复杂网络多源节点检测研究

作者姓名 李晓杰

指导教师姓名、职称 吴建设 教授

申请学位类别 工学硕士

学校代码 10701　　学　号 1502120798
分 类 号 TP75　　密　级 公开

西安电子科技大学

硕士学位论文

复杂网络多源节点检测研究

作者姓名：李晓杰

一级学科：电子科学与技术

二级学科：电路与系统

学位类别：工学硕士

指导教师姓名、职称：吴建设　教授

学　　院：人工智能学院

提交日期：2018 年 4 月

Research on Multi-source Identification in Complex Networks

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Circuits and Systems

By

Li Xiaojie

Supervisor: Wu Jianshe Title: Professor

April 2018

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名：_____ 日　　期：_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在___年解密后适用本授权书。

本人签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘要

基于有限的网络结构知识与网络节点状态信息实现传播源节点的检测一直以来都是一个意义重大却又难以解决的问题。过去几年，研究人员提出的方法大多针对的是树形网络上的单源节点检测，然而现实中的网络一般比树形网络复杂的多，而且由于传播所在时空的复杂性以及传播过程的不确定性，实际传播中往往会同时存在多个传播源，但是现有的方法却很少有针对多源节点检测的。针对以上问题，论文对一般网络上的多源节点检测问题展开研究，全文的主要工作如下：

1. 提出一种基于 SI 传播模型的多源节点检测算法。首先从传播时间的角度出发，将 SI 模型下的多源节点检测问题转化为寻找网络中可以最小化分区传播时间之和的 k 个节点问题，并抽象出该问题的目标函数，然后提出 KST 算法以迭代的方式最小化该目标函数，从而实现多源节点的检测。实验结果表明 KST 算法可以得到相对较高的检测准确度。同时，还提出用有效传播时间来估计网络中任意两点间的传播时间，进一步优化了 KST 算法的检测准确度。最后，提出了一种可以估计传播源个数的启发式算法用以解决实际中传播源个数一般难以提前获知的问题。

2. 在前面提出的 KST 算法基础之上提出了 SIR 传播模型下的多源节点检测算法，WP-KST 算法。首先针对 SIR 模型下不能正确区分恢复节点和易感染节点的问题，提出了一种权值传播算法，实现恢复节点的检测。仿真实验证明权值传播算法可以很好的检测出网络中的恢复节点，完成缺失信息的填充。接着在得到由感染节点与恢复节点以及这些节点间的连边所组成的扩展感染网络上，运用提出的 KST 算法进行多源节点的检测。实验结果表明 WP-KST 算法可以很好的解决 SIR 模型下的多源节点检测问题，且具有较高的检测准确度。

3. 研究了传感器观察方式下的源节点检测问题。假设传播遵循 SI 模型，首先提出基于传感器观察的单源节点检测算法，RDPC 算法。RDPC 算法首先利用反向传播算法筛选出网络中的可能源节点，然后针对每个可能源节点，检测其到所有感染传感器的传播时间与感染传感器记录的相对感染时间之间的线性相关性，选择具有最大线性相关性的节点作为传播源节点。通过实验验证了 RDPC 算法具有较高的检测准确度。此外，通过一个简单的划分思路，将 RDPC 算法扩展到了多源节点检测问题上，实验结果表明扩展后的 RDPC 算法可以很好的解决传感器观察方式下的多源节点检测问题。

关键字：复杂网络， 信息传播， 传播模型， 多源检测

ABSTRACT

It has long been a significant but difficult problem to identify propagation sources based on limited knowledge of network structures and the varying states of network nodes. In the past few years, researchers have proposed a series of methods to identify diffusion sources in networks. These methods mainly focus on the identification of a single diffusion source in tree networks, however the topologies of real world networks are far more complex than trees. Moreover, due to the spatiotemporally complex and the uncertainty of the propagation process, there often exit multiple diffusion sources in the actual propagation. However, only a few of existing methods are proposed for identifying multiple diffusion sources. In order to solve the above problems, the paper studies the problem of multiple source identification on general networks, the main work of the full text is as follows:

1．A multi-source identification algorithm based on SI model is proposed. Firstly, from the perspective of spreading time, the problem of multi-source identification under the SI model is transformed into finding k nodes in the network that can minimize the sum of all partition spreading times and formulate it as an objective function. Then the KST algorithm is proposed to minimize the objective function in an iterative manner to achieve identification of multiple sources. Experimental results show that KST algorithm can get relatively high identification accuracy. At the same time, we also propose the effective spreading time to estimate the propagation time between any two nodes in the network and the effective spreading time can further optimize the identification accuracy of the KST algorithm. Finally, a heuristic algorithm that can estimate the number of diffusion sources is proposed to solve the problem that the number of diffusion sources is difficult to know in advance.

2．Based on the KST algorithm, the multi-source identification algorithm under the SIR propagation model is proposed which is called WP-KST algorithm. Firstly, aiming at the problem that the recovery nodes cannot correctly distinguish from the susceptible nodes under the SIR model, a weight propagation algorithm is proposed to realize the detection of recovery nodes. The simulation experiment proves that the weight propagation algorithm can detect the recovery nodes in the network well and fill in the missing information. Then we use the KST algorithm to identify the multi-source on the extended

infection network which is consisted of infected nodes and recovery nodes and the edges between these nodes. The experimental results show that the WP-KST algorithm can solve the multi-source identification problem under SIR model well and has high detection accuracy.

3. The source identification problem under the sensor observation is studied. Assuming that the propagation follows the SI model, a single-source detection algorithm based sensor observation is first proposed which is called RDPC algorithm. The RDPC algorithm first uses the back-propagation algorithm to screen out possible source nodes in the network. Then for each possible source node, it detects the linear correlation between its spreading times to all infected sensors and the relative infection times recorded by the infected sensors. The node with the largest linear correlation is regard as the diffusion source. The experiment verifies that the RDPC algorithm has a higher detection accuracy. In addition, through a simple partitioning idea, the RDPC algorithm is extended to multi-source identification problems under the sensor observation. Experimental results show that the extended RDPC algorithm can solve the multi-source identification problem under the sensor observation well.

Key words: complex network, information diffusion, propagation model, multi-source identification

插图索引

表格索引

符号对照表

符号	符号名称
G	无权无向图
N	节点个数
E	网络边集合
V	网络点集合
A	网络邻接矩阵
a_{ij}	邻接矩阵中的元素
η_{ij}	邻居节点间传播概率
G_I	感染网络
$G_{I\cup R}$	扩展感染网络
Δ	平均误差距离
P_S	易感染状态概率
P_I	感染状态概率
P_R	恢复状态概率

缩略语对照表

缩略语	英文全称	中文对照
BFS	Breadth-first search	广度优先搜索
SI	Susceptible-infected	易感染-感染
SIS	Susceptible-infected-susceptible	易感染-感染-易感染
SIR	Susceptible-infected-recovery	易感染-感染-恢复
KST	K shortest spreading time	K 个最小传播时间节点
WP	Weight propagation	权值传播
DC	Divide and conquer	分治
CC	Closeness centrality	接近度中心性
RDPC	Reverse dissemination pearson correlation	反向传播与皮尔逊相关性

目录

第一章 绪论

1.1 复杂网络

自然界中存在着各种各样的复杂系统，如交通系统、电力系统以及生物系统等等，这些系统的复杂性源于其内部组成个体的多样性以及个体间交互行为的错综复杂性，要理解这些复杂系统，应该先从理解这些系统的结构与功能之间的关系开始。

复杂网络可以将复杂系统抽象为网络，所形成的网络拓扑结构信息是构建系统模型、研究系统性质和功能的基础。现实中很多复杂系统都可以通过构建网络来进行分析，系统中的个体用网络中的节点表示，个体间的联系用网络中的连边表示[1-4]。例如，社会这个复杂系统可以用一个大型社会网络来刻画，网络中的节点代表真实社会中的个体或组织，而连边则表示个体或组织间的交互行为。类似的，电力系统可以抽象为由变电站、发电厂、用户以及相互连接关系组成的复杂网络；代谢网络可以看作是由代谢酶个体以及它们之间发生的生化反应所组成的复杂网络等等。

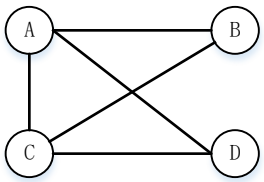
增加参考文献

我们生活的世界中存在着形形色色的网络，随着社会的发展与科技的进步，这些网络对我们生活的影响会越来越大。网络的发展在便利我们日常生活的同时也让我们面临越来越多的挑战，比如世界范围内传染病的传播，全球性的计算机病毒等等，所以有必要对复杂网络进行深入的研究，了解其特性，扬长避短，从而为我们的生活更好的服务。

1.1.1 复杂网络表示方式

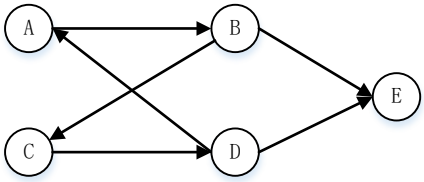
图论中通过用点表示元素、边表示元素间的作用关系来抽象各种实际网络，这种表示方式也成为复杂网络研究领域中的一种共同语言。

基于图论的知识，一个复杂网络通常由节点集和边集组成，记为$G(V,E)$，其中V表示网络的节点集合，E表示网络的连边集合，边集E中的每一条边连接着节点集V中的一对节点。网络通常用一个方阵来表示，称为邻接矩阵。矩阵元素a_{ij}表示节点i和节点j的连接关系，若节点i连接到了节点j上，则$a_{ij}=1$，否则$a_{ij}=0$。根据节点间的连边是否有方向以及是否有权重，网络可分为四类：无权无向图，无权有向图，加权无向图和加权有向图，图1.1给出了四种网络邻接矩阵的具体示例。在有向网络中箭头表示连边的方向，在有权网络中邻接矩阵元素a_{ij}表示连边上的权重，可以看出无权网络本质上是权重为1的有权网络。



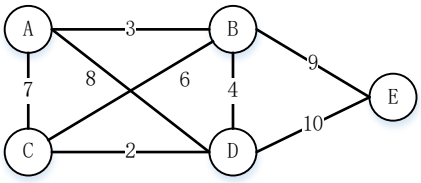
$$A_1=\begin{bmatrix}0&1&1&1\\1&0&1&0\\1&1&0&1\\1&0&1&0\end{bmatrix}$$

(A)无权无向网络



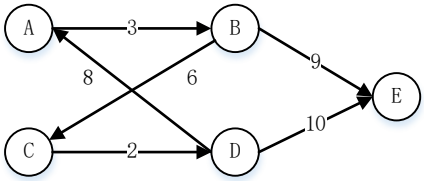
$$A_2=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&1\\0&0&0&1&0\\1&0&0&0&1\\0&0&0&0&0\end{bmatrix}$$

(B)无权有向网络



$$A_3=\begin{bmatrix}0&3&7&8&0\\3&0&6&4&9\\7&6&0&2&0\\8&4&2&0&10\\0&9&0&10&0\end{bmatrix}$$

(C)加权无向网络



$$A_4=\begin{bmatrix}0&3&0&0&0\\0&0&6&0&9\\0&0&0&2&0\\8&0&0&0&10\\0&0&0&0&0\end{bmatrix}$$

(D)加权有向网络

图1.1 四种网络邻接矩阵示意图

1.1.2 节点的度

无向网络中，节点的度定义为与该节点连接的其他节点的数目，即邻居节点的个数。

有向网络中，节点的度则分为出度和入度，出度定义为以该节点为起点的有向边个数，入度则为以该节点为终点的有向边个数。

1.1.3 最短路径与路阶数

在一个网络$G(V,E)$中，如果可以从某个节点 i 出发，沿着网络中的连边途经一些节点$a,b,...,c$到达另一个节点 j，则称网络中存在一条从节点 i 到节点 j 的路径$\{i,a,b,...,c,j\}$，该路径所经过的边集为$\{(i,a),(a,b),...,(c,j)\}$。

在无权网络中，连接两个节点的某一条路径上的边数称为该路径的长度。对于有权网络，路径的长度则为该路径中所有边的权重之和。对于网络中任意两个节点，它们之间可能存在不止一条路径，两个节点间所有路径中长度最小的路径被称为最短路

径。

路阶数常用来描述路径上的连边数，对于无权网络来说路阶数等价于路径长度，最短路径上路阶数最小，对于有权网络最短路径上路阶数不一定最小。

1.2 源点检测

近年来，关于复杂网络的研究主要有以下几个方面：节点相似性[5, 6]、社区检测[7, 8]、链路预测[9-11]、网络重构[12-14]以及网络动力学[15-18]等。其中网络动力学的研究可以让人们了解网络拓扑结构的变化对网络动力学过程的影响，从而优化和改善网络的动力学行为。作为动力学研究的一个重要方向，传播动力学主要研究社会和自然界中各种复杂网络的传播机理与动力学行为以及对这些行为高效可行的控制方法[19]。随着社会、交通和计算机网络等现代网络的急剧增加，传播现象在我们的生活中越来越常见。在过去的十几年里，现实生活出现了好多关于传播的问题，例如 2003 年发生在中国大陆的 SARS 病毒传播事件；2011 年谣言引发的全国范围内碘盐抢购事件；2017 年全球范围内爆发的“永恒之蓝”网络攻击事件等等。这些有害疾病或信息的传播对整个社会造成了严重的影响。如何通过对这类有害信息传播行为的研究，进一步准确直观地理解其传播机理，从而发现传播的薄弱环节，对其加以控制甚至消除就显得尤为重要。

为了控制甚至消除这类有害信息或疾病的传播，需要找到这类传播的源头[20]。通过准确检测出传播源头，可以进一步预测传播的发展趋势从而找出可以及时制止甚至消除传播的方法[21, 22]。例如，快速检测出一个传染病的首发病例可以加快疾病成因以及传播途径等的研究，从而促进高效治疗药物与治疗方法的研发。而且还可以通过推断潜在感染者，建立相应的隔离策略以阻止疾病的进一步传播[23, 24]。类似的，检测出计算机网络上第一个被感染的服务器或设施可以让我们发现网络上潜在的薄弱环节，从而采取一些预防措施来加强对这些环节的保护。

在一个已知网络结构的复杂网络上，信息或疾病在某个时刻从一个或多个源头爆发并开始向外进行传播，在传播持续一段时间后，通过观察网络找出哪个或哪些节点是传播源节点，这类问题称为源点检测问题。

1.2.1 源点检测研究现状

尽管传播源节点检测问题意义重大，但在 2009 年之前[25]，尚未吸引大量学者的研究，主要原因是基于有限的网络结构知识和网络节点状态信息实现传播源点的检测非常具有挑战性。

传统的检测技术，如 IP 回溯法[7]和踏脚石检测法[26]，都不足以检测出信息的传播源点，因为它们只能确定所收到信息的确切来源。而在信息的传播中，信息的来源一般都不会是传播的起源，而只是众多传播参与者之一[27]。在过去的几年里，研究人员提出了一系列的方法来检测网络中的传播源点。起初，检测方法所基于的传播模型为经典的 SI(susceptible-infected)模型[25, 28-33]，该模型中一个易感染节点会以一定概率被感染成为感染节点，且一旦被感染将永远处于感染状态。这些方法所适用的网络也基本是树形网络，并且 Shah[25]等人指出即使是在简单的树形网络下，单源节点的检测仍然是一个 NP 完全问题。

此外，研究人员还提出了一些基于其他传播模型的源点检测方法，如 Zhu[34]等人假设传播模型为 SIR(susceptible-infected-recovery)模型，模型中的感染节点会有一定的机会从感染中恢复过来且不会被再次感染。Luo[35]等人则研究了 SIS(susceptible-infected-susceptible)模型下的源点检测问题。所有的这些方法都只适用于树形网络，然而现实中传播所在的网络一般都不可能是树形网络。后来，研究人员通过一些启发式的方法放宽了网络拓扑的约束，提出了可适用于一般网络结构的检测算法[36-42]。另外，考虑到现实情况中要观察网络中所有感染节点比较困难，研究人员还提出了通过提前在网络中注入传感器以监控网络中的传播过程，从而利用传感器所提供的信息来进行传播源的检测[43-46]，例如 Pinto[44]等人假设每个感染节点的感染时间以及感染来自哪个邻居节点是已知的，提出了一种基于极大似然估计的算法。

上述提到的大多数算法都集中在单源节点检测问题上，然而现实中由于网络的复杂性，传播初期一般不会只有一个源，如感染病可以从多个地点开始，谣言传播中会有多个谣言散布者。目前存在的针对多源节点检测的算法很少，Luo 等人提出的 Multiple Rumor Center[35]多源检测算法只使用于树形网络，而且该方法的算法复杂度为 $O(n^k)$，n 是感染节点个数，k 是传播源个数，计算复杂度太高，不适用于大型网络。Fioriti 等人提出的 Dynamic age[47]来实现一般网络上的多源节点检测，他们认为与邻近矩阵特征值所对应的节点是最老的节点，也就是传播的源节点，但该方法的一个基本先决条件就是需要事先知道传播源的个数。

1.2.2 目前存在的主要问题

传播源节点检测问题意义重大，但从源点检测的研究现状来看，目前提出的这些方法在适用性方面还存在以下几点问题：

1. 网络拓扑结构：目前存在的大多数方法基于的网络结构都是树形网络结构，而实际中的大多数网络都不是树形网络。虽然可以通过使用 BFS(Breadth First Search)技术将一般网络重构为树形网络来进行源点检测，但由于 BFS 树中忽略了循环结构对传播的影响，其准确性得不到保证。因此，不能直接在一般网络上使用或扩

展基于树形网络的方法进行源点检测。

2. 源点个数：目前大多数方法都集中在单源点检测，很少有针对多源检测的方法。在现实世界中，信息通常是从多个初始点开始爆发的。从技术上讲，单源检测方法不能直接用于多源检测，这是因为从多个来源发起的传播不能简单地认为是多个单源传播过程的叠加。目前仅有的几个多源节点检测算法，如 Multiple Rumor Centers[32]和 Dynamic Age[47]，计算复杂度都比较高，无法在一般真实网络上实施，且检测准确率也有待提高。
3. 计算复杂度：在大多情况下，快速识别传播源在现实世界中具有重要意义。但是，目前大多数方法的计算复杂度都过高，有的复杂度高达 $O(N^k)$，过高的复杂度无法完成传播源的快速定位。而且，一般情况下真正传播都是发生在大规模网络上，复杂性变得更糟。

1.3 论文的主要工作和结构安排

1.3.1 主要工作

在过去的几年中，研究人员提出了一系列的方法来检测网络中的传播源。这些方法主要针对的是树形网络上的单源节点检测，而现实网络远远复杂于树形网络。而且由于传播所在时空的复杂性以及传播过程的不确定性，实际传播中往往同时会存在多个传播源，现有的方法却很少有针对多源节点检测的。针对以上问题，论文以一般网络上的多源节点检测问题展开研究，主要工作如下：

1. 提出了一种基于 SI 传播模型的多源节点检测算法。首先从传播时间的角度出发，将 SI 模型下的多源节点检测问题转化为寻找网络中可以最小化分区传播时间之和的 k 个节点问题，并抽象出该问题的目标函数，然后提出 KST(K Shortest spreading Time)算法以迭代的方式最小化该目标函数，从而实现多源节点的检测。实验结果表明 KST 算法可以得到相对较高的检测准确度。同时，还提出用有效传播时间来估计网络中任意两点间的传播时间，进一步优化了 KST 算法的检测准确度。最后，提出了一种可以估计传播源个数的启发式算法用以解决实际中传播源个数一般难以提前获知的问题。

2. 在前面提出的 KST 算法基础之上提出了 SIR 传播模型下的多源节点检测算法，WP-KST(Weight Propagation-KST)算法。首先针对 SIR 模型下不能正确区分恢复节点和易感染节点的问题，提出了一种权值传播算法，实现恢复节点的检测。仿真实验证明权值传播算法可以很好的检测出网络中的恢复节点，完成缺失信息的填充。接着在得到由感染节点与恢复节点以及这些节点间的连边所组成的扩展感染网络上，运用提

出的KST算法进行多源节点的检测。实验结果表明WP-KST算法可以很好的解决SIR模型下的多源节点检测问题，且具有较高的检测准确度。

3. 研究了传感器观察方式下的源节点检测问题。假设传播遵循SI模型，首先提出基于传感器观察的单源节点检测算法，RDPC(Reverse Dissemination Pearson Correlation)算法。RDPC算法首先利用反向传播算法筛选出网络中的可能源节点，然后针对每个可能源节点，检测其到所有感染传感器的传播时间与感染传感器记录的相对感染时间之间的线性相关性，选择具有最大线性相关性的节点作为传播源节点。通过实验验证了RDPC算法具有较高的检测准确度。此外，通过一个简单的划分思路，将RDPC算法扩展到了多源节点检测问题上，实验结果表明扩展后的RDPC算法可以很好的解决传感器观察方式下的多源节点检测问题。

1.3.2 结构安排

论文共分为六章，每一章的主要内容如下：

第一章是绪论。首先介绍了复杂网络的研究背景以及基本概念。然后提出了源点检测的概念，并详细介绍了源点检测问题的研究现状以及目前研究中存在的不足，最后介绍了本文的主要工作以及论文的结构安排。

第二章是基础理论，首先介绍了源点检测领域中常用的基础知识，包括传播模型、观察方式以及中心性指标。之后针对不同观察方式详细介绍了几种源点检测方法，并简单分析了各算法优缺点。

第三章提出基于SI传播模型的多源节点检测算法。首先从传播时间的角度出发，介绍了KST算法的理论依据以及算法详细过程，通过实验验证了KST算法的检测准确度。同时，还提出有效传播时间来进一步优化了KST算法的检测准确度。最后，针对实际中传播源个数一般难以提前获知的问题，给出的一种可以估计传播源个数的启发式算法。

第四章提出SIR模型下的多源节点检测算法，WP-KST算法。首先针对SIR模型下不能正确区分恢复节点和易感染节点这一问题，提出权值传播算法，通过实验验证了权值传播算法可以很好的检测出恢复节点。然后通过在扩展感染网络上运用KST算法实现多源节点的检测，实验结果验证了WP-KST算法可以很好的解决SIR模型下的多源节点检测问题，且具有较高的检测准确度。

第五章研究了传感器观察方式下的源节点检测问题。首先给出了传感器观察下的源点检测问题描述，然后假设传播遵循SI模型，提出了传感器观察下的单源节点检测算法，RDPC算法。通过实验验证了RDPC算法具有较高的检测准确度。最后，通过一种划分思想将RDPC算法扩展到多源节点检测问题上，实验验证了扩展后的RDPC算法可以很好的解决传感器观察方式下的多源节点检测问题。

