report. Technical report, University of Southern California, 2009.

[42] N. Karamchandani and M. Franceschetti. Rumor source detection under probabilistic sampling. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2184–2188, 2013.

[43] B. Karrer and M. E. J. Newman. Message passing approach for general epidemic models. *Phys. Rev. E*, 82:016101, Jul 2010.

[44] M. Karsai, N. Perra, and A. Vespignani. Time varying networks and the weakness of strong ties. *Scientific reports*, 4, 2014.

[45] M. J. Keeling and K. T. Eames. Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4):295–307, 2005.

[46] M. J. Keeling and P. Rohani. *Modeling infectious diseases in humans and animals*. Princeton University Press, 2008.

[47] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: measurements, models, and methods. In *International Computing and Combinatorics Conference*, pages 1–17. Springer, 1999.

[48] P. L. Krapivsky and S. Redner. Organization of growing random networks. *Physical Review E*, 63(6):066123, 2001.

[49] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 57–65. IEEE, 2000.

[50] I. Lawrence and K. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989.

[51] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng. Optimal distributed malware defense in mobile networks with heterogeneous devices. *Mobile Computing, IEEE Transactions on*, 2013. Accepted.

[52] Y. Y. Liu, J. J. Slotine, and A. laszlo Barabasi. Controllability of complex networks. *Nature*, 473:167–173, 2011.

[53] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová. Inferring the origin of an epidemy with dynamic message-passing algorithm. *arXiv preprint arXiv:1303.5315*, 2013.

[54] A. Louni and K. Subbalakshmi. A two-stage algorithm to estimate the source of information diffusion in social media networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 329–333. IEEE, 2014.

[55] W. Luo and W. P. Tay. Identifying infection sources in large tree networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, pages 281–289. IEEE, 2012.

[56] W. Luo and W. P. Tay. Identifying multiple infection sources in a network. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 1483–1489. IEEE, 2012.

[57] W. Luo and W. P. Tay. Finding an infection source under the sis model. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 2930 – 2934, 2013.

[58] W. Luo, W. P. Tay, and M. Leng. Identifying infection sources and regions in large networks. *Signal Processing, IEEE Transactions on*, 61(11):2850–2865, 2013.

[59] W. Luo, W. P. Tay, and M. Leng. How to identify an infection source with limited observations. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):586–597, 2014.

[60] W. Luo, W. P. Tay, and M. Leng. Rumor spreading and source identification: A hide and seek game. *arXiv preprint arXiv:1504.04796*, 2015.

[61] Y. Ma, X. Jiang, M. Li, X. Shen, Q. Guo, Y. Lei, and Z. Zheng. Identify the diversity of mesoscopic structures in networks: A mixed random walk approach. *EPL (Europhysics Letters)*, 104(1):18006, 2013.

[62] D. MacRae. 5 viruses to be on the alert for in 2014.

[63] A. R. McLean, R. M. May, J. Pattison, R. A. Weiss, et al. *SARS: A case study in emerging infections.* Oxford University Press, 2005.

[64] S. Meloni, A. Arenas, S. Gómez, J. Borge-Holthoefer, and Y. Moreno. Modeling epidemic spreading in complex networks: concurrency and traffic. In *Handbook of Optimization in Complex Networks*, pages 435–462. Springer, 2012.

[65] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[66] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security and Privacy*, 1(4):33–39, July 2003.

[67] Y. Moreno, M. Nekovee, and A. F. Pacheco. Dynamics of rumor spreading in complex networks. *Physical Review E*, 69(6):066130, 2004.

[68] M. Newman. A measure of betweenness centrality based on random walks. *Social networks*, 27:39–54, 2005.

[69] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[70] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[71] M. E. Newman. The mathematics of networks. *The new palgrave encyclopedia of economics*, 2:1–12, 2008.

[72] M. E. Newman and J. Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122, 2003.

[73] M. E. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

[74] M. E. J. Newman. *Networks: An Introduction*, chapter 17 Epidemics on networks, pages 700–750. Oxford University Press, 2010.