第六章是总结与展望，对本文所做的工作进行了总结，在此基础展望了下一步研究方向。

第二章 源点检测相关知识简介

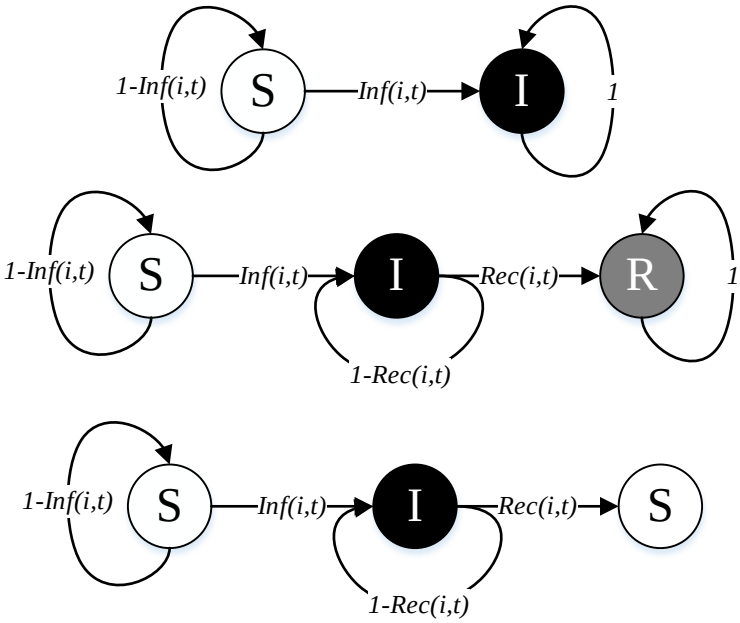
本章简要介绍源点检测的相关基础知识，首先介绍了传播模型，观察方式和中心性度量指标等基础概念，然后针对常用的三种观察方式简单介绍了几种目前存在的源点检测方法，并讨论了它们各自的优缺点。

2.1 传播模型

在源点检测研究中，传播模型用来描述信息在网络上的传播过程，最常用的传播模型就是传染病传播模型。传染病模型是生物学家通过对病毒传播的研究，建立的相对比较完善的数学模型，其中最基本的是 SI(susceptible-infected) 模型，SIR(susceptible-infected-recovery)模型和 SIS(susceptible-infected-susceptible)模型。为了方便理解起见，我们借用流行病领域的概念来表示网络中节点的状态，一个感染节点意味着该节点所代表的个人相信了流言或者该节点所代表的电脑感染了病毒等等，一个易感染节点意味着其所代表的个人还没有相信流言，计算机也还没被病毒所感染，可以根据具体情况理解易感染和感染的含义。作为源点检测领域的一个基础知识，不同的传播模型适用于不同的传播场景。迄今为止，研究人员主要采用以下三种传播模式：

1. SI 模型：在 SI 模型中，网络中的节点有两种可能的状态：易感染状态(S)和感染状态(I)。传播初始时刻只有传播源节点处于感染状态，其他节点都处于易感染状态。处于感染状态的节点可以将信息传播给易感染节点，而处于易感染状态的节点则可以接受来自感染节点信息。随着传播的进行，易感染节点在接受来自感染节点的信息后变为感染状态，且一旦一个节点被感染之后，它将永远处于感染状态。该模型常用来描述那些感染后不能治愈的疾病，如艾滋病等。
2. SIR 模型：在 SIR 模型中，网络中的节点有三种可能的状态：易感染状态(S)，感染状态(I)和恢复状态(R)。初始时刻只有源节点处于感染状态，其他节点都处于易感染状态。随着传播的进行，易感染状态节点在接受来自感染节点的信息后变为感染状态。当一个节点成为感染状态后，它有一定的概率从感染状态恢复从而成为恢复状态，且一旦成为恢复状态，它将永远处于恢复状态，不会被再次感染。对于像水痘这类治愈后获得免疫的疾病，往往可以用 SIR 模型来描述。
3. SIS 模型：在这个模型中，与 SI 模型类似，网络中的节点有两种可能的状态：易感染状态(S)和被感染状态(I)。但是与 SI 模型不同的是，传播过程有三个阶段，当一个节点被感染之后，它有一定的概率从感染状态中恢复，与 SIR 模型不同的

是恢复后的节点状态变为易感染状态，也就是说它有再次被感染的可能。对于像感冒之类治愈后还可能再次复发的疾病，往往采用 SIS 模型来描述。



是否需要为每个子图写标题

图2.1 三种传染病传播模型

图 2.1 是三种传染病传播模型的节点状态转换图，从上到下分别为 SI、SIR 和 SIS 模型。当然还有一些其他的传播模型，如 SIRS[48]，SEIR[49]，MSIR[50]，SEIRS[51]。但目前源点检测研究中主要针对的是上面这三种模型，其中以 SI 和 SIR 模型为主。

2.2 观察方式

源点检测的主要前提之一就是对网络中的传播情况进行观察，从而获得网络中全部或部分的节点状态信息。在不同的观察方式下，获取的信息会有所差异，所对应的检测方法也会有所不同。根据以往的研究文献，目前存在的观察方式主要有以下三种：

1. 完全观察：在传播过程中的某一时刻观察网络中的节点状态，这种类型的观察可以获得该时刻网络中每个节点的确切状态，具体是处于易感染状态、感染状态还是恢复状态，提供了对网络瞬态状况的全面了解。这种类型的观察使我们有足够的信息来检测传播源点。
2. 快照观察：也称为局部观察，具体是在传播过程中的某一时刻观察网络中的节点状态，但只能获得网络中节点在这一时刻的部分状态信息。部分状态信息的呈现方式有以下四种：
   a. 只能观察到网络中部分节点的状态信息；
   b. 网络中每个节点以一定概率来选择揭示它们是否被感染；

c．网络中所有的感染节点都可以观察到，但无法区分恢复节点和易感染节点；

d．只能观察到在快照拍摄那一时刻被感染的节点的状态信息；

3. 传感器观察：实际应用中网络规模都较大且不是所有节点信息都可以轻易获取，针对这种情况，研究人员提出了传感器观察方式。在网络中选择一些节点作为传感器，通过收集这些传感器节点上的传播信息，来进行源点检测。传感器节点和网络中普通节点的区别在于可以记录发生在各自身上所有的传播细节，包括它们的状态、状态转换时间以及感染方向等。

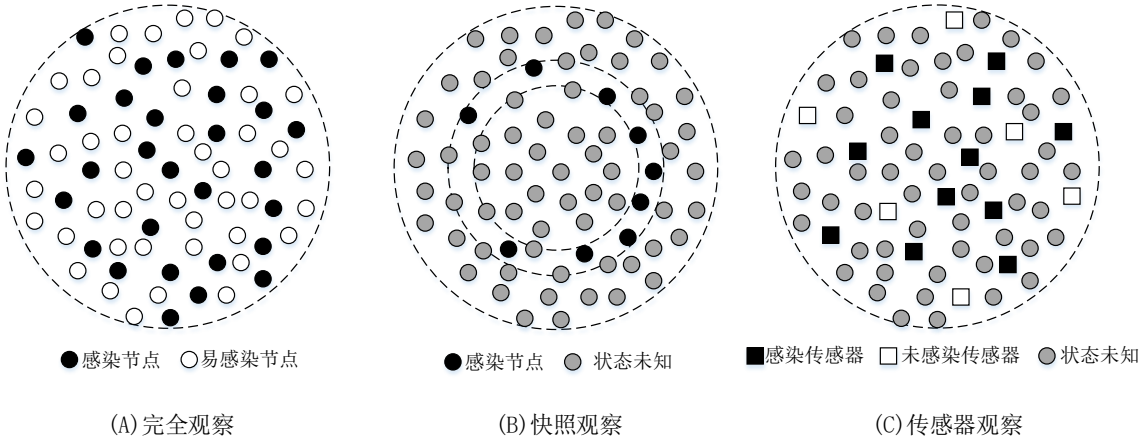


图2.2 三类观察方式

图 2.2 是对三类观测的一个具体说明，其中快照观察中以第四种部分信息呈现方式为例，即我们只能观察到在快照拍摄那一时刻网络中被感染的节点状态信息。显然，与完整观测相比较，快照观察和传感器观测提供的信息要少得多。

2.3 中心性指标

中心性指标可以用来描述一个节点对传播影响。因此，研究人员常常利用各种中心性指标检测可能的传播源节点。我们列出了四个常用的中心性指标：

1. 度中心性(degree centrality)：度中心性是在网络分析中用来刻画节点中心性的最直接度量指标，一个节点的度越大就意味着该节点的度中心性越高，在网络中也越重要。在现实中，影响力大的用户对应于网络中的节点具有相对高的连接度[52]，而且这些高度连接的节点在维持网络连通性方面发挥着至关重要的作用[53, 54]。度中心性的定义为其度与最大可能度的比值，即 $C_D(i)=k_i/(N-1)$，其中 k_i 为节点 i 的度，N 为网络总节点数。
2. 介数中心性(betweeness centrality)：研究人员常常发现网络中一些节点的度不是很大，但在信息传播中却起着十分重要的作用[55, 56]，衡量这类节点的指标叫做介数。一个节点的介数定义为网络中所有最短路径中通过该节点的数量比例，具体如下：

$$B_i = \sum_{\substack{1\le j<l\le N \\ j\neq i\neq l}} [n_{jl}(i)/n_{jl}] \tag{2-1}$$

上式中，n_{jl}为节点 j 和 l 两点间的最短路径数，$n_{jl}(i)$则为这两点间最短路径中经过了节点 i 的路径数，N 为网络中总节点数。节点的介数中心性就是该节点的归一化介数，即$C_B(i) = 2B_i/[(N-1)(N-2)]$。

3. 接近度中心性(closeness centrality)：接近度是拓扑空间里的基本概念之一，节点的接近度反映了节点在网络中处于中心的程度。对于无向连通图来说，一个节点的接近度中心性定义为该节点到其他所有节点最短距离之和的倒数乘以其他节点的个数[54, 57]，具体如下：

$$C_C(i) = \frac{(N-1)}{\sum_{j=1, j\neq i}^{N} d_{ij}} \tag{2-2}$$

上式中，d_{ij} 定义为节点和之间的距离，也就是最短路径上的路阶数，N 为网络总节点数。节点的接近度越大，表明该节点越处于网络的中心，它在网络中就越重要。

4. 特征向量中心性(eigenvector centrality)：特征向量中心性也是节点重要度的测度之一，它指派给网络中的每个节点一个相对得分，对某个节点分值的贡献中，连到高分值节点的贡献比连到低分值节点的贡献大。特征向量中心性通过邻接矩阵 A 来定义，对于节点 i，令它的中心性分值 x_i 正比于连到它的所有节点的中心性分值的总和，则

$$x_i = \frac{1}{\lambda}\sum_{j=1}^{N} a_{ij}x_j \tag{2-3}$$

上式中，N 为网络总节点数，λ 为常数。用向量描述，上式可写为特征向量方程 $Ax = \lambda x$。该方程的解中最大特征值所对应特征向量的归一化分量即为网络中节点的特征中心性。

2.4 相关算法简介

在本节中，我们根据 2.2 节介绍的三种观察方式，将源点检测方法分成了三类，针对这三种观察方式简单介绍了几种目前存在的源点检测方法，并讨论它们各自的优缺点。

(1) 完全观察下的源点检测方法

这种观察方式下所能获得信息最多，但是现实世界中一般不可能实现对网络的完全观察。下面介绍两种基于完全观察的源点检测方法。

a. Single Rumor Center：Shah 和 Zaman[25, 28]引入了 rumor centrality 的概念来进行单

源节点的检测。他们假设信息是在树形网络中传播，且遵循 SI 传播模型，如此，在完全观察下传播源点一定是一个感染节点。同时他们还假设每个节点只能从它邻居中的一个接收信息，假设一个感染节点为源，其 rumor centrality 定义为可以从该点出发向外传播的路径数，具有最大 rumor centrality 的节点称为 rumor center。对于规则树形网络而言，rumor center 可以认为就是传播源节点。对于一般网络，则先使用 BFS 树来表示原始网络，然后再寻找 BFS 树的 rumor center 作为传播源节点。Shah 和 Zaman 等人也证明了对于树形网络来说，rumor center 等价于接近度中心，对于一般网络则不相等。Single Rumor Center 方法在树形网络上可以取得很好的检测准确度，但是它的几个前提假设与现实相差较大。首先，为了消除边界影响，它认为传播过程是在一个类型非常特殊的网络上进行的：无限树形网络。而对于一般网络，在寻找传播源点之前，会将其重建为 BFS 树。其次，隐含地假定信息以单播的方式传播（一个传染性节点一次只能感染一个邻居）。第三，它假设相邻节点之间的感染概率为 1。然而现实世界中的网络比树形网络更复杂，信息经常以多播或广播的方式传播，且相邻节点之间的感染概率也各不相同。

b. Dynamic Age：Fioriti 等人[47]提出了一种在 SI 模型下基于简单的光谱技术的多源节点检测算法。他们首先利用了 Zhu[65]等人研究的特征值与节点年龄之间相关性的概念。Zhu 等人认为，对于一个网络而言，其邻接矩阵或拉普拉斯矩阵的特征向量与网络中节点的年龄有着某种紧密的联系，特征向量的分量值越大，说明其对应的节点在网络中存在的时间越长，年龄也越大。其中拉普拉斯矩阵定义如下：

$$L = D - A \tag{2-4}$$

上式中，A 为网络邻接矩阵，即节点 i 和 j 有连边则 $a_{ij}=1$，反之则 $a_{ij}=0$。D 则是度对角矩阵，d_{ii} 为节点 i 的度。此外，Fioriti 等人还结合了 Restrepo 等人提出节点动态重要性概念[58]。动态重要是评价网络中节点重要度的一个指标，其本质是计算一个节点被移除后邻接矩阵最大特征值的减少量。最终，Fioriti 等人提出了 dynamic age，其本质是计算节点被去除之后邻接矩阵的最大特征值的减少量，具有最大减少量的节点被认为是源节点，dynamic age 定义如下：

$$DA_i = \left|\lambda_m - \lambda_m^i\right| / \lambda_m \tag{2-5}$$

上式中，λ_m 是邻接矩阵的最大特征值，λ_m^i 是在移除节点 i 之后邻接矩阵的最大特征值。该方法中需要计算邻接矩阵的特征值，而矩阵特征值的计算复杂度比较高，一般为 $O(N^3)$，该方法不太适用于大型网络中的源点检测。而且，由于没有一个确定性的阈值来判断哪些节点是最老的节点，因此无法确定传播源节点的真实数目。

(2) 快照观察下的源点检测方法

在现实世界中，对整个网络进行完全观察是不可能实现的，特别是大型网络。快

照观察是相对更为接近现实的一种观察方式，它只能提供网络传播的部分知识。这里介绍两种基于快照观察的源点检测方法。

a. Jordan Center：Zhu 和 Ying[34]提出了一种新的中心性指标来进行单源节点检测，Jordan Center。假设信息在树形网络中传播，且遵循 SIR 传播模型，网络中所有感染节点都是已知的，但是无法区分易感染节点和恢复节点。他们认为从某个节点出发的传播最有可能导致目前观察到的网络快照，那该节点就最有可能为传播源节点，他们称这种方法为最优采样路径法。首先在树形网络上研究了最优采样路径的结构性质。通过定义一个节点到所有感染节点的最大距离为该节点的感染偏离度，他们证明最优采样路径法所对应的传播源节点就是感染偏离度最小的节点，即 jordan center。与 Single Rumor Center 类似，Jordan Center 方法适用的也是无限树形网络，这与现实世界的网络有很大的不同，该方法也只能用于单源检测。

b. 基于有效距离的方法：假设传播 SI 传播模型，Brockmann 和 Helbing 提出了一种基于有效距离的源点检测方法[37]。假设在快照观察下，仅能获得波前的感染情况，即仅能观察到在快照拍摄那一时刻被感染的节点状态信息。首先提出了一个新的概念，有效距离，来表示传播过程。从节点 i 到相邻节点 j 的有效距离 d_{ij} 定义为

$$d_{ij} = 1 - \log p_{ij} \tag{2-6}$$

上式中，p_{ij} 是节点 i 到节点 j 的传播概率。在用有效距离表示传播过程后，网络上的传播可看作均匀波向外扩散的形式，这样根据波前的传播过程，若一个节点是真实传播源，那该节点到所观察波前的有效距离的标准偏差和均值应该最小。网络中的信息传播过程是复杂的。由于现实网络的多尺度性与异质性，真实的传播过程及其复杂且很难直观地理解，而 Brockmann 和 Helbing[37]利用有效距离将其简化为简单的波前传播过程，有助与传播过程的理解。要使用基于有效距离的方法进行源识别，需要计算从任何可疑源到观察到的被感染节点的最短路径距离，计算复杂度较高，该方法与 Dynamic Age 一样也不适用于大规模网络。

(3) 传感器观察下的源点检测方法

在现实世界中，传播所在的网络规模都比较大，一般的源点检测方法的复杂度较高，不太适用于这类网络。而且由于涉及个人隐私以及信息咨询代价等问题，现实中不可能实现对网络中每个节点的观察。针对这种情况，一般是通过提前在网络选择一些节点作为传感器，利用传感器收集到的信息来实现传播源点的检测。传感器所能收集的信息一般包括传播到达的方向以及传播到达的时间等。下面介绍两种基于传感器观察的源点检测方法。

a. 高斯估计法：假设传播是在树形网络上进行的，且遵循 SI 传播模型，Pinto 等人提出了一种单源检测方法，高斯估计法[44]。假定网络中每个边上都有一个固定的传播时间，这些时间是独立的，并且遵循高斯分布。该方法分为两步，在第一步

中，首先根据信息到达传感器的方向，唯一地确定一课子树T_α，且确保这颗子树包含了传播源点[44]，从而减少了可能传播源的搜索范围。在第二步中，首先任意选择一个传感器o_1，并且计算该传感器与其他传感器之间的“观察延迟”。然后假设任意一个节点$s \in T_\alpha$作为源，通过使用边的确定性传播时间来计算每个传感器节点相对于传感器o_1的“确定性延迟”。最后选择能够最小化传感器节点“观测延迟”与“确定性延迟”之间误差的节点作为传播源点。该方法是针对树形网络提出的，对于一般网络 Pinto 等人假定信息沿着 BFS 树传播，然后在 BFS 树中寻找传播源节点。高斯估计法的复杂度为$O(N^3)$，不适用于大型网络。

b. 四指标估计法：Seo 等人[45]提出了一个四指标估计法来检测网络中的单传播源节点。假设传播遵循 SI 传播模型，若一个传感器节点被感染，则将该传感器视为正传感器，否则该传感器为负传感器。假设源节点必须靠近正传感器节点，且远离负传感器节点，提出了四种指标来定位传播源点。首先，找出一组可以到达所有正传感器的节点集合。其次，通过选择到所有正传感器节点的距离总和最小来进一步过滤掉集合中的部分节点。之后，他们选择可到达最小数量的负传感器的节点作为可能源点集合。最后，将满足上述三个指标且到达所有负传感器节点的距离和最大的节点视为源节点。四指标估计法需要计算从传感器到任何可能源节点的最短路径，计算复杂度是$O(N^2)$。

2.5 本章小结

本章简要介绍源点检测的相关基础知识，首先介绍了传播模型，观察方式和中心性度量指标等基础概念，然后针对常用的三种观察方式简单介绍了几种目前存在的源点检测方法，并讨论了它们各自的优缺点。

第三章 基于 SI 传播模型的多源节点检测算法

本章研究了 SI 传播模型下的多源节点检测问题，假设信息传播遵循 SI 模型且传播源个数已知，提出了一种简单的 KST 算法来识别一般网络中的多个传播源节点。此外，由于实际情况中初始传播源个数一般不可提前获知，还提出了一个可以估计传播源个数的启发式算法。最后通过实验仿真，证明了提出的算法在多源节点检测方面的优势。

3.1 引言

随着城市化的快速推进和通讯技术的发展，现实生活中的各种连接关系变得越来越紧密，这使得我们日常生活变得更加便利的同时也更容易受到各种传播风险的危害。例如，SARS，H1N1 或埃博拉病毒等传染病的广泛传播，造成数十万人甚至数百万人死亡；Cryptolocker 和 Alureon 这样的电脑病毒在网络安全事件中占有很大份额；Facebook，Twitter 等在线社交网络上谣言的迅速扩散等等。这些有危害的传播在爆发后，人们常常关心的关键问题有如下几个：有多少个传播源？传播从哪里开始出现？传播可以扩散到哪里？如果能准确解决这些问题，就可以预测传播蔓延的方向，从而制定及时的缓解策略甚至进一步消除危害的传播。正因如此，复杂网络中的源点检测问题在最近几年越来越倍受关注。

现有的大部分源点检测方法都集中在树形网络的单源检测问题上，甚至假设网络是每个节点具有相同数量邻居节点的规则树形网络。由于树形网络中不包含循环连接，节点对之间只有一条路径，降低了传播的不确定性和复杂度。然而，现实世界的网络拓扑结构比树形网络复杂的多，基于树形网络的源点检测方法显然不能适用于真实网络。而且，现实传播中信息传播源一般不会只有一个，从多个传播源发散出来的信息会相互影响，使得多源传播过程更为不确定。由于多源传播的时空复杂性以及传播过程的不确定性，使得多源检测的难度大大增加，现有的方法却很少有用于多源检测的。

在本章中，我们提出了一种 KST 算法来检测一般网络中的多个传播源。首先将感染网络分成 k 个分区，然后在每个分区中找到一个节点作为该分区的传播源节点，该节点可以在最短的时间内使得传播覆盖整个分区。最后通过不断最小化所有分区传播时间之和确定最终的传播源节点。KST 方法可应用于一般网络来识别多个来源，而且该方法的计算复杂度为 $O(n^2)$，相对其他多源检测算法来说效率更高。

3.2 预备知识

本节中详细介绍了所要用到的 SI 传播模型[59]以及常用的基于最小路阶数的传播时间估计方法与网络中任意节点对之间传播概率的计算方式。

3.2.1 SI 传播模型

SI(susceptible-infected)模型是经典流行病传播模型，通常被用来描述一些不可治愈的流行性疾病或缺乏有效控制的突发性事件。因为生物学家很早就开始对疾病传播进行了研究，并且建立了比较完善的流行病学传播的数学模型，时至今日流行病模型仍然是我们研究信息传播的基础[60]。

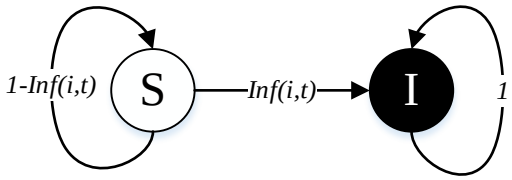


图3.1 SI 模型节点状态转换图

如图 3.1 所示，在 SI 传播模型中，网络中的节点有两种可能的状态：易感染状态(S)和感染状态(I)。传播初始时刻，除了处于感染状态的传播源之外，网络中的其它节点都处于易感染状态。网络中的感染节点通过一定的概率将疾病传染给它的邻居节点，一旦一个 S 状态的节点被感染，则它成为 I 状态节点，此时它也成为了一个新的感染源，可以将疾病传播给它的邻居节点。而且，成为 I 状态之后，节点的状态不再发生变化，将一直处于 I 状态。

假设传播在时间 $T=0$ 时开始，我们使用 $P_S(i,t)$ 和 $P_I(i,t)$ 分别表示节点 v_i 在时间为 $T=t$ 时处于易感染状态和感染状态的概率。初始时刻也就是 $T=0$ 时刻，所有的传播源节点都处于感染状态，它们对应的 $P_S(i,0)=0$ 、$P_I(i,0)=1$。同样，所有易感染节点都具有 $P_S(i,0)=1$ 和 $P_I(i,0)=0$ 的初始状态。然后利用式(3-1)和式(3-2)，可以得到每个节点在任意时刻处于各个状态的概率。

$$P_S(i,t)=[1-Inf(i,t)]\cdot P_S(i,t-1) \tag{3-1}$$

$$P_I(i,t)=Inf(i,t)\cdot P_S(i,t-1)+P_I(i,t-1) \tag{3-2}$$

其中 $Inf(i,t)$ 表示节点 v_i 在 $T=t$ 时刻被其邻居节点所感染的概率。$Inf(i,t)$ 计算方式如下：

$$Inf(i,t)=1-\prod_{j\in N_i}[1-\eta_{ji}\cdot P_I(j,t-1)] \tag{3-3}$$

上式中，η_{ji} 表示从节点 v_j 到其相邻节点 v_i 的传播概率，N_i 表示节点 v_i 的邻居节点集合。可以看出该模型通过解析的方式导出任意时刻每个节点处于各个状态的概率。当

然在实际应用中，每个时间点的长度取决于真实环境，它可以是一分钟，一小时或一天，此外还需要适当地设置节点之间的传播概率。

3.2.2 基于最小路阶数的传播时间估计方法

根据上面介绍的 SI 传播模型，一个感染节点要将疾病传染给它的未感染邻居至少需要一个时钟周期，而一个节点和它的邻居节点之间只有一个路阶数，这意味着可以根据路阶数来估计传播时间。因此，对于网络中的任何一对节点，如果疾病从它们两者中的一个节点传播到另一个节点，传播所需要的时间就是这两者之间的最小路阶数[61]。

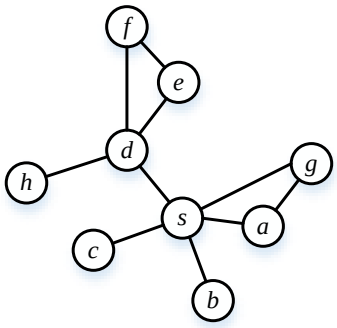


图3.2 传播时间说明图

如图 3.2 所示，给定一个具有单传播源 s 的网络 G_n，首先引入 $h(s,v_i)$ 来表示传播源 s 与网络中任一节点 v_i 之间的最小路阶数，即最短路径上的边数，如节点 s 到节点 g 的路径有两条，其中最短路径的长度为 1，那么传播从节点 s 到节点 g 所需的时间为 1。对于从 s 出发的传播要感染整个网络所需要的时间则可用式(3-4)来表示：

$$t = \max\{h(s,\upsilon_i) \mid \upsilon_i \in G_n\} \tag{3-4}$$

式(3-4)表明，要感染整个网络，从源点 s 出发的传播应该能在 t 时间内到达网络中的每个节点，即 t 应该为 s 到其他节点的最大路阶数。如图 3.2 中，该网络中以 s 出发的传播要感染整个网络所需时间最小为 2。

在现实世界中，疾病在传播过程中会受到各种各样因素的影响。在这个领域，大多数对传播时间的估计都是这种基于最小路阶数的估计方法，而且这种估计方式也简化对传播过程的理解[62]。因此，我们也采用这种基于最小路阶数的传播时间估计方法。

3.2.3 网络中任意两点间的传播概率

在源点检测领域中关于网络中任意两点间传播概率的计算，大都采用的是基于最短路径的计算方式，即认为网络中任意两点间信息的传播是沿着这两点间最短路径进行的，最短路径上的传播概率即为这两点间的传播概率。之所以这样认为是因为一般情况下最短路径相对其他路径来说拥有的传播概率最大，其他路径上的传播概率相对

可以忽略不计。

给定一个复杂网络G_n，假设η_{ij}表示感染节点v_i将疾病传染给与它直接相连的邻居节点v_j的传播概率。为了计算网络中任意两点间的传播概率，先对网络中的每条边进行如下转换：

$$q(i,j) = -\log(\eta_{ij}) \tag{3-5}$$

其中η_{ij}是节点v_i和它邻居节点v_j间连边上的传播概率。然后以$q(i,j)$值作为网络连边上的权值，可以用任何计算最短路径的算法找到网络中任意两点间的最短路径，本文中用的是 Dijkstra 算法。这样我们就可以得到网络中任意两点间的$q(i,j)$值，再利用式(3-6)将$q(i,j)$值转换为任意两点间的传播概率：

$$p(i,j) = e^{-q(i,j)} \tag{3-6}$$

$p(i,j)$就是网络中任意两点间最短路径上的传播概率。

3.3 KST 算法

在本节中，我们提出了一个可用于一般网络的多源节点检测算法，KST 算法。首先给出了 KST 算法的理论依据，接着对 KST 算法进行了详细说明，最后给出了 KST 算法的部分相关证明。

3.3.1 KST 算法理论依据

在现实世界中，由于网络的多尺度性和内在异质性，网络中的传播过程具有时空复杂性[61]，假设发生在网络上的传播过程遵循 SI 传播模型。在一个网络上，初始时刻有$k(k \geq 1)$个传播源$S^* = \{s_1, ..., s_k\}$，并且这些传播源在$T = 0$时刻同时开始向外传播[32, 47]。经过若干时间后，网络中一部分节点的状态会由易感染状态变为感染状态。我们将这些感染状态的节点所组成的网络称为感染网络G_I。对于每个传播源s_i来说，它们都有各自的感染区域$C_i(\subseteq G_I)$。换句话说，感染网络G_I可以看作由 k 个感染区域$C^* = \{C_1, C_2 ..., C_k\}$组成的，并且这 k 个感染区域满足$C^* = \bigcup_{i=1}^{k} C_i$和$C_i \bigcap C_j |_{i \neq j} = \varnothing$。每个区域中只有一个传播源节点，且该区域中的其他节点都可以看作是由该传播源直接或间接感染的。

给定感染网络G_I和传播源个数 k，我们的目标是找到出一组传播源节点集S^*。对于任意一个感染节点，假设它被感染过程需要的时间很短，因此我们认为它是被传播到达它所需时间最短的传播源所感染的。根据之前的分析，我们可以将感染网络划分为 k 个分区，使得属于每个分区中的所有感染节点到该分区相对应的传播源的传播时间最短。我们希望得到的这个对感染网络G_I的划分能够尽可能的与网络的真实感染区域相似。

在每个分区中，我们需要找到一个新的节点作为每个分区的传播源节点。根据前

面的介绍，假设每个节点是被传播到达它所需时间最短的传播源所感染且每个分区中所有节点都是被同一个传播源所感染，那么传播源节点应该满足以它开始的传播能在最短的时间内将整个分区所感染，换句话说，我们需要在每个分区中找到一个节点，从该节点到分区中其他节点的传播时间最大值最小。

通过以上分析，我们为多源节点检测问题定义了一个目标函数，目标函数的目的是找到一个对感染网络合适的划分方式，使得可以尽可能地降低所有分区传播时间的总和。该目标函数如下：

$$\min_{C^*} f = \sum_{i=1}^{k} t_i \tag{3-7}$$

其中 t_i 是分区 C_i 被完全感染所需要的传播时间。根据前面提到的估计传播时间的方法，$t_i = \max\{h(s_i, v_j) \mid v_j \in C_i\}$。式(3-7)表明，我们的目的是找到一个可以使每个分区的传播时间总和最小的划分，之后将每个分区中具有最小传播时间的节点视为传播源节点。

3.3.2 KST 算法细节介绍

我们将详细介绍 KST 算法过程，同时也给出 KST 算法的伪代码。

(1) 初始化可能源节点集

给定感染网络 G_I 和传播源个数 k，我们希望根据一组传播源将感染网络划分为 k 个区域。首先需要选择 k 个节点作为初始源节点集，这里我们从感染网络中选取 k 个距离最远的节点作为初始源节点集，因为以这种选取方式可以最大程度地保证这 k 个节点是由不同的传播源所感染的，即它们身上的传播可以追溯到不同的传播源节点上[41]。为了选择初始源节点集，我们先从感染网络中选择一对距离最近的感染节点，然后以此贪婪地选择其他 k-2 个节点，直到源节点集的数目为 k。详细算法过程如算法 3.1 所示。

(2) 感染网络划分

在有了一个初始源节点集后 $S^*=\{s_1,...,s_k\}$，我们要将给定感染网络 G_I 根据这组源节点集划分为 k 个分区。根据之前的分析，对于任意一个节点 $v_j \in G_I$，它应该是被传播到达它所需时间最短的传播源所感染的，即它应该被划分到与该传播源所对应的分区中去，也就是满足式(3-8)的传播源：

$$t(v_j, s_i) = \min_{s_l \in S} \left\{ t(v_j, s_l) \right\} \tag{3-8}$$

其中 $t(v_j, s_l)$ 表示节点 v_j 与传播源节点 s_l 之间根据最小路阶数所估计的传播时间，$t(v_j, s_i)$ 则指明应该将节点 v_j 划分到满足该式子的传播源节点 s_i 所对应的分区中。

感染网络划分算法的具体实现细节在算法 3.2 中给出。对于感染网络中某一节点 v_j，当我们以上面这种方式进行划分时，有时会出现两个或两个以上的源节点满足条件，这是因为我们采用的是基于最短路阶数来估计传播时间的方式。为了将这类节点

进行划分，我们将进一步比较这些传播源节点到节点 v_j 的传播概率，并将节点 v_j 划分到具有最大传播概率的传播源节点所对应的分区中去。网络中任意两点的传播概率的计算方式参考 3.2.3 小节。

Algorithm 3.1：Initialize Sources Set.

Input：An infection network G_I and source number k

Select two infected nodes s_1 and s_2 with the maximum distance(hops),i.e., $d(s_1,s_2)=\max_{a,b\in G_n}\{d(a,b)\}$, and let $S^0=\{s_1^0,s_2^0\}$.

Let $i=|S^0|$ and select an infected node $s_{i+1}^0\in G_I\setminus S^0$ such that

$$d(s_{i+1}^0,S^0)=\max_{a\in G_n\setminus S^0}\left(\min_{b\in S^0}d(a,b)\right)$$

i.e., selecting an infected node from $G_I\setminus S^0$ that is furthest away from set S^0.

Repeat this step until $|S^0|=k$.

Output：A set of sources $S^0=\{s_1^0,s_2^0,...s_k^0\}$.

Algorithm 3.2：Network Partition.

Input： A set of source $S=\{s_1,s_2,...,s_k\}$ in infection network G_I.

Initialization：Initialize *k* partitions：$C_1=\{s_1\},...,C_k=\{s_k\}$.

for (*j* start from 1 to *N*) **do**

Find the nearest source to node v_j as follows, $t(v_j,s_i)=\min_{s_i\in S}t(v_j,s_l)$.

if (there is one source s_i satisfies the condition)

Classify the node v_j to the partition C_i.

else

Find the maximum propagation probability form v_j to these source which satisfy above condition. And classify the node v_j to the corresponding partition.

end if

end for

Output：A partition of $G_I:C^*=\bigcup_{i=1}^k C_i$.

(3) 多源节点检测

在 3.3.1 小节给出了 KST 算法所对应的目标函数式(3-7)，我们的目的就是要最小化该目标函数。在给定感染网络 G_I 和传播源个数 $k(k\geq 1)$ 的前提下，先使用算法 3.1 来选择初始源节点集合，然后再以算法 3.2 的方式将感染网络划分为 k 个分区，通过

该步骤可以得到目标函数的一个局部最优解。为了进一步最小化目标函数，需要为每个分区重新选择一个新的源节点。根据式(3-7)，我们为每个分区选择一个可以使该分区被完全感染所需时间最小的节点作为新的传播源节点。这也是我们为什么称该算法为 KST(K Shortest spreading Time)算法的原因，因为我们的目标是想找到 k 个节点，使得从这 k 个节点出发的传播可以用最短传播时间到达网络中每一个感染节点，即可以以最短的时间来感染整个感染网络。这里也给出了相应的伪代码形式，如算法 3.3 所示，KST 算法的完整过程参见图 3.3。 引文

可以看出 KST 算法与数据挖掘领域的 K-means 算法十分相似。与 K-means 算法一样，KST 算法并不能保证能够取得目标函数的全局最优解，但这不影响 KST 算法在大多数情况下可以取得一个比较好的检测准确度。对于 KST 算法而言，它主要的计算花费是网络中任意两点间最短路径的计算，而相对来说其他计算量都可以看作是一个常数。在实验中，使用 Dijkstra 算法来计算网络中任意两点间的最短路阶数路径，它的计算复杂度为 $O(n^2)$，因此 KST 算法的计算复杂度也为 $O(n^2)$。

Algorithm 3.3：Identifying Diffusion Sources.

Input：An infection network G_I and the number of sources k.

Initialization：Initialize an positive integer L and choose a set of sources $S^{(0)}=\{s_1^{(0)},s_2^{(0)},...,s_k^{(0)}\}$ as **Algorithm 3.1**.

for (i start from 1 to L) **do**

Use **Algorithm 3.2** to partition G_I with center $S^{(0)}$, and obtain a partition $C^l=\bigcup_{i=1}^{k}C_i^{(l)}$.

Find the new center in each partition $C_i^{(l)}$ which has the minimum spreading time.

if ($S^{(l)}=\{s_1^{(l)},s_2^{(l)},...,s_k^{(l)}\}$ is the same as the $S^{(l-1)}$)

stop

end if

end for

Output：A set of estimated sources $S^{(l)}=(s_1^{(l)},s_2^{(l)},...,s_k^{(l)})$.

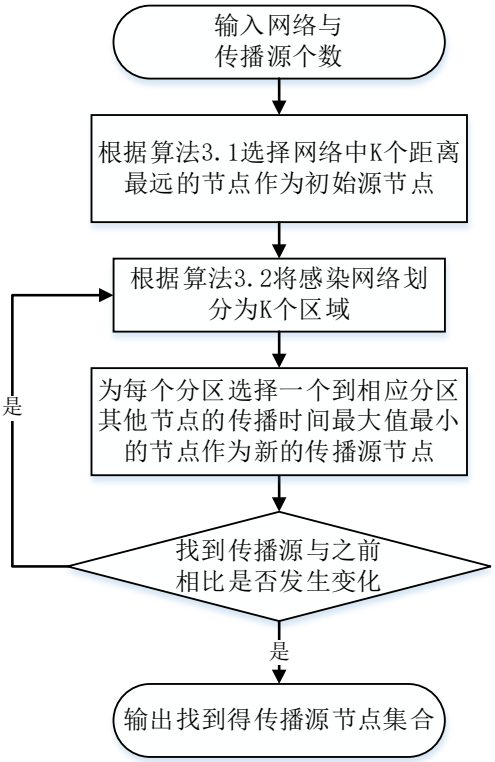


图3.3 KST 算法流程图

3.3.3 KST 算法相关证明

KST 算法通过不断迭代来优化目标函数，也就是说式(3-7)的值要在迭代过程中满足单调递减性，这里对 KST 算法的收敛性予以证明：

证明：假设在第 m 迭代后，我们得到一组传播源节点集 $S^m=\{s_1^m,s_2^m,...s_k^m\}$，然后我们会用感染网络划分算法将感染网络 G_I 划分为 k 个分区，即 $C^m=\bigcup_{i=1}^{k}C_i^m$，这时我们的目标函数为：

$$f^m=\sum_{i=1}^{k}t_i^m \tag{3-9}$$

其中 $t_i^m=\max\{h(s_i^m,v_j)\,|\,v_j\in C_i^m\}$。在第 $m+1$ 次迭代中，根据 KST 算法我们会为每个分区 C_i^m 重新找到一个新的传播源 s_i^{m+1}，从而得到一个新的传播源节点集 $S^{m+1}=\{s_1^{m+1},s_2^{m+1},...s_k^{m+1}\}$。根据 KST 算法，$s_i^{m+1}$ 应该选择能以最短的时间将分区 C_i^m 完全感染的节点，也就是说 $t_i^m\geq t_i^{m+1}$，因此第 $m+1$ 次的目标函数值

$$f^{m+1}=\left\{\sum_{i=1}^{k}t_i^{m+1}\right\}\leq\left\{\sum_{i=1}^{k}t_i^{m+1}\right\}=f^m \tag{3-10}$$

所以说目标函数值是单调递减的。

3.3.4 实验与分析

(1) 实验设置

我们分别在 Power Grid[35]和 Yeast protein-protein interaction network[36]这两个真实

网络上验证了 KST 算法的有效性。表 3.1 是这两个真实网络的基本统计参数，其度数分布如图 3.4 所示。使用的电脑配置：cpu 是 inter(R) core(TM) i5-6500 3.20GHZ, 8G 内存，windows 10 操作系统，程序使用 matlab 编写。

表3.1 网络统计参数

数据集	Power Grid	Yeast
节点数	4941	2361
连边数	13188	13554
平均度	2.67	5.74
最大度	19	64

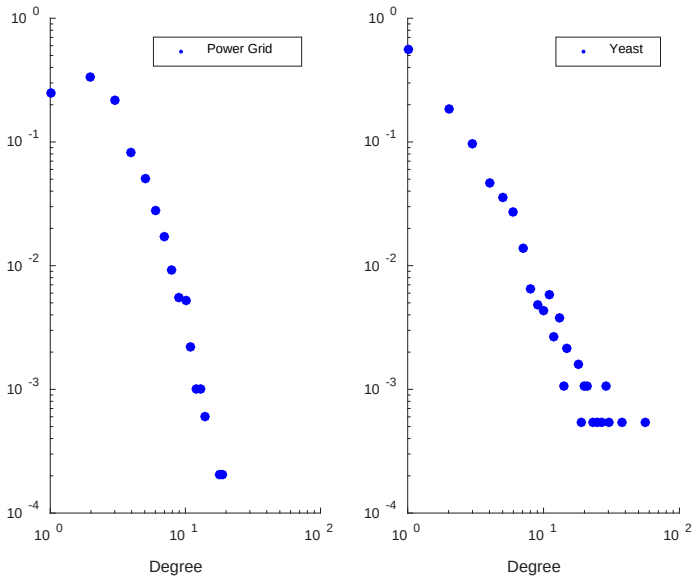


图3.4 网络度分布图

假设传播遵循 SI 模型且所有传播之间相互独立，设置网络连边上传播概率服从 (0,1) 之间的均匀分布。已有工作[59, 63]证明，传播概率的分布并不会影响 SI 传播模型的精度，均匀分布已经足以用来评估所提出方法的性能。类似的传播概率设置方式也可在文献[34]，文献[41]和文献[64]中找到[34, 41, 64]。

我们设定传播源节点的个数由 2 到 5，并且针对每种设置分别进行 100 次实验，最终统计结果为这 100 次检测结果的平均值。对于每次实验，先从网络中随机选择给定数目的节点作为传播的初始源节点，然后根据 SI 模型模拟传播进行的过程，图 3.5 给出了具体的模拟流程图。当传播过程进行到一定程度时，在得到的感染网络上运用 KST 算法进行源点检测。

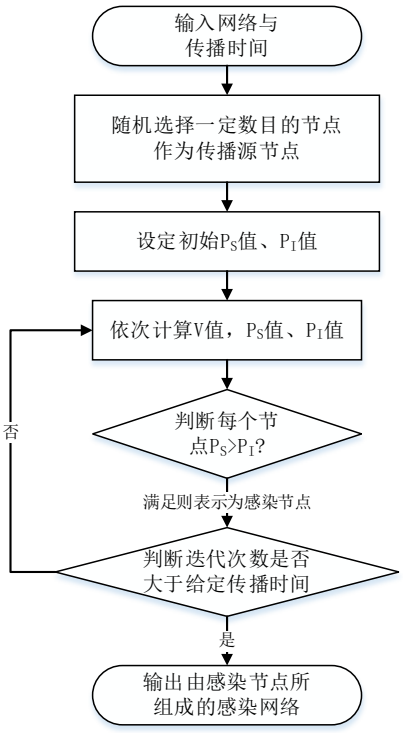


图3.5 SI 模型模拟传播流程图

(2) 评价指标

评价源点检测算法性能常用的评价指标是平均误差距离，用参数Δ来表示，它表示检测出的源节点与真实源节点之间平均相距距离的大小。在多源节点检测问题中，我们首先需要将检测出来的传播源节点与真实的传播源节点进行匹配，匹配的方法是使得所有真实传播源节点和与它相匹配的估计传播源节之间的误差距离和最小[28, 32]。平均误差距离由式(3-11)给出：

$$\Delta=\frac{1}{\left|S^*\right|}\sum_{i=1}^{\left|S^*\right|}h(s_i,\hat{s}_i) \tag{3-11}$$

其中 s_i 属于真实传播源节点集 $S^*=\{s_1,...,s_k\}$，$\hat{s}_i$ 属于检测出来的传播源节点集 $\hat{S}=\{\hat{s}_1,...,\hat{s}_k\}$。平均误差距离Δ越小，表明检测出来的传播源节点越靠近真实传播源节点。我们希望我们的方法可以尽可能准确地检测出真实传播源节点，或者至少是一组非常接近真实传播源节点的节点集。

(3) 对比算法

a. K-Center 算法

蒋等人[61]提出了 K-Center 方法来检测一般网络上的多个传播源节点。他们首先采用有效距离[37]转换原始网络，有效距离的定义如下：

$$e(i,j)=1-\log(\eta_{ij}) \tag{3-12}$$

其中 η_{ij} 为节点 v_i 到其邻居节点 v_j 的传播概率。通过使用有效距离，复杂的时空扩散过程可以简化为均匀波传播的模式[61]。基于转换后的网络，他们推导出一个多源检测

问题的目标函数。通过迭代的方式不断最小化该目标函数，最终选出的 k 个中心节点即可认为是传播源节点。具体的，首先从网络中随机选择 k 个节点作为初始源节点，之后将网络中的其他节点根据与这 k 个源节点的有效距离的大小划分为 k 个区域，每个节点应该被划分到其与传播源之间有效距离最小的传播源所对应的区域中。然后在每个区域中找一个到该区域中其他节点的有效距离和最小的节点作为该区域新的传播源节点。最后执行上述迭代过程直到选出的传播源不再发生变化为止，则认为这些节点为传播源节点。

b. Dynamic Age 算法

Fioriti 等人[47]提出了一种简单的基于光谱技术的多源节点检测方法，Dynamic Age。Zhu 等人[65]研究了的网络邻接矩阵特征值与节点年龄之间相关性的概念。他们认为网络邻接矩阵中最大特征值往往对应着网络中最老的节点。Restrepo 等人提出了[58]节点的动态重要性概念，其本质是计算节点被去除之后邻接矩阵最大特征值的减少量。去除节点之后最大特征值的大量减少意味着节点与扩散的老化相关。Fioriti 等人通过结合这两种技术，提出了 Dynamic Age 的概念。对于网络中任意节点 v_i，他的 Dynamic Age 值为：

$$DA_i = \frac{\left|\lambda_m - \lambda_m^i\right|}{\lambda_m} \tag{3-13}$$

其中 λ_m 是邻接矩阵的最大特征值，λ_m^i 是节点 v_i 被移除之后邻接矩阵的最大特征值。节点具有最高的动态年龄被认为是来源。给定由原网络中感染节点所组成的感染网络，该方法是为每个节点计算 DA_i 值并对其进行排序，更高的 DA_i 值表示该节点更有可能是初始的传播者，即具有最大 Dynamic Age 的节点可认为是传播源节点[47]。

(4) 实验结果与分析

表3.2 三种算法平均误差距离统计结果

实验设置		平均误差距离		
网络	源点个数	KST	K-Center	Dynamic Age
Power Grid	2	1.39	1.78	2.593
	3	1.67	2.373	3.434
	4	1.775	2.725	3.957
	5	1.893	2.152	4.467
Yeast	2	0.925	1.721	3.057
	3	0.98	2.222	3.823
	4	1.15	2.383	3.7
	5	1.226	3.04	4.133