[75] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 85–96. ACM, 2011.

[76] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[77] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[78] C. Pash. The lure of naked hollywood star photos sent the internet into meltdown in new zealand. *Business Insider Australia*, September 7 2014, 4:21 PM.

[79] F. Peter. 'bogus' ap tweet about explosion at the white house wipes billions off us markets, April 23 2013. Washington.

[80] S. Pettie and V. Ramachandran. A shortest path algorithm for real-weighted undirected graphs. *SIAM Journal on Computing*, 34(6):1398–1431, 2005.

[81] A.-K. Pietilainen. CRAWDAD data set thlab/sigcomm2009 (v. 2012-07-15). Downloaded from http://crawdad.org/thlab/sigcomm2009/, July 2012.

[82] P. C. Pinto, P. Thiran, and M. Vetterli. Locating the source of diffusion in large-scale networks. *Phys. Rev. Lett.*, 109, Aug 2012.

[83] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 11–20, Washington, DC, USA, 2012. IEEE Computer Society.

[84] B. A. Prakash, J. Vreeken, and C. Faloutsos. Efficiently spotting the starting points of an epidemic in a large graph. *Knowledge and Information Systems*, 38(1):35–59, 2014.

[85] A. Rapoport. Spread of information through a population with socio-structural bias: I. assumption of transitivity. *The bulletin of mathematical biophysics*, 15(4):523–533, 1953.

[86] J. G. Restrepo, E. Ott, and B. R. Hunt. Characterizing the dynamical importance of network nodes and links. *Phys. Rev. Lett.*, 97:094102, Sep 2006.

[87] B. Ribeiro, N. Perra, and A. Baronchelli. Quantifying the effect of temporal resolution on time-varying networks. *Scientific reports*, 3, 2013.

[88] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007.

[89] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[90] M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39):15224–15229, 2007.

[91] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for ip traceback. *ACM SIGCOMM Computer Communication Review*, 30(4):295–306, 2000.

[92] C. Scoglio, W. Schumm, P. Schumm, T. Easton, S. R. Chowdhury, A. Sydney, and M. Youssef. Efficient mitigation strategies for epidemics in rural regions. *PloS one*, 5(7):e11569, 2010.

[93] V. Sekar, Y. Xie, D. A. Maltz, M. K. Reiter, and H. Zhang. Toward a framework for internet forensic analysis. In *ACM HotNets-III*, 2004.

[94] E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*, volume 8389, 2012.

[95] D. Shah and T. Zaman. Detecting sources of computer viruses in networks: Theory and experiment. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '10, pages 203–214. ACM, 2010.

[96] D. Shah and T. Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57:5163 – 5181, 2011.

[97] D. Shah and T. Zaman. Rumor centrality: A universal source detector. *SIGMETRICS Perform. Eval. Rev.*, 40(1):199–210, June 2012.

[98] J. Shetty and J. Adibi. The enron email dataset database schema and brief statistical report. *Information Sciences Institute Technical Report, University of Southern California*, 4, 2004.

[99] L.-P. Song, Z. Jin, and G.-Q. Sun. Modeling and analyzing of botnet interactions. *Physica A: Statistical Mechanics and its Applications*, 390(2):347–358, 2011.

[100] M. Spiliopoulou. Evolution in social networks: A survey. In C. C. Aggarwal, editor, *Social Network Data Analytics, Chapter 6*, pages 149–175. Springer US, 2011.

[101] K. Starbird, J. Maddock, M. Orand, P. Achterman, and R. M. Mason. Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. *iConference 2014 Proceedings*, 2014.

[102] J. H. O. Sỳkora. Graph-theoretic concepts in computer science. 1998.

[103] W. E. R. Team. Ebola virus disease in west africathe first 9 months of the epidemic and forward projections. *N Engl J Med*, 371(16):1481–95, 2014.