表 3.2 给出了 KST 算法和两个对比算法分别在 Power Grid 网络和 Yeast 网络上的检测结果。从检测结果来看，在 Power Grid 网络上，当传播源个数$|S|$=2 时，KST 算法检测出的传播源节点距离真实传播源节点的平局误差距离为 1.39，相对来说优于 K-Center 算法的 1.78 和 Dynamic Age 算法的 2.593；当传播源个数$|S|$=5 时，KST 算法的平均误差距离为 1.893，而 K-Center 算法的则为 2.725，Dynamic Age 则高达 4.467。在 Yeast 网络上，KST 算法同样具有明显的优势，当传播源个数$|S|$=2 时，KST 算法检测结果仅为 0.925，而 K-Center 则为 1.721，Dynamic Age 高达 3.057；当$|S|$=3 时，KST 的检测结果仍在一个路阶数内，而 K-Center 算法的结果在 2-3 路阶数内，Dynamic Age 则在 3-4 个路阶数内。可以看出，无论是在 Power Grid 网络还是 Yeast 网络上，KST 算法都可以取得相对较好的检测结果。

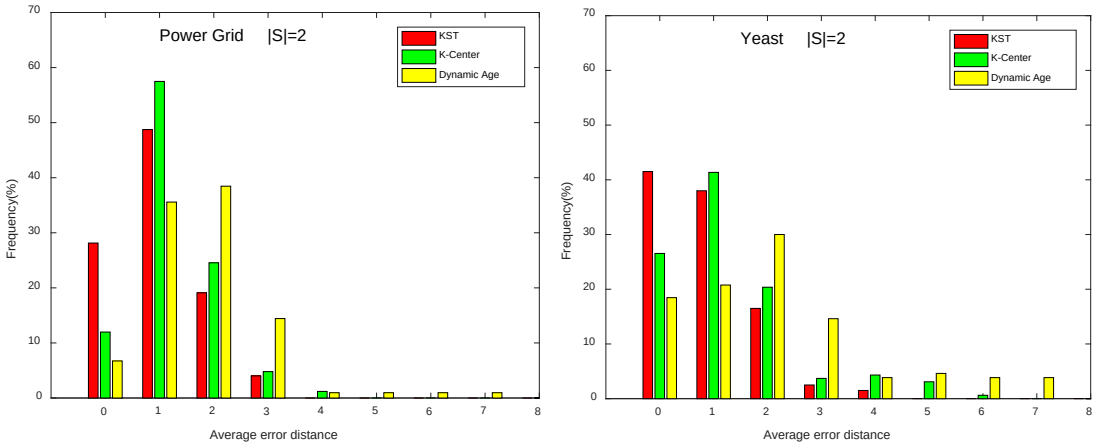


图3.6 源节点为 2 时三种算法平均误差距离直方图

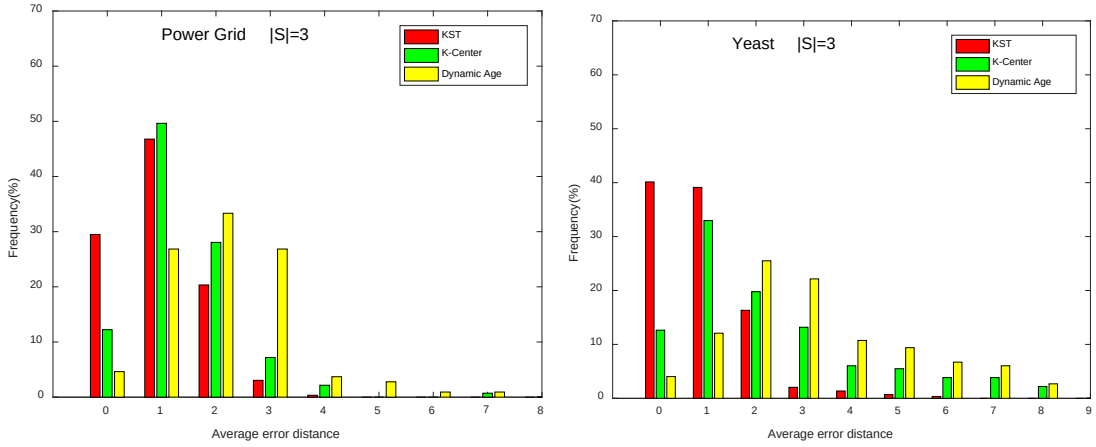


图3.7 源节点为 3 时三种算法平均误差距离直方图

为了对检测结果有一个更直观的理解，我们还给出当传播源个数分别为$|S|$=2 和

$|S|$=3时，三种算法检测结果的分布直方图。图 3.6 是源节点数目$|S|$=2时在 Power Grid 和 Yeast 网络上的检测结果，以 Power Grid 网络上的结果来，KST 算法大约有 80%的概率检测出的源节点距离真实源节点在 1 个路阶数内，且其中有 30%左右的概率可以完全准确地检测出传播源节点。相对来说虽然 K-Center 的检测结果也有 70%的概率在 1 个路阶数内，但它只有 10%左右可以准确检测出真实传播源节点。而 Dynamic Age 的检测结果就更差了，仅有 40%左右在 1 个路阶数内。图 3.7 是源节点为 3 时三种算法的检测结果，同样可以看出 KST 算法的检测结果优于其他两个对比算法。总的来说，KST 算法检测出的源节点更为接近真实传播源节点，检测结果优于两个对比算法。

3.4 KST-Improved 算法

3.4.1 有效传播时间

在 KST 算法中我们采用基于最小路阶数的方式来估计网络中任意两点间传播时间，这种估计方式虽然可以简化信息传播的过程，但是它没有考率传播概率对传播时间的影响。而实际中，传播概率对传播时间的影响很大，即使距离（路阶数）相同的情况下，传播概率不同所需要的传播时间也会有所不同。针对这种情况，我们提出了有效传播时间(effective spreading time)来估计网络中任意两点间的传播时间，其定义如下：

$$et(i,j) = -\log p_{ij} \tag{3-14}$$

其中 p_{ij} 是网络中任意两点 v_i 和 v_j 之间的传播概率， p_{ij} 的计算方式参考 3.2.3 小节。

我们使用有效传播时间来替代 KST 算法中基于最小路阶数的估计方式来进行网络中任意节点对之间传播时间的估计。这时，KST 算法中的网络划分步骤变为根据有效传播时间划分。对于任意节点 $v_j \in G_I$，它应该被划分到满足下式的传播源 s_i 所对应的分区 C_i 中：

$$et(v_j, s_i) = \min_{s_l \in S} et(v_j, s_l) \tag{3-15}$$

我们称这种改进后的 KST 算法为 KST-Improved 算法。

3.4.2 实验与分析

(1) 实验设置

实验设置与 3.3.4 小节中的实验设置一致，假设传播遵循 SI 模型且所有传播之间相互独立，网络连边上传播概率服从 (0,1) 之间的均匀分布，设定传播源节点的个数由 2 到 5，并且针对每种设置分别进行 100 次实验，最终统计结果为这 100 次检测结果的平均值。

(2) 实验结果与分析

表3.3 KST 方法和 KST-Improved 算法平均误差距离统计结果

实验设置		平均误差距离	
网络	源点个数	KST	KST-Improved
Power Grid	2	1.39	1.31
	3	1.67	1.687
	4	1.775	1.6
	5	1.893	1.741
Yeast	2	0.925	0.91
	3	0.98	0.936
	4	1.15	1.099
	5	1.226	1.158

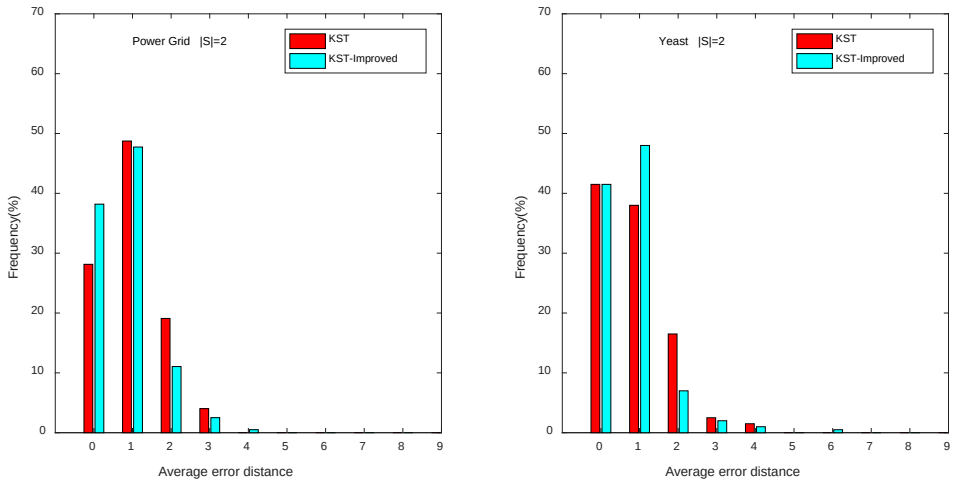


图3.8 源节点为 2 时 KST 与 KST-Improved 算法平均误差距离直方图

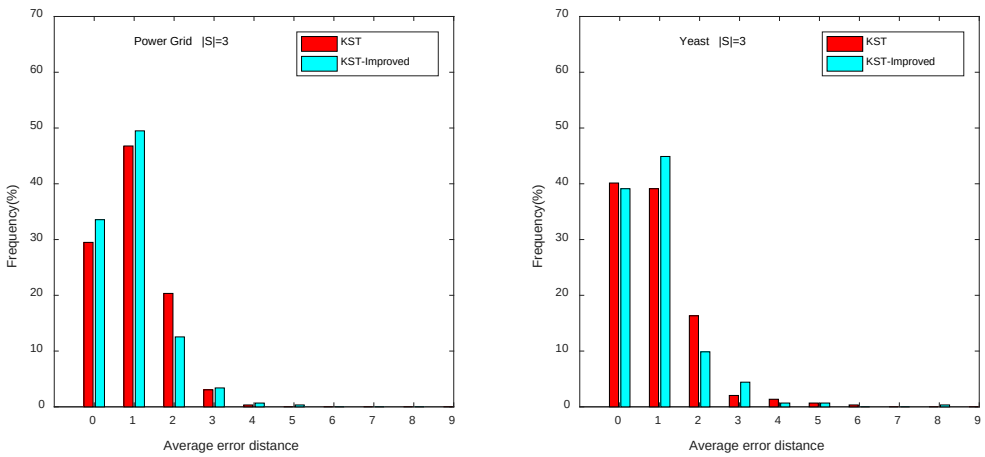


图3.9 源节点为 3 时 KST 与 KST-Improved 算法平均误差距离直方图

通过实验将 KST-Improved 算法与 KST 算法进行了对比，实验结果如表 3.3 所示。从表中可以看出，KST-Improved 算法稍优于 KST 算法，如在 Power Grid 网络上，当传播源个数为 2 时，KST 检测的平局误差距离为 1.39 而 KST-Improved 则为 1.31。同样，我们也给出检测结果的直方图以便对检测结果有更深入的了解。如图 3.8 所示，当传播源个数为 2 时，在 Power Grid 网络上，KST-Improved 方法有大约 40%的概率可以正确找到真实传播源，KST 方法则在 30%左右。图 3.9 是源节点为 3 时两种算法的检测结果，同样可以看出 KST-Improved 算法的检测结果稍优。

总的来看，KST-Improved 算法稍优于 KST 算法，证明了有效传播时间比基于最小路阶数的传播时间更精确，可以提高 KST 算法的检测准确度。

3.5 估计传播源个数

3.5.1 启发式算法

和目前存在的大多数多源节点检测算法类似，KST 算法需要提前知道传播源的个数，但在大多数实际应用中，一般不可能提前获得传播源个数。这里我们提出了一个简单的启发式算法来进行传播源个数的估计，算法伪代码参考算法 3.4 所示。

Algorithm 3.4：Estimate the Number of Diffusion Sources .

Input：An infection graph G_I.

Initialization：Initialize the number of sources $k=1$，$\upsilon^0=0, t^0=\infty$，$alpha^0=\infty$.

for (*k* start from 1 to an given maximum value *L*) **do**

Use **KST-Improved** to identify the *k* sources.

Estimate the spreading time t^k of the infection graph with *k* sources and calculate the variance υ^k.

Calculate the $alpha^k=\upsilon^k+t^k+k/w$ value, $w=0.2$ in the experiments.

if $alpha^k < alpha^{k-1}$ and $alpha^k < alpha^{k+1}$

stop.

end if

end for

Output：the estimate sources number *k*.

从 3.3.1 小节，我们知道 *k* 个传播源是同时开始向外传播的，也就是说，当我们在某一时刻对网络进行观察时，每个传播源向外传播的持续时间是相等的。据此，我们让估计源个数从 $k=1$ 开始依次递增，通过定义一个可以随估计传播源个数变化的

参数 $alpha^k$，以测量当传播源个数为 k 时，这 k 个分区传播时间之间的差异，我们希望该参数能在估计源个数等于真实源个数时取到最小值，即假设估计源个数等于真实传播源个数时，每个传播源向外传播的时间大抵相等。考虑到一般情况，当估计传播源个数小于真实传播源个数时，整个网络的传播时间加大，而估计源个数大于真实传播源个数时，整个网络的传播时间减小，最终将 $alpha$ 定义如下：

$$alpha^k = \upsilon^k + t^k + k / w \tag{3-16}$$

其中 υ^k 是 k 个分区传播时间的方差；t^k 是当估计源个数为 k 时整个网络的传播时间，即 $t^k = \max\{t_i^k, i = 1,...,k\}$；最后一项 k / w 是惩罚项，w 是一个常数，实验中取 $w = 0.2$。算法中用到传播时间使用前面提出的有效传播时间来估计。

3.5.2 实验与分析

(1) 实验设置

假设传播遵循 SI 模型且所有传播之间相互独立，网络连边上传播概率服从 (0,1) 之间的均匀分布，实验中设置真实传播源个数从 1 到 3，针对每种设置根据 SI 模型模拟传播过程，得到感染网络，然后在感染网络上运用提出的启发式算法估计真实传播源的个数。针对不同真实传播源个数，分别统计 100 次估计结果。

(2) 实验结果与分析

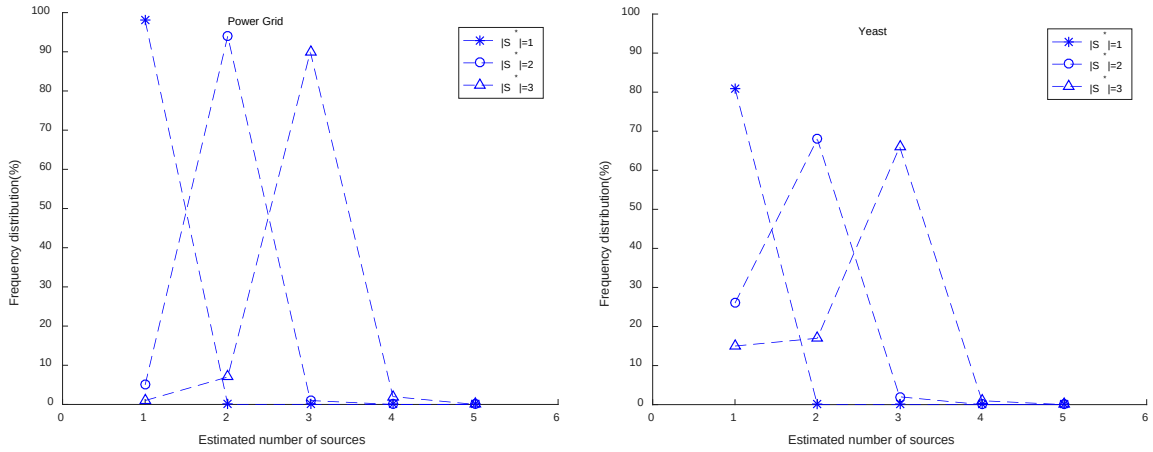


图3.10　估计传播源个数

实验结果如图 3.10 所示，水平轴表示算法所估计的传播源个数，垂直轴表示估计值出现的相应频率。从实验结果来看，在 Power Grid 网络上，我们提出的估计传播源个数的启发式算法有超过 80%的概率可以准确检测出对应传播源个数，有的甚至接近 100％。对于 Yeast 网络，我们可以看到，真实传播源个数从 1 到 3，我们的方法都有超过 60%的概率可以准确地估计出传播源个数，总的来看我们提出的源点个数估计算法有较高的估计准确度。

3.6 本章小结

本章研究了基于 SI 传播模型的多源节点检测问题。首先详细介绍了 SI 模型以及基于最小路阶数的传播时间估计方法和任意两点间传播概率的计算方式。然后从传播时间的角度给出了多源节点检测的目标函数，在此基础上提出 KST 算法以迭代的方式最小化该目标函数，从而实现多源节点的检测。通过在实际网络拓扑中的实验仿真，证明了 KST 算法可以得到相对更高的检测精度。

考虑到基于最小路阶数的传播时间估计方式没有考虑传播概率的问题，提出有效传播时间来估计网络中任意两点间传播时间。通过实验证明了提出的有效传播时间可以进一步优化 KST 算法的检测准确度。

最后，针对实际中传播源个数难以提前获知这一情况，提出了一种可以估计传播源个数的启发式算法，实验结果表明该算法能够以比较高的精度来进行传播源个数的估计。

第四章 基于 SIR 传播模型的多源节点检测算法

在源点检测的研究中，常用的传播模型有三种：SI 模型、SIS 模型和 SIR 模型。其中 SIS 和 SIR 模型都引入了恢复阶段，两者的区别在于个体恢复后所处状态不同：SIS 模型中感染个体在恢复之后重新变为易感染状态，其可以被再次感染；SIR 模型中感染者在恢复后具有了免疫能力，不会被再次感染。本章在第三章的基础之上研究了 SIR 传播模型下的多源节点检测问题，假设网络上的传播过程遵循 SIR 模型，且仅能实现对传播状态的一个局部观察，即仅能获得网络中节点的部分状态信息。针对以上情况，本章提出了 WP-KST 算法来实现 SIR 模型下的多源节点检测。实验结果表明 WP-KST 算法可以以较高的准确度解决 SIR 模型下的多源节点检测问题。

4.1 引言

随着社会的发展，各式各样的复杂网络为信息传播提供了一个充满活力的平台。目前学者们已经提出了不同类型的信息传播模型[66]来模拟网络中的信息传播过程，这些模型大致可以分为两类：影响扩散模型（如 IC 和 LT[67]模型）和流行病传播模型。其中在流行病传播模型中，网络中的节点都会处于以下几种状态之一：易感染状态(S)，感染状态(I)或恢复状态(R)。根据这些状态之间的转换，流行病传播模型可以进一步划分为 SI，SIR 和 SIS 模型。在源点检测研究中常用流行病传播模型来模拟信息在网络上的传播过程，不同的模型模拟不同的传播过程。其中 SIR 模型是用来模拟传播过程中包含免疫阶段的模型，比如很多计算机病毒可以通过杀毒软件进行清除，而像某些谣言个体在相信一段时间后会看清谣言的真相。而 SIS 模型则用来模拟一些包含恢复过程但不会获得免疫力的传播过程，例如我们对于伤寒、肺结核等细菌性疾病没有免疫力，感染个体在恢复后还是会被同样的疾病感染[35]。

现有的源点检测方法大都基于 SI 传播模型的，然而以现实中谣言的传播为例，收到谣言的人可能会以一定的概率转发谣言之外，他们也可能发现该消息是谣言，然后以一定的概率将其删除（恢复），显然基于 SI 模型的源点检测方法并没有考虑到这种情况。目前只有一少部分源点检测方法是基于 SIS 与 SIR 传播模型的，例如，例如 Saito[68]和 Luo 提出的 SIS 模型下的源点检测算法，Chen 与 Zhu[34]也研究了 SIR 模型下的源点检测问题。然而，这些方法大都针对的是单源节点检测问题，没有考虑多个传播源的情况。实际上，信息通常是从多个传播源开始向外传播的，比如，社交网络中的谣言经常是从多个传播者处开始传播。

假设网络上的传播过程遵循 SIR 模型，且仅能实现对传播状态的一个局部观察，

我们提出了 WP-KST 方法来实现 SIR 模型下的多源节点检测。首先提出了一个权值传播算法，用来解决 SIR 模型下不能正确区分恢复节点和易感染节点的问题。然后在由权值传播算法检测出来的扩展感染网络上，运用第三章中的 KST 算法实现多源节点的检测。实验结果显示我们提出的权值传播算法可以很好地检测出网络中的恢复节点，而 WP-KST 方法则可以以较高的准确度检测出感染网络中的多个传播源节点。

4.2 预备知识

这一节我们介绍了本章中要用到的一些预备知识，包括 SIR 传播模型以及局部观察方式。

4.2.1 SIR 传播模型

SIR(susceptible-infected-recovery)模型是常用的传染病传播模型，该模型考虑了个体的恢复过程，个体在恢复后具有了免疫能力，不会被再次感染，SIR 模型常用来模拟传播过程中包含免疫阶段的传播。

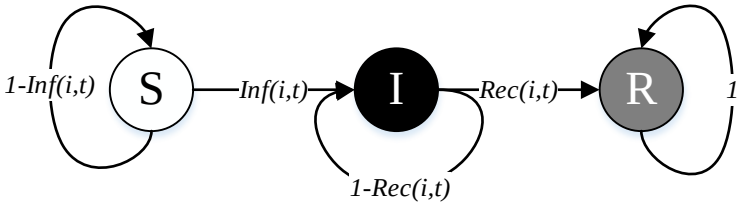


图4.1 SIR 模型节点状态转换图

如图 4.1 所示，在 SIR 模型中，网络中的每个节点有三种可能的状态：易感染状态(S)，感染状态(I)和恢复状态(R)。传播初始时刻，除了处于感染状态的传播源之外，网络中的其它节点都处于易感染状态。网络中的感染节点通过一定的概率将疾病传染给它的邻居节点，一旦一个 S 状态的节点被感染，则它成为 I 状态节点，此时它也成为了一个新的感染源，可以将疾病传播给它的邻居节点。而处于感染状态的节点可以以一定概率恢复，成为 R 状态。处于恢复状态的节点将获得永久免疫力，不会再次被感染。

假设传播在时间$T=0$时开始，我们使用$P_S(i,t)$、$P_I(i,t)$和$P_R(i,t)$分别表示节点v_i在时间为$T=t$时处于易感染状态、感染状态和恢复状态的概率。初始时刻也就是$T=0$时刻，所有的传播源节点都处于感染状态，它们对应的$P_S(i,0)=0$、$P_I(i,0)=1$，$P_R(i,t)=0$。同样，所有易感染节点都具有$P_S(i,0)=1$和$P_I(i,0)=0$，$P_R(i,t)=0$的初始状态。根据 SIR 模型，一个易感染节点转换为感染节点后不会再次成为易感染节点，以及一个感染节点成为恢复节点后不会再次成为易感染节点或感染节点，我们可以利用

式(4-1)、式(4-2)以及式(4-3)得到网络中每个节点在任意时刻处于各个状态的概率：

$$P_S(i,t) = [1 - Inf(i,t)] \cdot P_S(i,t-1) \tag{4-1}$$

$$P_I(i,t) = Inf(i,t) \cdot P_S(i,t-1) + (1 - Rec(i)) P_I(i,t-1) \tag{4-2}$$

$$P_R(i,t) = P_R(i,t-1) + Rec(i) \cdot P_I(i,t-1) \tag{4-3}$$

其中 $Inf(i,t)$ 表示节点 v_i 在 $T = t$ 时刻被其邻居节点所感染的概率。$Inf(i,t)$ 计算方式如下：

$$Inf(i,t) = 1 - \prod_{j \in N_i} [1 - \eta_{ji} \cdot P_I(j,t-1)] \tag{4-4}$$

上式中的 η_{ji} 表示从节点 v_j 到其相邻节点 v_i 的传播概率，N_i 表示节点 v_i 的邻居节点集合。式中的 $Rec(i)$ 表示节点 v_i 由感染状态成为恢复状态的概率。随着传播的进行，当某个易感染节点满足 $P_I(i,t) > P_S(i,t)$，即认为该节点被感染，之后在某一时刻会出现 $P_R(i,t) > P_I(i,t)$，这时则表明该节点已经成为恢复节点。

4.2.2 局部观察

我们在第二章第 2.2 小节介绍了源点检测问题的一个重要前提就是对网络进行观察，观察方式是我们了解传播过程、传播进展以及网络中节点状态信息的途径。不同的观察方式所获得的信息不同，也使得解决源点检测问题的算法思路有所不同。目前存在的观察方式可分为三种：完全观察、快照观察、传感器观察。其中完全观察可以获得所有节点的准确状态信息，但是在现实中不容易实现。于是就有了所谓的局部观察，局部观察也称为快照观察，是指在传播进行过程中的某一时刻对网络进行观察，且仅获得网络中节点的部分状态信息。

在本章中，假设信息的传播遵循 SIR 模型，在传播进行到某一时刻对网络进行观察，我们可以观察到网络中所有的感染节点，但是无法区分易感染节点和恢复节点。具体而言，对于 SIR 模型来说，我们只能观察到那些在观察时刻处于 I 状态的节点，而无法区分其余节点的真实状态是 S 还是 R。也就是说，当我们在传播进行过程中的某一时刻观察网络时，我们看到的网络中的节点要么是“感染的”节点，要么是“健康的”节点，至于健康的节点是否曾经被感染过我们无法获知。

给定一个原始网络，假设网络中只有一个传播源节点 S 且传播过程遵循 SIR 模型。图 4.2 给出了完全观察与局部观察两种方式下我们所能获得的节点状态信息。完全观察方式下，我们能获得网络中所有节点当前所处的准确状态信息，如节点 D、G、L 和 S 处于恢复状图，而节点 A、B、C、E、F、H、K、J、R、U 和 X 处于感染状态。现实中我们很难实现对网络的完全观察，大多数情况下仅能实现对网络的一个局部观察。在这种观察方式下，我们能获得感染状态的节点信息，其余节点对我们而言都是“健康的”节点，例如节点 D、G、L 和 S 这些节点真实所处状态为恢复状态，但是我们能观察到的是它们是处于易感染状态的，与那些一直处于易感染状态而未曾被感染

过的节点状态是一样的。显然，局部观察方式只能获得相对较少的节点状态信息，增加了源点检测的难度。

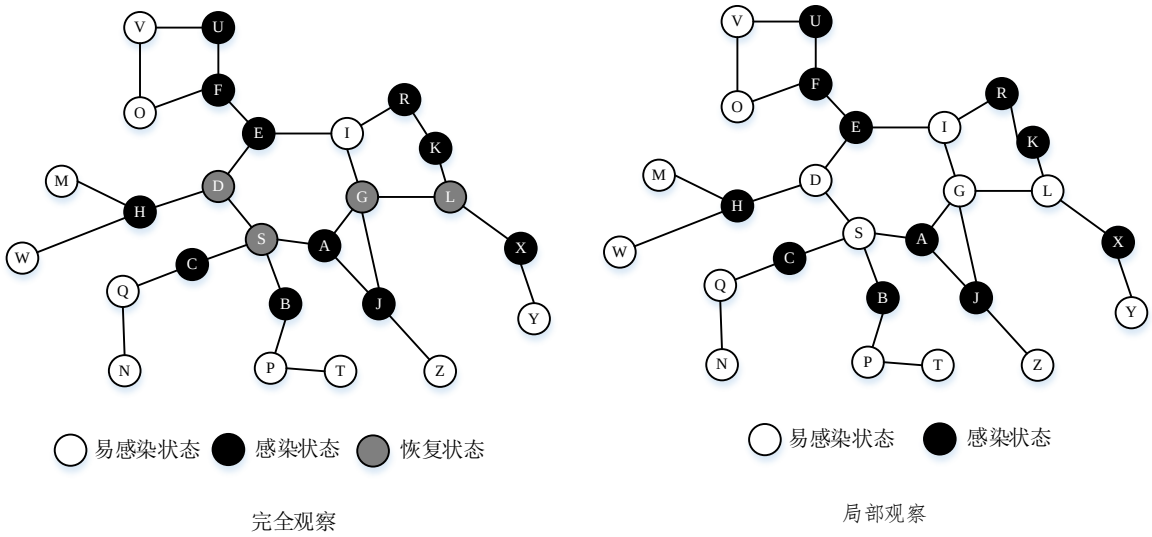


图4.2 SIR 模型下完全观察与局部观察示意图

4.3 WP-KST 算法

这一节详细介绍了局部观察方式下针对 SIR 模型提出的 WP-KST 多源节点检测算法。

WP-KST 算法在第三章提出的基于 SI 模型的多源节点检测算法 KST 的基础上加上一个权值传播算法，用以解决局部观察方式下不能准确区分恢复节点和易感染节点的问题。WP-KST 方法主要包含两步，第一步首先根据权值传播算法检测网络中的恢复节点，解决了局部观察下无法正确区分恢复节点和易感染节点的问题。之后将权值传播算法检测出来的恢复节点与观察到的感染节点提取出来组成一个新的网络，称为扩展感染网络。第二步在得到的扩展感染网络上运用 KST 算法实现多源节点的检测。

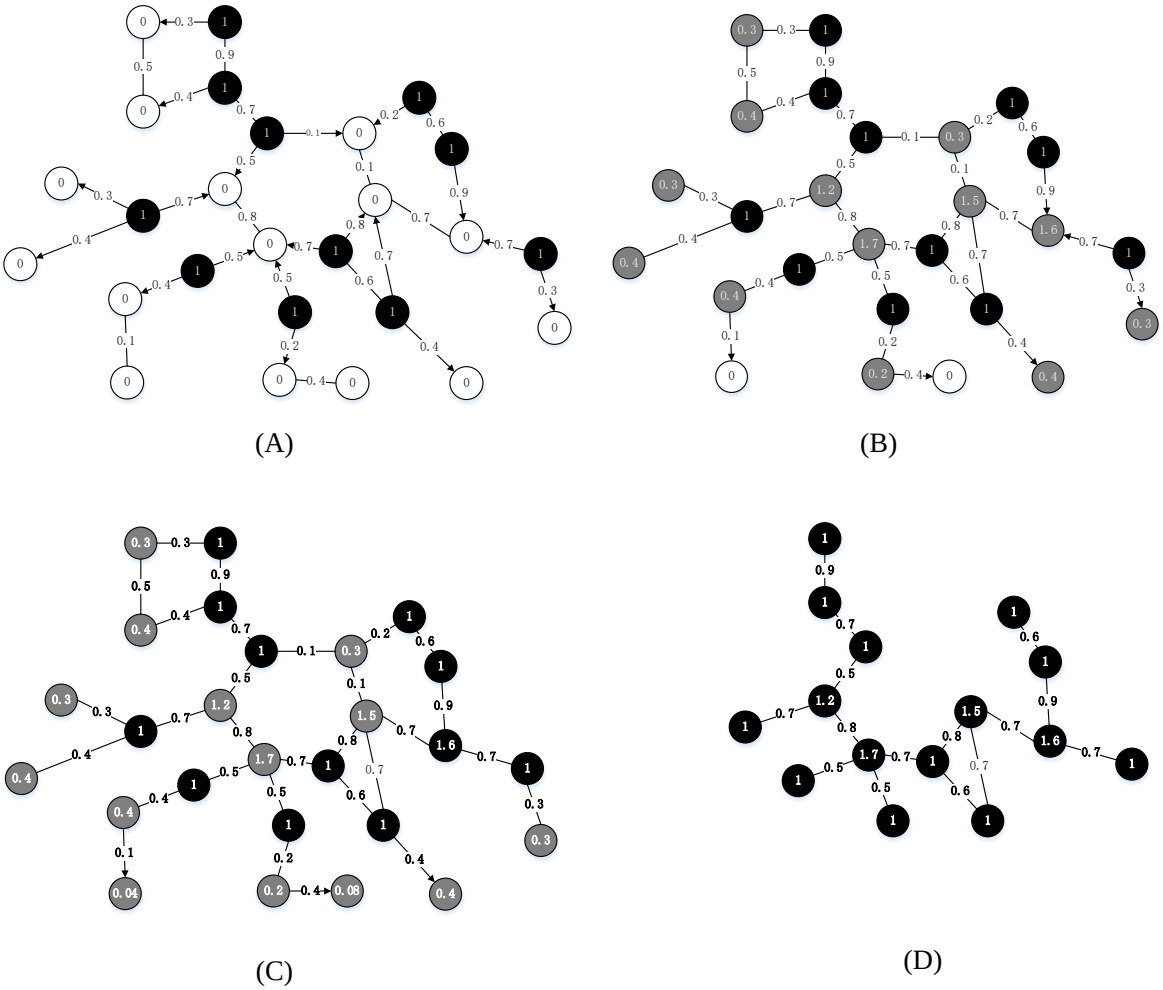
4.3.1 权值传播算法

基于 SIR 模型的源点检测问题中一个难点是恢复节点不能与易感染节点区分开来，针对此问题，我们提出一种权值传播算法(Weight Propagation Algorithm)。该算法根据观察到的感染网络 G_I 从易感染节点中推断出可能的恢复节点，完成缺失信息的填充，得到扩展感染网络 $G_{I\cup R}$。感染网络和扩展感染网络的定义如下：

感染网络：感染网络 G_I 是原始网络 G_n 的一个子网络，其由我们所能观察到的所有感染节点 V_I 以及这些感染节点间的连边 E_I 组成。

扩展感染网络：扩展感染网络 $G_{I\cup R}$ 是原始网络 G_n 的一个子网络，由网络中所有

的感染节点V_I与恢复节点V_R以及这些节点间的连边$E_{I\cup R}$组成。



(A) (B)

(C) (D)

图4.3 权值传播算法示意图

在算法中，我们首先给每个节点定义一个权值θ，这个权值用来表示一个节点处于恢复或感染状态的可能性，即其属于扩展感染网络的可能性，权值越高意味着该节点越有可能属于扩展感染网络。初始时，将所有观察到的感染节点的权值设为$\theta=1$，而其他节点的权值为$\theta=0$。然后，更新每个感染节点邻居的θ值，取其所有邻居节点的θ值与相应传播概率乘积之和，即$\theta(i)=\sum_{t\in Neighbors(i)}\theta(t)*\eta_{ti}$。最后将更新过$\theta$值的节点当作新的感染节点，更新它们邻居节点的$\theta$值。执行上述操作直到网络中所有节点的$\theta$值都更新过则算法结束，网络中每个节点的$\theta$值只更新一次。最终我们认为网络中$\theta$值大于 1 的节点要么是恢复节点要么是感染节点，这些节点与观察到的感染节点连同这些节点间的连边组成扩展感染网络。权值传播算法的详细过程参考算法 4.1。图 4.3 给出了一个具体示例用来说明权值传播算法的具体过程。

在图 4.3 中，我们用黑色节点表示观察到的感染节点，其权值为$\theta=1$，白色节点表示观察到的“健康的”节点，其权值为$\theta=0$，网络连边上的数字表示相邻两点间的

传播概率，箭头表示权值传播的方向。感染节点以连边上的传播概率向外传播它们的权值，其邻居节点在收到来自感染节点传播来的权值后，将自身权值更新为收到的所有权值之和。最终网络中每个节点的权值如图 4.3(C)所示，将网络中所有权值大于 1 的节点与观察到的感染节点连同这些节点间的连边提取出来组成扩展感染网络，如图 4.3(D)所示，可以看出检测出来的扩展感染网络与真实的扩展感染网络完全一致。

Algorithm 4.1：Weight Propagaiton Algorithm.

Input：A diffusion network G_n, infected node set V_I.

Initialization：Initialize nodes with unique labels C and unique score θ, $C(i),\theta(i)=\begin{cases}1 & i\in V_I\\ 0 & otherwise\end{cases}$，initialize the extended node set $G_{I\cup R}=V_I$，$I^{'}=V_I$.

for (iter start from 1 to n) **do**

for ($x\in I^{'}$) **do**

$temp=\varnothing$.

for ($i\in Neighbors(x)$) **do**

if ($C(i)==0$)

$\theta(i)=\sum_{t\in Neighbors(i)}\theta(t)*\eta_{ti}$.

$temp=temp\bigcup i$.

$C(i)=1$.

end if

end for

$I^{'}=I^{'}\bigcup temp$.

end for

end for

for ($i\in G_n$) **do**

if ($\theta(i)>1$)

$G_{I\cup R}=G_{I\cup R}\bigcup i$.

end if

end for

Output：An extended infected node set $G_{I\cup R}$.

该算法的主要依据：一般情况下，网络中的恢复节点通常被感染节点所包围，而易感节点通常位于感染节点的外围。也就是说，由感染节点将得分值向外传播，网络中恢复节点相比易感节点会接收到更多的得分。举个例子，假设网络中不存在循环结

构，即网络是树形结构网络，那么网络中任意一个叶子节点的所有得分值只能来自其父节点。这样，易感染节点的叶子节点也必须是易感染节点。由于易感染节点的初始得分为 0，感染节点的初始得分等于 1，因此所有易感染节点的得分都来自其唯一的父节点，所以得分 $\theta_{i\in V_S}<1$。而对于恢复节点来说，在大多数情况下，它都至少有一个感染节点作为叶子节点，这些节点的得分等于其父节点和子节点的总和，得分 $\theta_{i\in V_R}>1$。当然由于一般网络中存在环路，比树形网络复杂的多，但也存在网络中的恢复节点通常被感染节点所包围这一特性，我们在 4.4 节中通过实验验证了权值传播算法在一般网络结构上可以以较高的准确度恢复出扩展感染网络。

4.3.2 KST 算法

WP-KST 方法的第二步是在得到的扩展感染网络上运用 KST 算法来实现多源节点的检测，其中 KST 算法是我们在第三章提出的基于 SI 模型的多源节点检测算法。给定感染网络以及传播源的个数，KST 会以一个较高的准确度检测出网络中的多个传播源节点。在 SIR 模型下，通过权值传播算法得到的扩展网络中所有节点都是处于感染状态或恢复状态，表明这些节点都被感染过，这与 SI 模型下的感染网络类似，因此我们通过在扩展感染网络上运用 KST 算法即可实现局部观察下的多源节点检测问题。

4.4 实验与分析

4.4.1 权值传播算法检测准确度实验

这一节我们在三个真实网络上通过实验验证了权值传播算法检测恢复节点的准确性。

(1) 实验设置

我们分别在 Power Grid[35]网络和两个 protein-protein interaction network[69]网络上验证了权值传播算法的有效性，表 4.1 给出了这三个真实网络的基本统计参数，图 4.4 给出了三种网络的度分布图。

表4.1 网络统计参数

数据集	Power Grid	PPI-1	PPI-2
节点数	4941	1297	2307
连边数	13188	3724	7914
平均度	2.67	2.87	3.43
最大度	19	29	89

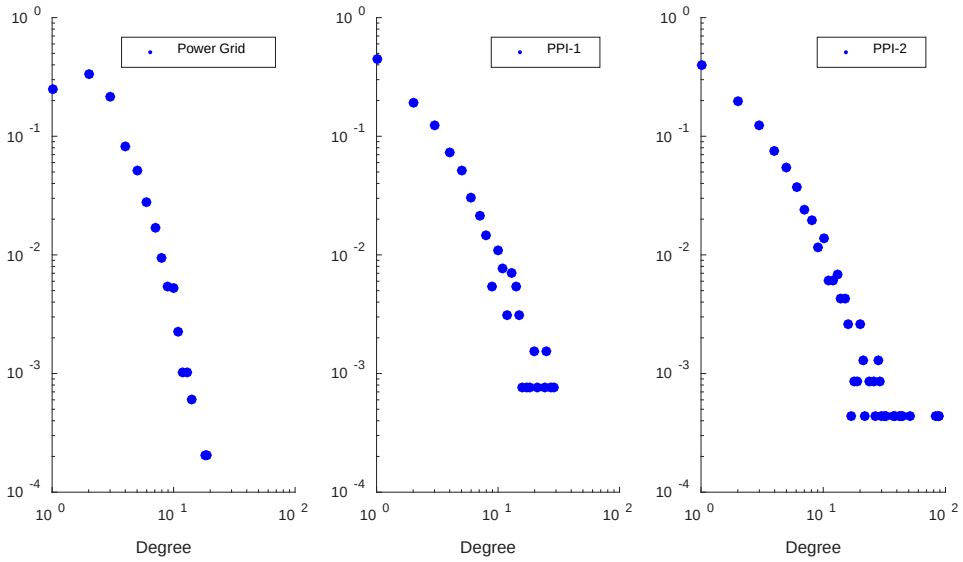


图4.4 网络度分布图

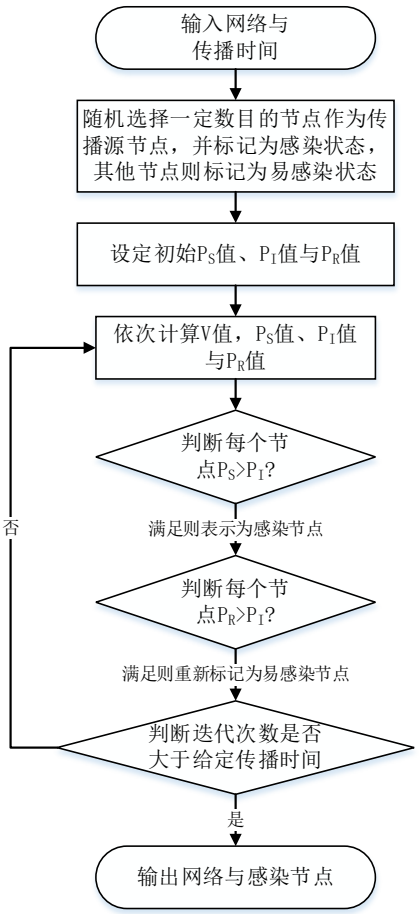


图4.5 SIR 模型模拟传播流程图

假设网络上的传播之间相互独立，同时设置网络上每条连边的传播概率服从(0,1)之间的均匀分布，每个节点的恢复概率服从(0,0.3)之间的均匀分布。我们设定传播源节点的个数由 2 到 5，并且针对每种设置分别进行 100 次的模拟实验，最终的

结果为这 100 次计算结果的平均值。对于每次实验，先从网络中随机选择给定数目的节点作为传播的初始源节点，然后利用 SIR 模型来模拟传播，模拟传播的具体过程如图 4.5。在传播过程持续一段时间后，根据已知的网络结构信息和感染节点状态信息运用权值传播算法检测网络中的恢复节点。

(2) 评价指标

我们采用机器学习领域中常用的几个评价指标来度量权值传播算法检测出来的扩展感染网络与真实扩展感染网络之间的差距，以此来反映权值传播算法恢复扩展感染网络的性能。在机器学习、数据挖掘领域中，常常会用一些指标来评判算法的优劣，其中最常用的就是准确率($Accuracy$)，简单来说就是

$$Accuracy = N_{pre} / N_{total} \tag{4-5}$$

上式中， N_{pre} 表示准确预测的样本数， N_{total} 表示测试集总的样本数。准确率有的时候过于简单，不能全面反应算法的性能，除了准确率，还有一些常用的指标，包括 F1 分数($F1-score$)， 召回率($Recall$)和精确率($Precision$)。

首先来看一个二分类问题，当我们将预测结果与实际结果比较时，会出现以下四种可能的情况：

表4.2 混淆矩阵

		预测		合计
		T	F	
实际	P	True Positive (TP)	False Negative (FN)	Actual Positive(TP+FN)
	N	False Positive (FP)	True Negative (TN)	Actual Negative(FP+TN)
合计		Predicted Positive (TP+FP=P）	Predicted Negative （FN+TN=N）	TP+FP+FN+TN

第一种：事实为正例，也被判定为正例：True Positive(TP)，正确划分为正类;

第二种：事实为正例，却被判定为负例：False Negative(FN)，错误划分为负类;

第三种：事实为负例，也被判定为负例：True Negative(TN)，正确划分为负类;

第四种：事实为负例，却被判定为正例：False Positive(FP)，错误划分为正类;

根据这四种情况，我们可以定义：

$$Precision = \frac{TP}{TP + FP} \tag{4-6}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{4-7}$$

$$F1-score = \frac{2TP}{2TP+FN+FP} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4-8}$$

可以看到， $Precision$ 就是预测正确的正例数占预测为正例数据的比例，体现了模型

对负样本的区分能力， *Precision* 越高，说明预测算法对负样本的区分能力越强； *Recall* 就是预测为正例的数据占实际为正例数据的比例，体现了分类模型对正样本的识别能力， *Recall* 越高，说明对正样本的识别能力越强； $F1-score$ 则是精确率和召回率的调和平均值，可以看作是准确率和召回率的一种权衡，它的最大值是 1，表示模型是完美的模型，最小值是 0，表示模型是完全相反的模型， $F1-score$ 越高说明预测算法越好，分类越准确。 $F1-score$ 的值同时受到准确率、召回率的影响，单纯地追求准确率或召回率的提升没有太大意义。

(3) 实验结果与分析

表 4.3 给出了 SIR 模型下权值传播算法在 Power Grid 以及两个 PPI 网络上的检测结果，其中 *Precision* 值表示由权值传播算法检测出来的扩展感染网络中属于真实扩展感染网络的节点个数与检测出来的扩展感染网络的总节点数的比值，相应的， *Recall* 值表示由权值传播算法检测出来的扩展感染网络中属于真实扩展感染网络的节点数与真实扩展感染网络的总节点数的比值。 $F1-score$ 则是这两个比值的调和平均值，反映算法的综合性能，我们主要以 $F1-score$ 为主要评价指标。

表4.3 权值传播算法检测结果

实验设置		*Precision*	*Recall*	$F1-score$
网络	源点个数			
Power Grid	K=2	0.9429	0.8240	0.8743
	K=3	0.9366	0.8614	0.8951
	K=4	0.9279	0.8478	0.8837
	K=5	0.9328	0.8256	0.8732
PPI-1	K=2	0.8672	0.8356	0.8432
	K=3	0.8611	0.8516	0.8526
	K=4	0.8552	0.8600	0.8552
	K=5	0.8614	0.8636	0.8606
PPI-2	K=2	0.6452	0.9233	0.7463
	K=3	0.7131	0.9312	0.7987
	K=4	0.7575	0.9366	0.8302
	K=5	0.7992	0.9314	0.8564

从表 4.3 来看，权值传播算法在三种网络上都可以取得很好的检测结果。以 Power Gird 网络为例，源点个数从 2 到 5，算法的 *Precision* 值都在 0.9 以上，说明检测出来