[104] M. P. Viana, D. R. Amancio, and L. d. F. Costa. On time-varying collaboration networks. *Journal of Informetrics*, 7(2):371–378, 2013.

[105] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, WOSN '09, pages 37–42, 2009.

[106] Y. Wang, S. Wen, Y. Xiang, and W. Zhou. Modeling the propagation of worms in networks: A survey. *Communications Surveys Tutorials, IEEE*, PP(99):1–19, 2013.

[107] Z. Wang, W. Dong, W. Zhang, and C. W. Tan. Rumor source detection with multiple observations: Fundamental limits and algorithms. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '14, pages 1–13. ACM, 2014.

[108] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[109] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, WORM '03, pages 11–18, 2003.

[110] S. Wen, W. Zhou, J. Zhang, Y. Xiang, W. Zhou, and W. Jia. Modeling propagation dynamics of social network worms. *Parallel and Distributed Systems, IEEE Transactions on*, 24(8):1633–1643, 2013.

[111] L. Weng, F. Menczer, and Y.-Y. Ahn. Virality prediction and community structure in social networks. *Scientific reports*, 3, 2013.

[112] P. Wood and G. Egan. Symantec internet security threat report 2011. Technical report, Symantec Corporation, April, 2012.

[113] Y. Xiang, X. Fan, and W. T. Zhu. Propagation of active worms: a survey. *International journal of computer systems science & engineering*, 24(3):157–172, 2009.

[114] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang. Worm origin identification using random moonwalks. In *Security and Privacy, 2005 IEEE Symposium on*, pages 242–256. IEEE, 2005.

[115] K. Yang, A. H. Shekhar, D. Oliver, and S. Shekhar. Capacity-constrained network-voronoi diagram: a summary of results. In *International Symposium on Spatial and Temporal Databases*, pages 56–73. Springer, 2013.

[116] Y. Yao, X. Luo, F. Gao, and S. Ai. Research of a potential worm propagation model based on pure p2p principle. In *Communication Technology, 2006. ICCT'06. International Conference on*, pages 1–4. IEEE, 2006.

[117] D. H. Zanette. Dynamics of rumor propagation on small-world networks. *Physical review E*, 65(4):041908, 2002.

[118] W. Zang, P. Zhang, C. Zhou, and L. Guo. Discovering multiple diffusion source nodes in social networks. *Procedia Computer Science*, 29:443–452, 2014.

[119] G.-M. Zhu, H. Yang, R. Yang, J. Ren, B. Li, and Y.-C. Lai. Uncovering evolutionary ages of nodes in complex networks. *The European Physical Journal B*, 85(3):1–6, 2012.

[120] K. Zhu and L. Ying. Information source detection in the sir model: A sample path based approach. In *Information Theory and Applications Workshop (ITA)*, pages 1–9, 2013.

[121] K. Zhu and L. Ying. A robust information source estimator with sparse observations. *Computational Social Networks*, 1(1):1, 2014.

[122] K. Zhu and L. Ying. Information source detection in the sir model: a sample-path-based approach. *IEEE/ACM Transactions on Networking*, 24(1):408–421, 2016.

[123] Y. Zhu, B. Xu, X. Shi, and Y. Wang. A survey of social-based routing in delay tolerant networks: Positive and negative social effects. *Communications Surveys Tutorials, IEEE*, 15(1):387–401, Jan 2013.

[124] Z. Zhu, G. Lu, Y. Chen, Z. Fu, P. Roberts, and K. Han. Botnet research survey. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 967–972, July 2008.

[125] C. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 138–147, 2002.

[126] C. C. Zou, D. Towsley, and W. Gong. Modeling and simulation study of the propagation and defense of internet e-mail worms. *IEEE Transactions on Dependable and Secure Computing*, 4(2):105–118, 2007.

[127] C. C. Zou, D. Towsley, and W. Gong. Modeling and simulation study of the propagation and defense of internet e-mail worms. *IEEE Transactions on dependable and secure computing*, 4(2):105–118, 2007.