的扩展网络中由 90%以上的节点是属于真实扩展网络的； *Recall* 值也都在 0.8 以上，说明算法能将真实扩展网络中 80%以上的节点检测出来；$F1-score$ 值也都在 0.85 以上，说明算法具有很好的稳健性。

我们将这三个网络的结果进行对比发现，Power Grid 网络的检测结果普遍优于 PPI 网络，而且 Power Grid 网络中都是 *Precision* 值大于 *Recall* 值，PPI-1 中 *Precision* 值基本等于 *Recall* 值，PPI-2 网络却是 *Precision* 值小于 *Recall* 值。分析其原因是网络本身复杂性造成的。根据前面三个网络的统计参数可以看出 Power Grid 网络的平均度最小，下来是 PPI-1 网络，平均度最大的是 PPI-2 网络。网络的平均度越大说明该网络连接越紧密，网络中更容易出现环路。对于这样的网络，权值传播算法会将更多的节点判断为恢复节点，使得检测出来的扩展感染网络远远大于真实扩展感染网络，从而使得 *Recall* 值增大，而 *Precision* 减小。

4.4.2 WP-KST 算法检测准确度实验

在这一节，我们验证了 WP-KST 方法在局部观察方式下实现多源节点检测的有效性。

(1) 实验设置

实验设置与 4.4.1 节中的实验设置一致，假设网络上的传播之间相互独立，网络上每条连边的传播概率服从 $(0,1)$ 之间的均匀分布，每个节点的恢复概率服从 $(0,0.3)$ 之间的均匀分布，传播源节点的个数由 2 到 5，并且针对每种设置分别进行 100 次的模拟实验，最终的结果为这 100 次计算结果的平均值。对于每次实验，首先从网络中随机选择给定数目的节点作为传播的初始源节点，然后利用 SIR 模型来模拟传播。在传播过程持续一段时间后，利用 WP-KST 方法在网络上实现多源节点的检测。

(2) 对比算法

1. Zang 等人[70]提出以分治的思想来解决 SIR 模型下的多源节点检测问题，首先根据网络的社区结构将网络划分为 k 个社区，然后在每个社区中寻找具有最大无偏介数作为传播源节点。其中网络中任意节点 v_i 的无偏介数定义如下：

$$B_i^* = \frac{(B_i)^\alpha}{k_i} \tag{4-9}$$

上式中 B_i 指的是节点 v_i 的介数。网络中任意节点 v_i 的介数指的是网络中任意两点间最短路径中通过了节点 v_i 的最短路径数与总的最短路径数之比，其定义如下：

$$B_i = \sum_{s\neq t, s\neq i, t\neq i} \frac{n_{st}^i}{n_{st}} \tag{4-10}$$

2. Comin 等人[39]证明中心性度量指标可以用来检测传播源节点，其中常用的有接近中心性指标。接近中心性指标是用来评价网络一个节点到其他所有节点距离的大

小，它认为如果网络中一个节点与网络中其他所有节点的距离都很短，则该节点就是网络的中心。接近中心性指标的定义为一个节点 v_i 到网络中其他节点的最短路径平均值的倒数，具体公式如下：

$$C_i = \frac{1}{\frac{1}{n-1}\sum_{j,j\neq i} hops(i,j)} \tag{4-11}$$

其中 $hops(i,j)$ 指的是节点 v_i 与节点 v_j 间的最短路径长度。

(3) 实验结果与分析

表 4.4 中统计的是三个算法的平均误差距离，其中 WP-KST 指我们提出的算法，DC(Divide and conquer)则指 Zang 等人提出的分治算法，CC(Closeness centrality)则指接近中心性指标。平均误差距离是源点检测问题中最常用的评价指标，表示所检测出的源节点与真实源节点之间平均相距距离的大小，值越小表明算法找到的源节点更接近真实源节点，具体介绍可参考论文第三章第 3.3.4 小节。

表4.4 三种算法平均误差距离统计结果

实验设置		平均误差距离		
网络	源点个数	WP-KST	DC	CC
Power Grid	2	1.5570	4.0333	4.5800
	3	1.7255	4.6923	5.6296
	4	1.5115	5.0515	6.3636
	5	1.5991	5.4353	7.3200
PPI-1	2	2.0000	4.0471	3.5805
	3	2.3368	4.3065	3.8889
	4	2.1184	4.3988	4.0417
	5	2.4619	4.3952	3.9663
PPI-2	2	2.6500	4.5707	3.1818
	3	2.6867	4.4916	2.9495
	4	2.8707	4.6641	2.9621
	5	2.9160	4.6460	2.9800

表 4.4 给出了三种算法分别在 Power Grid 网络和两个 PPI 网络上的多源节点检测结果，可以看出 WP-KST 算法与其他算法相比优势明显，即 WP-KST 算法找到的源节点更为接近真实源节点。以 PPI-1 网络为例，当传播源个数为 2 时，WP-KST 算法的平均误差距离为 2，说明其找的源节点距离真实源节点平均为 2-3 个路阶数，相应的 CC 算法的平均误差距离则为 3-4 个路阶数，DC 算法的平均误差距离最大，为 4-5

个路阶数。图 4.6 与图 4.7 分别给出了传播源个数为 2 和 3 时，三种算法检的测结果直方图。以 Power Gird 网络为例，当传播源为 2 时，WP-KST 算法大约有 50%的概率检测出来的传播源节点距离真实源节点在 1 个路阶数内，且其中有 15%左右的概率可以完全准确地检测出传播源节点。相对来说虽然 DC 算法只有不到 20%的概率检测结果在 1 个路阶数内，而 CC 检测结果就更差了。总的来说，WP-KST 算法的检测出的源节点更为接近真实传播源节点，检测结果优于 DC 和 CC 这两个对比算法。

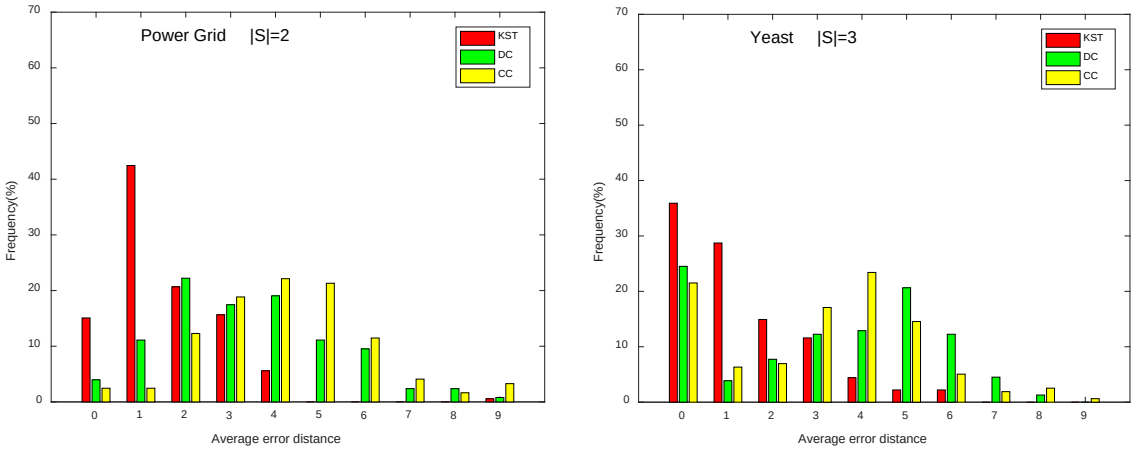


图4.6 源节点为 2 时三种算法平均误差距离直方图

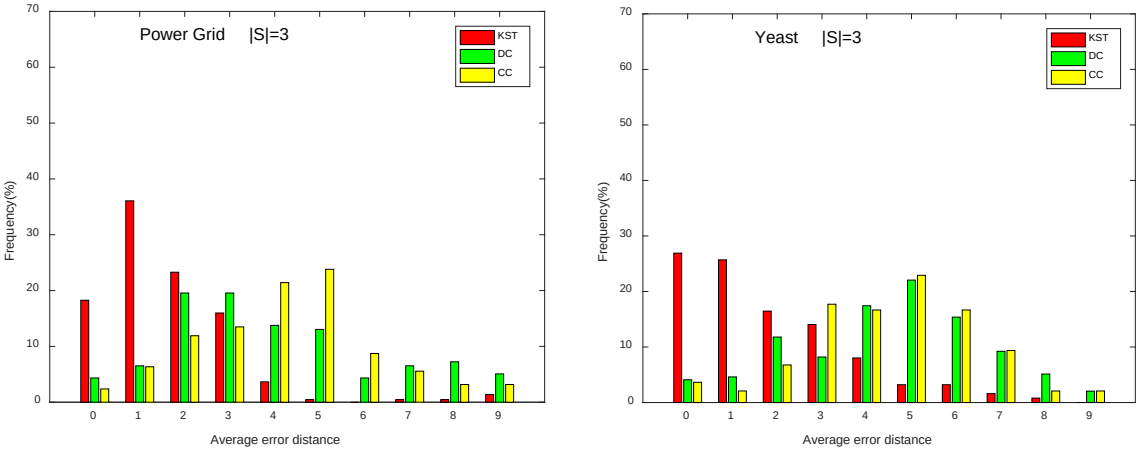


图4.7 源节点为 3 时三种算法平均误差距离直方图

DC 和 CC 算法都是通过中心性指标来确定传播源节点，其中 CC 算法直接选取感染网络中具有最大接近中心性的 k 个节点作为传播源节点，然而在多源检测问题中传播源是分散的，并不是聚集在感染网络的中心。DC 方式虽然在进行源点检测之前先进行了网络的划分，使得每个社区中包含一个传播源节点，但传播过程的复杂性使得以无偏介数来定位源节点显得并不太理想。相对来说，WP-KST 算法在进行源点检测之前先进行了恢复节点的检测，得到更为接近真实情况的扩展感染网络，为之后的多源检测过程做好了前提工作，再加上 KST 算法本身具有较高的检测准确度，使得

我们的 WP-KST 算法可以取得相对较好的结果。

4.5 本章小结

本章研究了基于 SIR 传播模型的多源节点检测问题。首先详细介绍了 SIR 传播模型，同时说明了在 SIR 模型下存在不能正确区分恢复节点和易感染节点的问题。针对该问题，提出了一种权值传播算法，通过将网络中的感染节点的初始权值向外传播，网络中的恢复节点会收到相对更高的权值，从而现实恢复节点的检测。仿真实验证明权值传播算法可以很好的检测出网络中恢复节点，完成缺失信息的填充。然后在得到的由感染节点与恢复节点以及这些节点间连边组成的扩展感染网络上，运用 KST 算法进行多源节点的检测。实验结果表明 WP-KST 算法可以很好的解决 SIR 模型下的多源节点检测问题，且具有较高的检测准确度。

第五章 基于传感器观察的源节点检测方法

在现实情况中，由于各种各样的原因，很难获取到传播所在网络中每一个节点的状态信息。常用的解决方法是提前监控网络中的一些节点或者随机查询网络中的一些节点，利用这些节点提供的信息实现传播源节点的检测，这些节点被称为传感器节点，而这种比较符合实际情况的先验知识获取方式则称为传感器观察。本章研究了传感器观察方式下的源节点检测问题，假设传播遵循 SI 模型，提出了一种单源节点检测 RDPC 算法，实验结果验证了 RDPC 算法具有较高的检测准确度。此外，我们通过一个简单的划分思路将 RDPC 算法扩展到了多源节点检测问题上，通过实验验证了扩展后的 RDPC 算法可以很好的解决传感器观察方式下的多源节点检测问题。

5.1 引言

随着大数据时代的到来，海量信息正在通过各种大规模网络迅速传播。从实际和技术两个方面来看，大规模真实网络上实现传播源节点检测意义重大。目前，研究人员已经提出了一些基于真实网络拓扑的源点检测方法[29, 31, 71]，但这些方法都需要对网络中的每个节点进行测试以此来判断其是否是真实传播源节点，计算量很大，不适用于大规模真实网络。而且考虑到现实情况中，一般不可能实现对每个传播参与者的查询，且由于个人隐私等问题，也不可能从每个传播参与者处获得其自身关于传播的相关信息。针对以上问题，常用的解决方法是通过在网络中选择一定数量的节点作为传感器，利用从这些传感器处获得的信息进行源节点的检测。所谓传感器其实质也是网络中用户或计算机等，它们和网络中普通节点的区别在于可以记录发生在各自身上所有的传播细节，包括它们的状态、状态转换时间以及感染方向等。通过这种方式，只需要知道部分传播参与者的相关信息即可实现传播源节点检测，非常适用于大型网络中传播源节点的检测。

目前存在的基于传感器观察的研究大都针对的是单源节点的检测[32, 44]。它们假设传播在开始时刻只有一个传播源节点，且大都采用极大似然估计的方法来确定传播源节点[32, 44, 72]。基于极大似然估计的方法一般复杂度都比较高，不太适合现实中的大规模网络。而且，现实中传播源一般不会只有一个，从多个传播源发散出来的信息会相互影响，使得多源传播过程更为不确定，加大了多源节点检测的难度。目前针对多源检测的算法很少，而基于传感器观察的多源节点检测方法几乎没有。

针对以上问题，本章首先提出了一种基于传感器观察的单源节点检测算法，RDPC 算法。RDPC 算法首先利用反向传播算法[73]从网络中选出所有可能的传播源节点，针

对这些可能源节点，依次判断其到感染传感器之间的传播时间与传感器记录的真实感染时间的线性相关性，选择具有最大线性相关性的节点作为传播源节点。实验结果验证了 RDPC 算法具有较高的检测准确度且耗时较短。然后，针对传播初始时刻传播源不止一个的问题，提出一种简单的划分思想将 RDPC 算法扩展到了多源检测问题上，解决了传感器观察下的多源节点检测问题。

5.2 传感器观察下的源点检测问题描述

给定一个网络 G_n，先提前在网络中选择一定数量的节点作为传感器，记为 $O=\{o_1,o_2,...o_m\}$，传感器的选择方式与具体数量 m 根据具体情况决定。假设网络上的传播行为遵循 SI 模型，在某一时刻传播从一个或多个节点爆发并开始向外传播，在传播进行一段时间后，网络中的部分节点会被传播所感染从而由易感染状态变为感染状态，其中也包括部分传感器节点，我们称这些被感染的传感器为感染传感器。在这样的网络中，我们所能获取的关于传播的信息则是由这些感染传感器提供的它们各自的感染时刻。所谓感染时刻就是传感器节点在前一个时刻还处于易感染状态而在这一时刻被感染成为感染状态。假设网络中有 l 个传感器被感染，记为 $O_I=\{o_1,o_2,...o_l\}$，用 $T=\{t_1,t_2,...,t_l\}$ 来表示感染传感器记录的其自身的感染时刻，而未感染传感器则记为 O_S。然而，由于我们并不知道传播开始的时刻，所以传感器的感染时刻 $T=\{t_1,t_2,...,t_l\}$ 并没有直接的利用价值，不过我们可以根据感染时刻得到传感器的相对感染时刻，用 $D=\{d_1,d_2,...,d_l\}$ 来表示，其中 $d_j=\max(T)-t_j$。

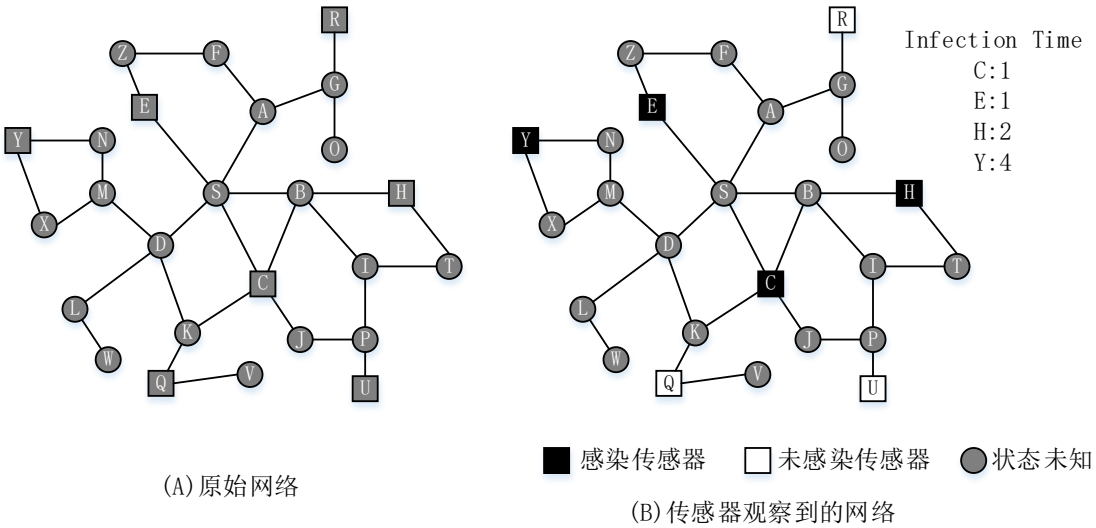


图5.1 SI 模型下传感器观察示意图

图 5.1 是传感器观察下传播情况的一个具体示例。图 5.1(A)是原始网络，其中的方格表示我们设置的传感器节点，即 $O=\{C,E,H,Q,R,U,Y\}$。在某一时刻爆发的传播

在经过一段时间的传播后，形成如图 5.1(B)所示的感染网络，其中未感染传感器集合为$O_S=\{Q,R,U\}$，感染传感器集合为$O_I=\{C,E,H,Y\}$，相应的感染时刻为$T=\{1,1,2,4\}$。根据前面的介绍，感染传感器集合$O_I=\{C,E,H,Y\}$的相关感染时刻则为$D=\{3,3,2,0\}$。传感器观察下的源点检测就是根据已知的网络结构、传感器节点的状态信息以及感染传感器的感染时刻信息找出传播源节点。

5.3 传感器观察下的单源节点检测算法

在这一节，我们详细介绍了提出的传感器观察方式下的单源节点检测算法，RDPC(Reverse Diffusion Pearson Correlation Algotirhm)算法。该算法主要分为两步：首先利用 Wang 等人[72]提出的反向传播算法从网络中选出可能的传播源节点。然后针对这些可能的传播源节点，依次测试其到每个感染传感器之间的传播时间与传感器真实记录的传播时间的线性相关性，选择具有最大线性相关性的节点为传播源节点。

5.3.1 反向传播算法

在传感器观察方式下，我们仅能获得传感器收集到的信息，这些信息一般包括它们的状态以及状态转换时间等。也就是说，网络中的传感器可以反映它们自身所处的状态，是感染状态还是易感染状态，如果是被感染状态，还会反映它们各自被感染的具体时刻。然而由于我们并不知道传播开始的具体时刻，所有被感染传感器所记录的其自身被感染的时刻是没有直接意义的。但从这些感染时刻，我们可以得到感染传感器的相对感染时刻。也就是说可以根据相对感染时刻知道哪个传感器是先被感染的，而哪个传感器又是后被感染的。反向传播算法就是基于这样的思路来搜索可能传播源节点的。一个节点只有满足了以它开始的传播在经过若干时间的传播后，能够让这些感染传感器形成观察到的相对感染时刻，它才有可能是传播源节点。

反向传播算法(Reverse Dissemination Algorithm)由 Wang 等人[72]提出。算法中首先给每个感染传感器一个标签，这个标签用来表示感染传感器本身，一般取感染传感器的 ID。然后，根据感染传感器记录的自身被感染时刻，让那些最后感染的节点先开始传播，接着是次最后感染的节点开始传播，以此为顺序将感染传感器的标签向外传播。因为传播的是感染传感器的标签，所以传播过程中设置连边上的传播概率为 1。整个过程持续一段时间后，选取网络中那些可以同时收到所有感染传感器标签的节点作为可能源节点。算法的伪代码参考算法 5.1。

在图 5.2 中，我们给出一个简单的例子来说明反向传播算法的具体过程。网络中共有四个感染传感器$O_I=\{C,E,H,Y\}$，它们记录着自身的感染时刻，其中传感器 C、E 的感染时刻为 1，传感器 H 的感染时刻为 2，传感器 Y 的感染时刻为 4。根据前面

的介绍，由相对感染时刻的计算方法可得传感器 C、E 的相对感染时刻为 3，传感器 H 的相对感染时刻为 2，传感器 Y 的相对感染时刻为 0。根据反向传播算法，首先应该给感染传感器设置标签，这里我们取每个传感器的 ID 作为其标签。然后以相对感染时刻为顺序开始向外传播标签，最先开始相对感染时刻为 0 的传感器 Y，它将标签‘Y’传递给节点 N、X。第二次向外传播时，由于没有相对感染时刻为 1 的传感器，所以只让上一步收到标签的节点继续向外传播所收到标签，即收到标签‘Y’的节点 N、X 继续将标签‘Y’向外传播，使得节点 M 也收到标签‘Y’。第三次向外传播时，相对感染时刻为 2 的传感器 H 开始向外传播，节点 B 和节点 T 收到标签‘H’。同时上一步收到标签的节点 M 将标签‘Y’传递给节点 D。第四次向外传播时，相对感染时刻为 3 的传感器 C、E 开始向外传播，节点 S 和节点 Z 收到标签‘E’，节点 B、J、K、S 收到标签‘C’。同时上一步收到标签的节点 D 将标签‘Y’传递给节点 K、L、S；上一步收到标签‘H’的节点 B 将标签‘H’传感节点 S、I，节点 T 将收到的标签‘H’传给节点 I。此时，网络收到节点收到标签的具体情况如图 5.2(E)所示，所有节点中只有节点 S 同时收到了来自四个感染传感器的标签，即节点 S 最有可能是传播源点。

Algorithm 5.1：Reverse Dissemination Algorithm.

Input：An network G_n, the infect sensors $S_I = \{s_1, s_2, s_3, ..., s_m\}$, the infection time $T = \{t_1, t_2, t_3, ..., t_m\}$ and a given maximum value L.

Initialization：Calculate the $D = \{d_1, d_2, d_3, ..., d_m\}$ where $d_j = \max(T) - t_j$, initialize the label of infect sensor is their own id, initialize $I^{'} = \varnothing$,$U = \varnothing$.

for (k start from 0 to L) **do**

$I = I^{'}$

Find these infect sensors ω who can satisfy $K == d_i$, reset $I = I \bigcup \omega$ and $I^{'} = \varnothing$.

for (each node i in I) **do**

propagate the label of node i to its neighbors with propagation probability 1.

$I^{'} = I^{'} \bigcup N_i$ where N_i is the neighbors of node i.

end for

for (i from 1 to n)

if (node i receive all labels from S_I)

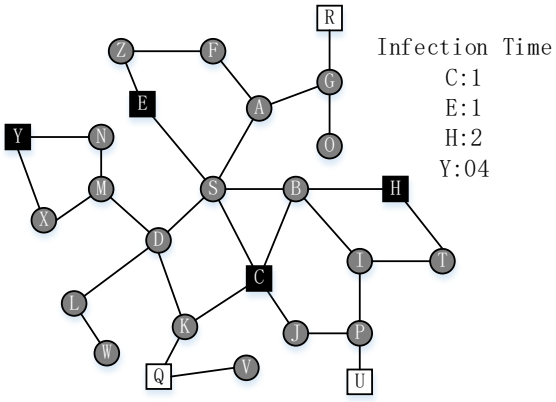
$U = U \bigcup i$.

end if

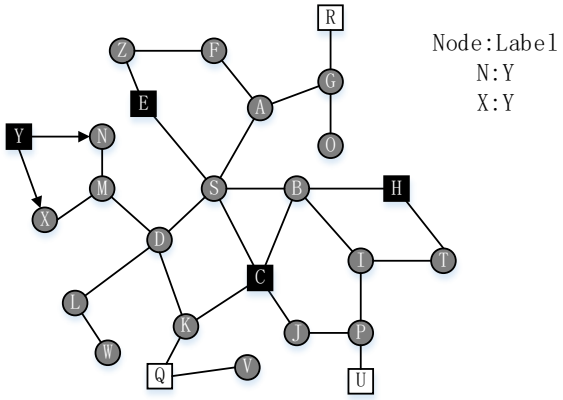
end for

end for

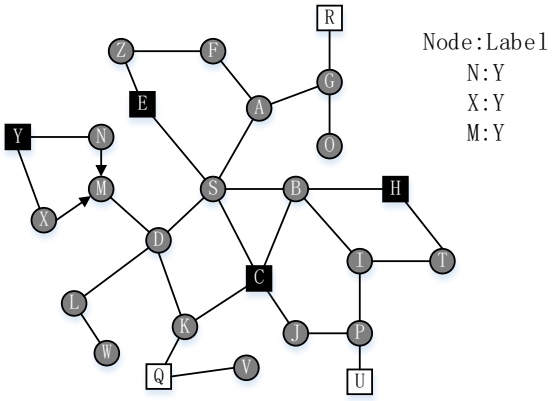
Output：A potential propagation sources set U.



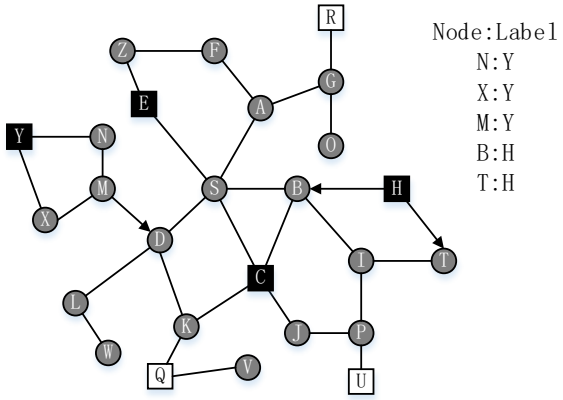
(A)反向传播算法开始时刻



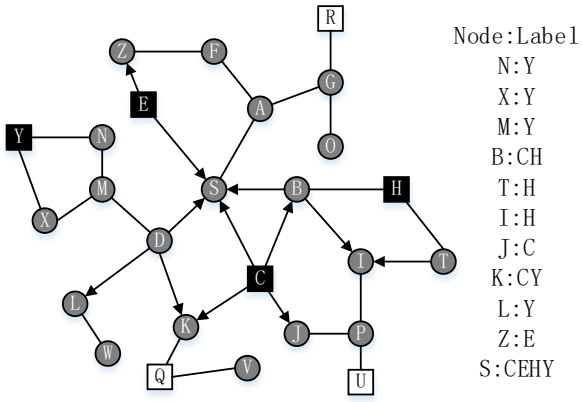
(B)反向传播算法第一次向外传播



(C)反向传播算法第二次向外传播



(D)反向传播算法第三次向外传播



(E)反向传播算法第四次向外传播

图5.2 反向传播算法示意图

5.3.2 线性相关性检测算法

由于现实网络的拓扑结构的复杂性以及传播过程的不确定性，使得反向传播算法并不能唯一地确定传播源节点。一般情况下，利用反向传播算法可以找出满足上述条件的一个节点集合，集合中的节点都有可能是真实传播源节点。为了确定唯一的传播

源节点我们提出用线性相关性来做进一步判断。

根据前面的介绍，我们所能获取的信息包括网络的拓扑结构以及网络中每个感染传感器记录的其自身感染时刻。假设我们可以估计出传播从网络中某个节点到所有感染传感器所需的时间，那么对于真实传播源节点来说，以它开始的传播到达感染传感器的时间应该和这些传感器自身记录的时间有着很高的相似度，我们用线性相关性来表示这种关系，即以某点开始的传播到所有感染传感器所需的传播时间与所有感染传感器记录的感染时刻间的线性相关性最大，则该节点最有可能为传播源节点。然而，根据前面的介绍，我们并不知道传播开始的具体时间，所有感染传感器所记录的感染时刻并没有直接意义，所以这里我们依然采用相对感染时刻。

算法的具体过程是在得到由反向传播算法检测出来的可能源节点集合后，将可能源节点集合作为输入，针对集合中的每个可能源节点，测试其到所有感染传感器的传播时间和感染传感器相对感染时刻的线性相关性。算法伪代码如算法 5.2。

Algorithm 5.2：Pearson Correlation Algorithm.

Input：An network G_n, the infect sensors $S_I=\{s_1,s_2,s_3,...,s_m\}$, the infection time $T=\{t_1,t_2,t_3,...,t_m\}$ and a potential propagation sources set *U*.

Initialization：Calculate the $D=\{d_1,d_2,d_3,...,d_m\}$ where $d_j=\max(T)-t_j$, $\varepsilon^*=-\infty$.

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

Estimate the spreading time from u_i to S_I, denoted as $X=\{x_1,x_2,x_3,...,x_m\}$.

Compute the correlation coefficient between X and D,

$$\gamma=\frac{\sum_{j=1}^{k}(x_j-\overline{x})(d_j-\overline{d})}{\sqrt{\sum_{j=1}^{k}(x_j-\overline{x})^2}\sqrt{\sum_{j=1}^{k}(d_j-\overline{d})^2}}$$

if ($\varepsilon>\varepsilon^*$)

Set $\varepsilon^*=\varepsilon$, and $s^*=u_i$.

end if

end for

Output：The estimated optimal propagation source s^*.

算法中用到的衡量两个变量相关性的参数 r 为皮尔逊相关系数（Pearson correlation coefficient），它是统计学中用于度量两个变量 *X* 和 *Y* 之间线性相关性的指标，其值介于-1 与 1 之间，定义如下：

$$r=\frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)\left(Y_i-\overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i-\overline{Y}\right)^2}} \tag{5-1}$$

其中 X_i 指变量 X 的第 i 个分量，$\overline{X}$ 为变量 X 的均值。

皮尔逊相关系数的绝对值越大，则变量 X 与 Y 相关度越高。当 X 与 Y 线性相关时，相关系数取值为 1（正线性相关）或-1（负线性相关）。具体的，如果有两个变量 X、Y，最终计算出的相关系数的含义可以有如下理解：

1. 当相关系数为 0 时，X 和 Y 两变量无关系。
2. 当 X 的值增大（减小），Y 值增大（减小），两个变量为正相关，相关系数在 0.00 与 1.00 之间。
3. 当 X 的值增大（减小），Y 值减小（增大），两个变量为负相关，相关系数在-1.00 与 0.00 之间。

在线性相关性检测算法中我们需要估计可能源节点到所有感染传感器的传播时间，这里我们采用我们在第三章 3.4.1 小节中提出的有效传播时间，即

$$et(i,j)=-\log p_{ij} \tag{5-2}$$

其中 p_{ij} 是网络中任意两点 v_i 和 v_j 之间基于最短路径的传播概率，其计算方式参考第三章第 3.2.3 小节。有效传播时间考虑了传播概率的影响，相比基于最小路阶数的传播时间，以它估计的传播时间更能反映真实传播情况。

5.3.3 实验与分析

1．RDPC 算法检测准确度实验

(1) 实验设置

假设传播遵循 SI 模型，且所有传播之间相互独立，我们设置网络连边上的传播概率为 0.6，通过实验来验证 RDPC 算法的有效性以及检测准确性。实验采用之前工作中用到的网络：Power Grid 网络和 PPI-2 网络，网络的具体统计参数在第四章 4.4.1 小节给出。所统计的实验结果都是对 100 次实验数据做平均得到的。每次实验，首先随机选取网络种 30%的节点作为传感器，然后再从网络中随机选择一个节点作为传播源节点，之后根据第三章中介绍的 SI 模型模拟传播进行的过程看，具体的模拟流程图在第三章 3.3.4 小节给出。当传播过程进行到一定程度后，基于传感器提供的自身感染状态与感染时刻运用 RDPC 算法进行传播源节点的检测。

(2) 对比算法

实验中我们主要与 Wang[72]等人提出的基于极大似然估计的方法进行了对比。Wang 等人首先提出了反向传播算法，以此来减少传播源节点的搜索范围。在得到可能源节点集合后，Wang 等人使用极大似然的方法来最终确定唯一的传播源节点。他

们针对可能源节点集合中的每一个节点，模拟以其作为传播源节点时传播进行的过程。当传播在进行一段时间后，通过判断网络中各传感器状态与传感器真实记录状态的一致程度来确定真实传播源。具体的，他们先引入 $P_S(i,t_i;u)$ 和 $P_C(i,t_i;u)$，分别表示由节点 u 开始的传播在进行到 t 时刻时，节点 i 处于易感染状态和传染状态的概率。传染状态指代一个节点在 t 时刻由易感染状态转变为感染状态。$P_S(i,t_i;u)$ 的计算方式可参考第三章 3.2.1 小节，而 $P_C(i,t_i;u)$ 的计算方式则为：

$$P_C(i,t;u) = inf(i,t)\cdot P_S(i,t-1;u) \tag{5-3}$$

其中 $inf(i,t)$ 时节点表示节点 i 在 t 时刻被其邻居节点所感染的概率，具体计算方式在第三章 3.2.1 小节给出。

给定感染传感器集合 O_I 和易感染传感器集合 O_S 时，当以节点 u 开始的模拟传播过程进行了一段时间后，根据网络中各传感器对应的 $P_S(i,t_i;u)$ 与 $P_C(i,t;u)$ 值，以及其相应的感染时间 D，计算如下的极大似然函数：

$$L(u,t) = \prod_{i\in O_I} P_C(i,t_i;u)\prod_{j\in O_S} P_S(j,t;u) \tag{5-4}$$

其中的 t 表示当前模拟传播所进行的时间，$t_i = t - d_i$ 表示节点 i 的被感染时间。对于每一个可能源节点 $u\in U$，先找到以其开始的传播需要向外传播的时间 $\hat{t}_u$，即以 u 开始的传播在进行到 $\hat{t}_u$ 时刻时，网络中的各传感器状态最接近真实状态，$\hat{t}_u$ 应满足式(5-5)所示的关系：

$$L(u,\hat{t}_u) = \max_{O_I,O_S\subseteq O} L(u,t) \tag{5-5}$$

针对每个可能源节点 $u\in U$，得到其相应的传播时间后，通过最大化极大似然函数值确定真实传播源节点：

$$u_f = \arg[\max_{u\in U} L(u,\hat{t}_u)] \tag{5-6}$$

(3) 实验结果与分析

表 5.1 给出了两种算法在 Power Grid 网络和 PPI-2 网络上的平均误差距离以及平均计算用时，其中 RDPC 指我们提出的方法，ML 指 Wang[72]等人提出的基于极大似然估计的方法。

表5.1 RDPC 和 ML 算法平均误差距离与计算用时统计结果

网络	平均误差距离		计算用时(s)	
	RDPC	ML	RDPC	ML
Power Grid	0.68	1.03	31.58	139.08
PPI-2	0.37	0.87	5.17	75.12

从表 5.1 可以看出，RDPC 的检测准确度优于 ML 算法，且计算用时更少。以 Power Grid 网络为例，RDPC 算法平均误差距离为 0.68，说明 RDPC 算法找到的源节点平均

距离真实源节点不到一个路阶数，而 ML 算法的平均误差距离则为 1.03。同时，RDPC 平均计算时间仅为 31.58 秒，远远小于 ML 算法的 139.08 秒。RDPC 算法在得到可能源节点集后，只需针对每个节点检验其与感染传感器之间的有效传播时间与传感器的相对感染时间的相关性即可确定最终的传播源节点。相比而言，ML 算法需要针对每个可能源节点模拟传播进行的过程，耗时较大，而且模拟过程的随机性会导致检测准确度的下降。这里也给出了检测结果直方图，如图 5.3 所示，在 Power Grid 网络中，RDPC 算法由 70%左右可以准确找到传播源节点，而 ML 仅有不到 40%，PPI-2 网络中也是类似的结果。总的来看，RDPC 算法检测准确度更高，计算用时更少。

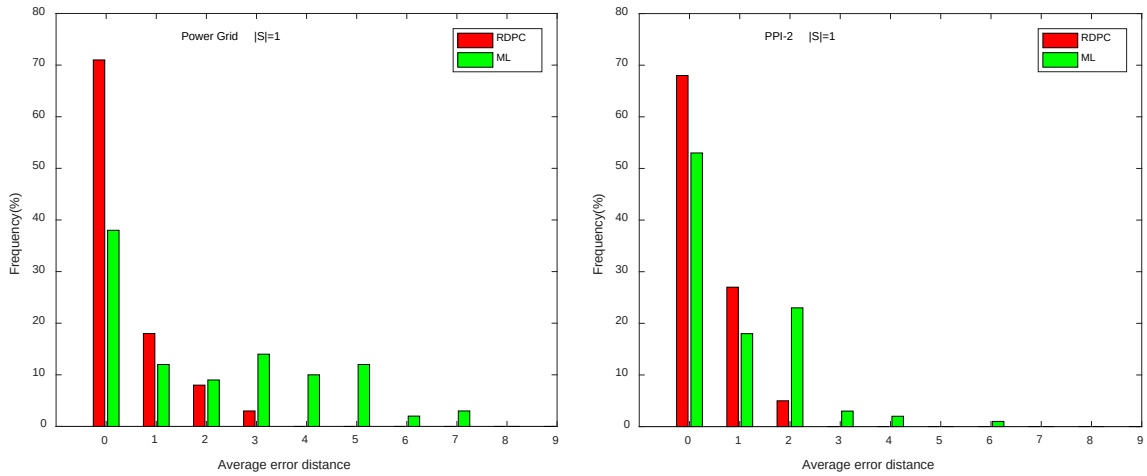


图5.3 RDPC 与 ML 算法平局误差距离直方图

2. RDPC 算法线性相关性实验

在我们所提出的方法中，当获得可能源节点集合后，我们利用所有感染传感器的相对感染时间和其与可能传播源间有效传播时间的线性相关性来确定最终的传播源。在这一节，我们通过实验来验证所提方法的有效性。

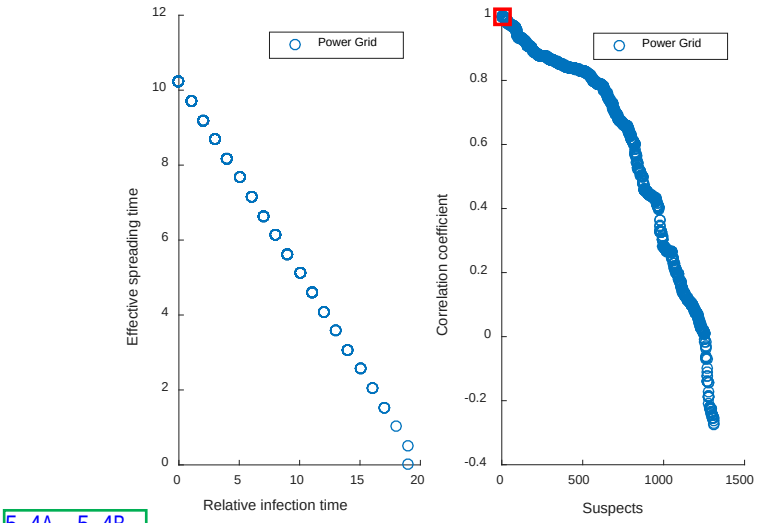
(1) 实验设置

实验设置与 5.3.3 小节中的实验设置一致，假设传播遵循 SI 模型，且所有传播之间相互独立，网络连边上的传播概率为 0.6。每次实验，随机选取网络种 30%的节点作为传感器。

(2) 实验结果与分析

图 5.4 给出了 Power Grid 网络上实验结果，其中图 5.4(A)表示的是网络中所有感染传感器的相对感染时刻和其与真实传播源间有效传播时间的关系，可以看出它们之间基本呈线性关系。我们还比较了所有可能源节点的相关系数，如图 5.4(B)可以看出，真实传播源节点（见方格）相对其他可能源节点具相对较高的相关系数。图 5.5 是 Yeast 网络上的实验结果，结果与 Power Grid 网络上的一致。总之，真实传播源节点与感染传感器间有效传播时间和感染传感器相对感染时间具有最大线性相关性证明

了我们所提方法的有效性。



5.4A 5.4B

图5.4 Power Grid 网络上相关性检测实验结果

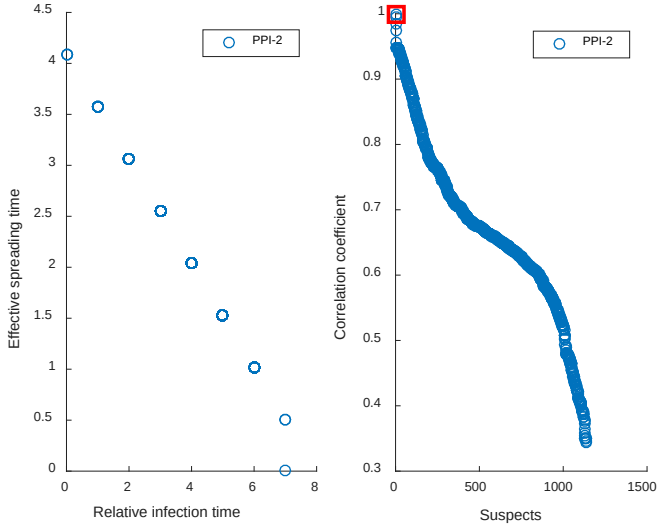


图5.5 PPI-2 网络上相关性检测实验结果

5.4 传感器观察下的多源节点检测算法

5.4.1 MRDPC 算法

目前存在的基于传感器观察的源点检测算法都是针对单源节点的检测，包括上面提出的 RDPC 算法，基于传感器观察的多源节点检测算法几乎没有，这一节我们通过一个简单的划分思路将前面提出的 RDPC 算法扩展到多源检测问题上，提出 MRDPC(Multiple Reverse Diffusion Pearson Correlation Algotirhm)算法。

根据前面介绍的，在传感器观察方式下，我们所能获取的有用信息就是感染传感

器的相对感染时刻了，但这一有用信息在多源检测问题中失去了作用。因为存在多个传播源，我们并不能确定哪些传感器是由同一个传播源所感染的，因此也不能计算出感染传感器的相对感染时间。为了解决这个问题，我们采用一种简单的根据距离来划分的思路，将所有感染传感器划分为 k 个集合，我们认为每个集合中的传感器是由同一个传播源所感染的，再分别针对每个传感器集合利用提出的 RDPC 算法进行传播源的检测，从而实现多个传播源节点的检测。MRDPC 算法的具体步骤如下：

Algorithm 5.3：Initialize Infected Sensor Set.

Input：Network G_n，infected sensors O_I and source number k

Select two infected sensors o_1 and o_2 with the maximum distance(hops),i.e., $d(o_1,o_2)=\max\limits_{a,b\in O_I\in G_n}\{d(a,b)\}$, and let $O^{'}=\{o_1,o_2\}$.

Let $i=|O^{'}|$ and select an infected sensor node $o_{i+1}\in O_I\setminus O^{'}$ such that

$$d(o_{i+1},O^{'})=\max_{a\in O_I\setminus O^{'}}\left(\min_{b\in O^{'}}d(a,b)\right)$$

i.e., selecting an infected node from $O_I\setminus O^{'}$ that is furthest away from set $O^{'}$.

Repeat this step until $|O^{'}|=k$.

Output：A set of infeted sensor $O^{'}=\{o_1,o_2,...o_k\}$.

第 1 步：首先从网络中选取 k 个距离最远的感染传感器，这样的选取方式可以最大程度的保证这 k 个传感器是被不同的传播源所感染的，具体的选择细节参考算法 5.3。

第 2 步：根据得到的 k 个距离最远的感染传感器，将所有感染传感器划分为 k 个集合。首先，以这 k 个感染传感器分别初始化一个集合。然后针对剩余的每一个感染传感器，根据其到这 k 个感染传感器距离，将其划分到这 k 个传感器中距离它最近的传感器所对应的集合中去。一般而言，距离越近的节点越有可能是被相同传播源所感染，我们的划分过程正是基于这样的思想。完成这一步后，网络中的所有被感染传感器已经被分为成 k 个集合 $C=\{C_1,C_2,...C_k\}$。

第 3 步：在得到感染传感器的 k 个集合之后，我们将每个集合中的传感器视为是由同一个传播源所感染的，然后根据前面的相对感染时刻计算方式计算每个集合中感染传感器的相对感染时间。假设有集合 $C_i=\{o_1^i,o_2^i,...o_l^i\}$，且该集合中传感器所记录的感染时刻记为 $T_i=\{t_1^i,t_2^i,...t_l^i\}$，则该集合中传感器对应的相对感染时刻为：$D_i=\{d_1^i,d_2^i,...d_l^i\}$，其中 $d_j^i=\max(T_i)-t_j^i$。

第 4 步：利用反向传播算法选择可能的传播源节点。首先给每个感染传感器赋值为各自的标签(ID)，然后根据上一步计算出来的相对感染时刻，将这些标签向外传播。

与前面的单源检测中的反向传播算法稍有差别，这里我们并不是选择可以同时接受到所有感染传感器标签的节点作为可能源节点，而是检测是否有节点可以同时收到某一个集合中的所有传感器的标签，如果有则认为该节点是该集合的一个可能源节点。

第 5 步：第 4 步之后，我们针对每个传感器集合 $C_i=\{o_1^i,o_2^i,...o_l^i\}$ 都会得到一个相应的可能源节点集 $U_i=\{u_1^i,u_2^i,...u_v^i\}$。然后针对该集合中的每一个节点 u_j^i 计算其到集合 $C_i=\{o_1^i,o_2^i,...o_l^i\}$ 中所有传感器的传播时间与这些传感器的相对感染时刻 $D_i=\{d_1^i,d_2^i,...d_l^i\}$ 的线性相关性，找到一个具有最大线性相关性的节点 s_i 作为该集合的传播源节点。最终可以得到一个含有 k 个传播源节点的集合 $S=\{s_1,s_2,...s_k\}$，从而实现基于传感器观察的多源节点的检测。

5.4.2 实验与分析

(1) 实验设置

我们在 Power Grid 网络与 PPI-2 网络上通过实验验证了 MRDPC 算法的有效性。假设传播遵循 SI 模型，网络连边上的传播概率均为 0.6 且传播之间相互独立。每次实验，首先随机选网络种 30%的节点作为传感器，然后再从网络中随机选择 k 个节点作为传播源节点，然后根据 SI 模型模拟传播过程。当传播过程进行到一定程度后，基于传感器提供的自身感染状态与感染时刻运用 MRDPC 算法进行多源节点的检测。

(2) 实验结果与分析

表5.2 MRDPC 算法的平均误差距离统计结果

实验设置		平均误差距离
网络	源点个数	MRDPC
Power Grid	2	0.5000
	3	1.2653
	4	1.6720
	5	2.1250
PPI-2	2	1.7500
	3	2.2500
	4	2.6338
	5	2.9340

表 5.2 是 MRDPC 算法的检测结果，可以看出无论是在 Power Grid 网络还是 PPI-2 网络上 MRDPC 算法都可以实现多源节点的检测。在 Power Grid 网络上，当传播源节点个数为 2 时平均误差距离为 0.5，即检测出的源节点平均距离真实源节点为 0.5

个路阶数，传播源个数为 5 时也在 2-3 个路阶数之内。在 PPI-2 网络上，传播源为 2 时检测结果在 1-2 个路阶数以内，传播源为 5 时也在 2-3 个路阶数内。图 5.6 是 MRDPC 算法的平均误差距离直方图，可以看出当传播源为 2 时，无论是在 Power Grid 网络还是 PPI-2 网络上，MRDPC 算法都有 60%以上的概率可以检测出多个传播源，传播源为 3 时，也在 30%以上。总之，MRDPC 算法可以很好的实现传感器观察下的多源节点检测。

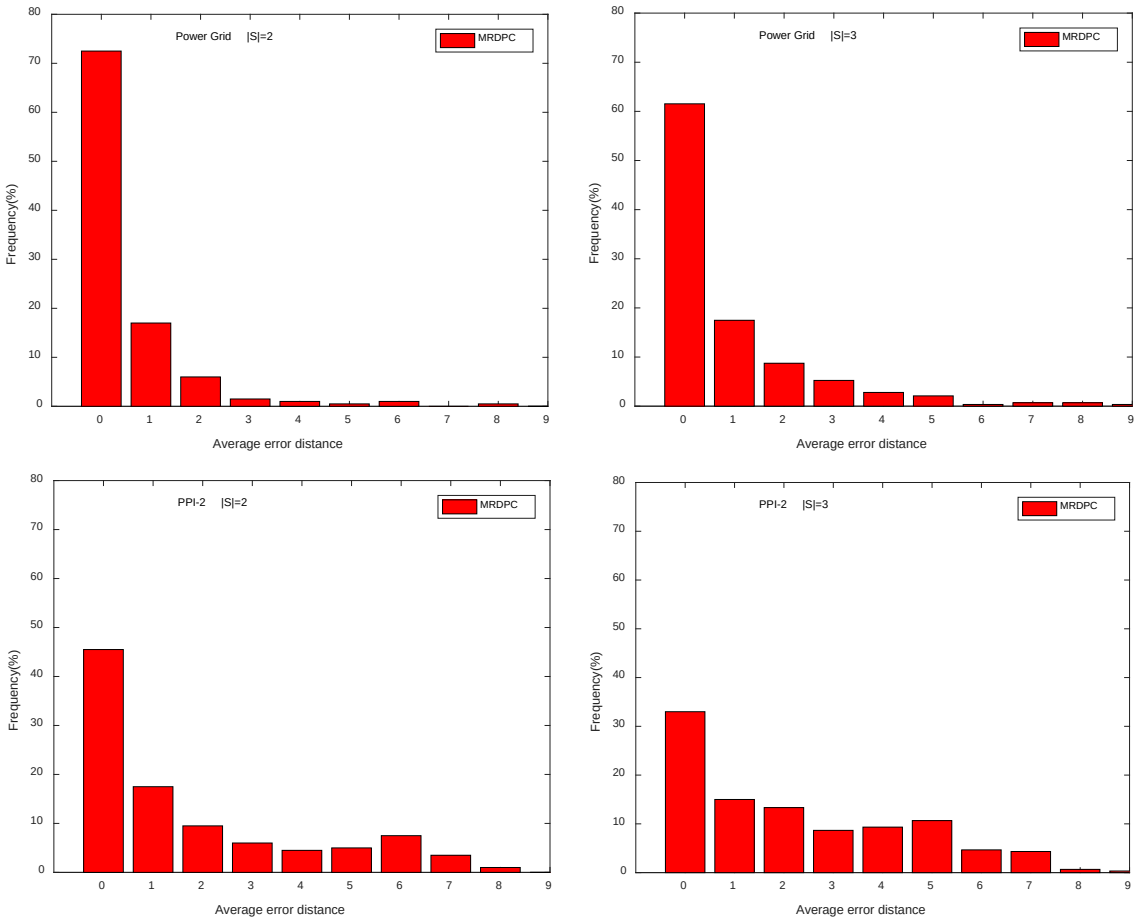


图5.6 MRDPC 算法平均误差距离直方图

5.5 本章小结

本章研究了传感器观察方式下的源点检测问题。首先给出了传感器观察下的源点检测问题描述，然后假设传播遵循 SI 模型，提出了传感器观察下的单源节点检测算法，RDPC 算法，RDPC 算法首先利用反向传播算法筛选出网络中的可能源节点，然后针对每个可能源节点，检测其到所有感染传感器的传播时间与感染传感器记录的相对感染时间之间的线性相关性，选择具有最大线性相关性的节点作为传播源节点。通过实验验证 RDPC 算法的有效性以及检测准确度。

此外，通过一个简单的划分思路将 RDPC 算法扩展到多源节点检测问题上，提出 MRDPC 算法，实验结果表明 MRDPC 算法可以很好的解决传感器观察下的多源节点检测问题。

第六章 总结与展望

6.1 总结

论文研究了一般网络上的多源节点检测问题，针对不同传播模型、观察方式，提出不同的算法来实现多源节点的检测，主要工作内容如下：

1. 提出基于 SI 传播模型的多源节点检测算法。基于网络拓扑结构与传播源个数，从传播时间的角度出发，将 SI 模型下的多源节点检测问题转化为寻找网络中可以最小化分区传播时间之和的k个节点问题，并抽象出该问题的目标函数，然后提出 KST 算法以迭代的方式最小化该目标函数，从而实现多源节点的检测。通过在真实网络中的实验仿真，验证了 KST 算法可以得到相对较高检测准确度。同时，考虑到基于最小路阶数的传播时间没有考虑传播概率的问题，提出有效传播时间来估计网络中任意两点间传播时间，通过实验证明了提出的有效传播时间可以进一步优化 KST 算法的检测准确度。最后，针对实际中传播源个数难以提前获知这一情况，还提出了一种可以估计传播源个数的启发式算法，实验结果表明该算法能够以比较高的精度来进行传播源个数的估计。

2. 在前面提出的 KST 算法的基础之上提出了 SIR 传播模型下的多源节点检测算法，WP-KST 算法。首先针对 SIR 模型下不能正确区分恢复节点和易感染节点的问题，提出了一种权值传播算法，通过将网络中的感染节点的初始权值向外传播，网络中的恢复节点会收到相对更高的权值，从而实现恢复节点的检测。仿真实验证明权值传播算法可以很好的检测出网络中的恢复节点，完成缺失信息的填充。接着在由由感染节点与恢复节点以及这些节点间的连边组成的扩展感染网络后，运用 KST 算法进行多源节点的检测。实验结果表明 WP-KST 算法可以很好的解决 SIR 模型下的多源节点检测问题，且具有较高的检测准确度。

3. 研究了传感器观察方式下的源节点检测问题。传感器观察是针对现实传播中所有节点状态信息不易获取这一现象提出的一种比较符合实际情况的先验知识获取方式，通过提前监控网络中的一些节点或者随机查询网络中的一些节点，根据这些节点所提供的状态信息实现传播源节点的检测。假设传播遵循 SI 模型，首先提出了一种基于传感器观察的单源节点检测算法，RDPC 算法。RDPC 算法首先利用反向传播算法筛选出网络中的可能源节点，之后针对每个可能源节点，检测其到所有感染传感器的传播时间与感染传感器记录的相对感染时间之间的线性相关性，选择具有最大线性相关性的节点作为传播源节点。通过实验验证了 RDPC 算法具有较高的检测准确度。此外，通过一个简单的划分思路，将网络中的所有感染传感器划分为k个集合，认为

每个集合中的传感器都是由同一个传播源所感染的，以此方式将 RDPC 算法扩展到了多源节点检测问题上，提出 MRDPC 算法，最后的实验验证了 MRDPC 算法可以很好的解决传感器观察方式下的多源节点检测问题。

6.2 展望

本文主要研究了一般网络上的多源节点检测问题，针对论文研究过程中发现的问题与不足之处，今后仍然需要在以下几个方面进行深入研究：

1. 本文第三章中提出的 KST 算法可以很好的完成 SI 模型下的多源节点检测，且检测准确度相对较高。KST 算法的计算复杂度为$O(n^2)$，相对其他多源检测算法有所提高。随着大数据时代的来临，复杂网络的规模越来越大，许多算法由于复杂度过高很难在实际应用中发挥应有的作用，所以仍需要进一步压缩算法的时间复杂度。

2. 在第五章研究的传感器观察方式下的源节点检测问题中，传感器位置的选取采用随机选择方法。而在实际中，可以根据具体情况选择一些特殊位置的节点作为传感器，比如选择具有较高介数的节点、入度大的节点，不同的传感器选择方式产生不同的传感器集合，使得检测精度有所不同。此外，传感器数量也是影响源点检测准确度的一个要素，论文中选取 30%的节点作为传感器。实际中，传感器的多少直接决定着咨询代价、有用信息量等关键因素，今后的工作应该进一步研究传感器数量与位置对检测准确的影响。

3. 目前所有的源点检测方法都是基于单一网络，即认为信息传播是在一个独立的网络上进行的，然而现实中信息传播是一个十分复杂的过程，可能会涉及到多个有连接关系的网络，例如人们在听到来自 Facebook 网络的流言后可能会将其转发到 Twitter、博客等其他在线社交网络上。因此，组合网络中的源点检测问题需要投入更多关注。

参考文献

[1] ULRIKE M. Statistical Mechanics of Complex Networks[J]. Review of Modern Physics, 2007,74(1).

[2] WHEELER K P. Reviews of modern physics.[M]. American Physical society, 1929. 717-730.

[3] STROGATZ S H. Exploring complex networks[J]. NATURE, 2001,410(6825):268.

[4] 戴存礼. 复杂网络上动力学系统的同步行为研究[D]. 南京航空航天大学, 2008.

[5] LORRAIN F, WHITE H C. Structural equivalence of individuals in social networks[J]. Social Networks, 1977,1(1):67-98.

[6] JACCARD P. Etude de la distribution florale dans une portion des Alpes et du Jura[J]. Bulletin De La Societe Vaudoise Des Sciences Naturelles, 1901,37(142):547-579.

[7] NEWMAN M E, GIRVAN M. Finding and evaluating community structure in networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004,69(2 Pt 2):26113.

[8] NEWMAN M E J. Finding community structure in networks using the eigenvectors of matrices. Phys: Rev. E, 2013[C]. 36104.

[9] ZHOU T, LÜ L, ZHANG Y C. Predicting missing links via local information[J]. EUR PHYS J B, 2009,71(4):623-630.

[10] HANLEY J A, MCNEIL B J. The meaning and use of the area under a receiver operating characteristic (ROC) curve.[J]. RADIOLOGY, 1982,143(1):29.

[11] FENG X, ZHAO J C, XU K. Link prediction in complex networks: a clustering perspective[J]. EUR PHYS J B, 2012,85(1):1-9.

[12] YEUNG M K S, TEGNÉR J, COLLINS J J. Reverse engineering gene networks using singular value decomposition and robust regression.[J]. P NATL ACAD SCI USA, 2002,99(9):6163-6168.

[13] CHING E S, LAI P Y, LEUNG C Y. Reconstructing weighted networks from dynamics[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2015,91(3):30801.

[14] LAUBENBACHER R, STIGLER B. A computational algebra approach to the reverse engineering of gene regulatory networks[J]. J THEOR BIOL, 2003,229(4):523-537.

[15] ARENAS A, DÍAZGUILERA A, GUIMERÀ R. Communication in networks with hierarchical branching.[J]. PHYS REV LETT, 2001,86(14):3196-3199.

[16] LI T, WANG T, SONG A G, et al. Exponential synchronization for arrays of coupled neural networks with time-delay couplings[J]. International Journal of Control Automation & Systems, 2011,9(1):187-196.

[17] NOWAK M A, MAY R M. The spatial dilemmas of evolution[J]. International Journal of Bifurcation & Chaos, 1993,3(01):35-78.

[18] SANTOS F C, PACHECO J M. Scale-free networks provide a unifying framework for the emergence of cooperation.[J]. PHYS REV LETT, 2005,95(9):98104.

[19] 孙玺菁，司守奎. 复杂网络算法与应用[M]. 国防工业出版社, 2015.

[20] JIANG J, WEN S, YU S, et al. Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies[J]. IEEE Communications Surveys & Tutorials, 2017,19(1):465-481.

[21] KEPHART J O. Directed-Graph Epidemiological Models of Computer Viruses.: Research in Security and Privacy[C], 1991 IEEE Computer Society Symposium on, 2002. 343.

[22] HAN L, HAN S, DENG Q, et al. Source Tracing and Pursuing of Network Virus[C], IEEE International Conference on Computer and Information Technology Workshops, 2008. 230-235.

[23] NEWMAN M E, GIRVAN M. Finding and evaluating community structure in networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004,69(2 Pt 2):26113.

[24] SCOGLIO C, SCHUMM W, SCHUMM P, et al. Efficient mitigation strategies for epidemics in rural regions.[J]. PLOS ONE, 2010,5(7):e11569.

[25] SHAH D, ZAMAN T. Rumors in a Network: Who's the Culprit?[J]. IEEE T INFORM THEORY, 2011,57(8):5163-5181.

[26] SEKAR V, XIE Y, MALTZ D A, et al. Toward a Framework for Internet Forensic Analysis[J]. In Third Workshop on Hot Topics in Networking (HotNets-III, 2004.

[27] SEKAR V, MALTZ D A, REITER M K. Worm Origin Identification Using Random Moonwalks[J]. IEEE Symposium on Security & Privacy, 2005:242-256.

[28] SHAH D, ZAMAN T. Detecting sources of computer viruses in networks: theory and experiment[J]. Acm Sigmetrics Performance Evaluation Review, 2010,38(1):203-214.

[29] DONG W X, ZHANG W, TAN C W. Rooting out the rumor culprit from suspects: IEEE International Symposium on Information Theory Proceedings, 2013[C]. 2671-2675.

[30] KARAMCHANDANI N, FRANCESCHETTI M. Rumor source detection under

probabilistic sampling: IEEE International Symposium on Information Theory Proceedings, 2013[C]. 2184-2188.

[31] SHAH D, ZAMAN T. Rumor centrality: a universal source detector[M]. ACM, 2012. 199-210.

[32] LUO W, TAY W P, LENG M. Identifying Infection Sources and Regions in Large Networks[J]. IEEE T SIGNAL PROCES, 2013,61(11):2850-2865.

[33] MILLING C, CARAMANIS C, MANNOR S, et al. On identifying the causative network of an epidemic: Communication, Control, and Computing, 2012[C]. 909-914.

[34] ZHU K, YING L. Information source detection in the SIR model: A sample path based approach: Information Theory and Applications Workshop, 2013[C]. 1-9.

[35] LUO W, TAY W P. Finding an infection source under the SIS model: IEEE International Conference on Acoustics, Speech and Signal Processing, 2013[C]. 2930-2934.

[36]LAPPAS T, TERZI E, GUNOPULOS D, et al. Finding effectors in social networks[J]. 2010.

[37] BROCKMANN D, HELBING D. The hidden geometry of complex, network-driven contagion phenomena.[J]. SCIENCE, 2013,342(6164):1337.

[38] ALTARELLI F, BRAUNSTEIN A, DALL'ASTA L, et al. Bayesian inference of epidemics on networks via belief propagation[J]. PHYS REV LETT, 2013,112(11):118701.

[39] COMIN C H, COSTA L F. Identifying the starting point of a spreading process in complex networks[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2011,84(5 Pt 2):56105.

[40] BERRY R, SUBRAMANIAN V G. Spotting trendsetters: Inference for network games: Communication, Control, and Computing, 2012[C]. 1697-1704.

[41] CHEN Z, ZHU K, YING L. Detecting Multiple Information Sources in Networks under the SIR Model[J]. IEEE Transactions on Network Science & Engineering, 2016,3(1):17-31.

[42] LUO W, TAY W P. Identifying multiple infection sources in a network: Signals, Systems and Computers, 2013[C]. 1483-1489.

[43] LU Y M. A fast Monte Carlo algorithm for source localization on graphs[J]. 2013,8858:88581N.

[44] PINTO P C, THIRAN P, VETTERLI M. Locating the source of diffusion in large-scale networks[J]. Phys.rev.lett, 2012,109(6):68702.

[45] SEO E, MOHAPATRA P, ABDELZAHER T. Identifying rumors and their sources in social networks[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2012,8389:40.

[46] ALDALAHMEH S, GHOGHO M. Robust distributed detection, localization, and estimation of a diffusive target in clustered wireless sensor networks: IEEE International Conference on Acoustics, Speech and Signal Processing, 2011[C]. 3012-3015.

[47] FIORITI V, CHINNICI M. Predicting the sources of an outbreak with a spectral technique[J]. Computer Science, 2014,8:6775-6782.

[48] SONG L P, JIN Z, SUN G Q. Modeling and analyzing of botnet interactions[J]. Physica A Statistical Mechanics & Its Applications, 2011,390(2):347-358.

[49] YAO Y, LUO X, GAO F, et al. Research of a Potential Worm Propagation Model based on Pure P2P Principle: International Conference on Communication Technology, 2006[C]. 1-4.

[50] HETHCOTE H W. The Mathematics of Infectious Diseases[J]. SIAM REV, 2000,42(4):599-653.

[51] COOKE K L, VAN D D P. Analysis of an SEIRS epidemic model with two delays[J]. J MATH BIOL, 1996,35(2):240-260.

[52] ALBERT R, JEONG H, BARABASI A L. Error and attack tolerance of complex networks[J]. NATURE, 2004,406(6794):378.

[53] FALOUTSOS M, FALOUTSOS P, FALOUTSOS C. On power-law relationships of the Internet topology[J]. Comput.coomun.rev, 1999,29(4):251-262.

[54]NEWMAN M E J. Networks :an introduction[M]. Oxford University Press, Inc., 2010. 741-743.

[55] HOLME P, KIM B J, YOON C N, et al. Attack vulnerability of complex networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2002,65(2):56109.

[56] LIU Y Y, SLOTINE J J, BARABÁSI A L. Controllability of complex networks.[J]. NATURE, 2011,473(7346):167.

[57] NEWMAN M E J. A measure of betweenness centrality based on random walks[J]. Social Networks, 2003,27(1):39-54.

[58] RESTREPO J G, OTT E, HUNT B R. Characterizing the dynamical importance of network nodes and links.[J]. PHYS REV LETT, 2006,97(9):94102.

[59] ZOU C C, TOWSLEY D, GONG W. Modeling and Simulation Study of the Propagation and Defense of Internet E-mail Worms[J]. IEEE Transactions on

Dependable & Secure Computing, 2007,4(2):105-118.

[60] CHOU Y F, HUANG H H, CHENG R G. Modeling Information Dissemination in Generalized Social Networks[J]. IEEE COMMUN LETT, 2013,17(7):1356-1359.

[61] JIANG J, WEN S, YU S, et al. K-Center: An Approach on the Multi-Source Identification of Information Diffusion[J]. IEEE Transactions on Information Forensics & Security, 2015,10(12):2616-2626.

[62] WANG Y, WEN S, XIANG Y, et al. Modeling the Propagation of Worms in Networks: A Survey[J]. IEEE Communications Surveys & Tutorials, 2014,16(2):942-960.

[63] WEN S, ZHOU W, ZHANG J, et al. Modeling Propagation Dynamics of Social Network Worms[J]. IEEE Transactions on Parallel & Distributed Systems, 2013,24(8):1633-1643.

[64] LUO W, TAY W P, LENG M. How to Identify an Infection Source With Limited Observations[J]. IEEE J-STSP, 2014,8(4):586-597.

[65] ZHU G M, YANG H J, YANG R, et al. Uncovering evolutionary ages of nodes in complex networks[J]. EUR PHYS J B, 2012,85(3):1-6.

[66] TILLETT H E. Infectious Diseases of Humans; Dynamics and Control.[J]. Epidemiology & Infection, 1992,108(1):211-757.

[67] KEMPE D, KLEINBERG J, TARDOS É. Maximizing the spread of influence through a social network: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003[C]. 137-146.

[68] SAITO K, KIMURA M, OHARA K, et al. Efficient discovery of influential nodes for SIS models in social networks[J]. Knowledge & Information Systems, 2012,30(3):613-635.

[69] WANG P, YU X, LÜ J. Identification and evolution of structurally dominant nodes in protein-protein interaction networks[J]. IEEE Transactions on Biomedical Circuits & Systems, 2014,8(1):87.

[70] ZANG W, ZHANG P, ZHOU C, et al. Locating multiple sources in social networks under the SIR model: A divide-and-conquer approach[J]. J COMPUT SCI-NETH, 2015,10:278-287.

[71] ZANG W, ZHANG P, ZHOU C, et al. Discovering Multiple Diffusion Source Nodes in Social Networks [J]. Procedia Computer Science, 2014,29:443-452.

[72] WANG D, WEN S, XIANG Y, et al. Catch Me If You Can: Detecting Compromised Users Through Partial Observation on Networks: IEEE International Conference on

Distributed Computing Systems, 2017[C].

[73] JIANG J, WEN S, YU S, et al. Rumor Source Identification in Social Networks with Time-varying Topology[J]. IEEE Transactions on Dependable & Secure Computing, 2018,PP(99):166-179.

致谢

时光荏苒，转眼间三年的硕士生涯已接近尾声。三年多的时光既漫长又短暂，其中充满了酸甜苦辣，更有收获和成长。三年来，感谢陪我一起度过美好时光的每位尊敬的老师和亲爱的同学，在这里感想他们三年来对我在学习以及生活上的帮助。

首先，要诚挚感谢我的导师吴建设教授。吴老师教授严肃的科学态度，严谨的治学精神，精益求精的工作作风，深深地感染和激励着我。是他把我带入学术科学研究的殿堂，一步步地教会我如何写科研工作报告，如何记录科研数据，如何分析算法、设计算法。每当我遇到科研难题时，吴老师都会给我进行耐心细致的解答，提出许多宝贵的意见。吴老师的淳淳教诲我都会铭记在心，在此我衷心的向吴老师表示感谢。

感谢我所在的西安电子科技大学人工智能学院智能感知与图像理解教育部重点实验室我们提供的科研环境和学术氛围。三年来，所里举办了一系列学术讲座，尤其是每年一次的学术之春、学术之秋，请来科学院院士、国内外知名专家教授，让我们了解前沿学术以及未来的发展方向，对我们的科研意义重大。

感谢所有同门师兄弟姐妹，是他们共同创造了一个积极进取、活泼向上的科研环境。在科研中，他们帮我分析问题，开拓思维；在生活中，互相帮助，与我一起在科研闲暇时娱乐放松，和他们在一起的三年，我收获了很多。

还要感谢我的家人，二十多年来，是他们养育了我，教导了我。我人生中的一切，无不源于他们。身在千里之外，他们仍时刻地关心着我，挂念着我，是他们的支持给了我奋斗的动力。浓浓亲情，无以言表。

最后，感谢陪伴我一起走过每一段路的人，谢谢。

作者简介

1. 基本情况

李晓杰，男，陕西宝鸡人，1992 年 2 月出生，西安电子科技大学人工智能学院电路与系统专业 2015 级硕士研究生。

2. 教育背景

2011.09～2015.07 西安电子科技大学，本科，专业：电子信息科学与技术
2015.09～ 西安电子科技大学，硕士研究生，专业：电路与系统

3. 攻读硕士学位期间的研究成果

3.1 发表学术论文

3.2 申请（授权）专利

3.3 参与科研项目及获奖