Detecting Multiple Information Sources in Networks under the SIR Model

Zhen Chen, Kai Zhu, and Lei Ying, *Member, IEEE*

Abstract—In this paper, we study the problem of detecting multiple information sources in networks under the Susceptible-Infected-Recovered (SIR) model. First, assuming the number of information sources is known, we develop a sample-path-based algorithm, named clustering and localization, for trees. For g-regular trees, the estimators produced by the proposed algorithm are within a constant distance from the real sources with a high probability. We further present a heuristic algorithm for general networks and an algorithm for estimating the number of sources when the number of real sources is unknown.

Index Terms—Sample path approach, information source detection, multiple information sources

1 Introduction

Recently, there have been a lot of interests in the problem of detecting information sources in networks. The solutions to this problem have important applications in practice, such as identifying the sources of infectious diseases and finding the sources of leaked confidential information. Shah and Zaman analytically studied this problem under the SI model and developed the rumor centrality estimator [1], [2], [3]. Detecting multiple information sources using the rumor centrality estimator has been investigated in [4], [5], and detecting a single information source with partial observations by using the rumor centrality estimator has been considered in [6]. In [7], the detection rate of the rumor centrality estimator when a priori distribution of the source node is given has been evaluated.

Besides the SI model, information source detection under the SIR model, in which "susceptible" nodes and "recovered" nodes cannot be distinguished, has also been studied. There are a number of scenarios where it is useful to model "recovered" nodes and assume "susceptible" nodes and "recovered" nodes are indistinguishable. For example, in a blog-network, a user may post a rumor and then subsequently remove the rumor after realizing that it is not the truth. After the post was deleted, from the data crawled from the web, which has been a common methods to collect online social network datasets, it is difficult to distinguish whether the user has never posted the rumor or posted/deleted it. Similarly, classified information may spread in a social network, but people who spread the information may refuse to admit that they know the information and have spread it. This scenario again can be modeled as "recovered" but indistinguishable from "susceptible".

• *The authors are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ. E-mail: {zchen113, kzhu17, lei.ying.2}@asu.edu.*

Manuscript received 19 Nov. 2014; revised 13 Jan. 2016; accepted 18 Jan. 2016. Date of publication 28 Jan. 2016; date of current version 11 Mar. 2016. Recommended for acceptance by A. Wierman.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TNSE.2016.2523804

Zhu and Ying developed a sample-path-based estimator in [8] for detecting a single information source under the SIR model. They later proved that the sample path estimator remains to be an effective estimator even with sparse observations [9]. The effectiveness of the sample path estimator for the SI model with partial observations and for the SIS model have been investigated in [10] and [11], respectively.

Several other source detecting algorithms have also been proposed recently, including an eigenvalue-based estimator [12], a dynamic message-passing algorithm based on a mean-field-like approximation for the SIR model [13], a fast Monte Carlo algorithm [14], and an algorithm that utilizes sparsely placed observers [15].

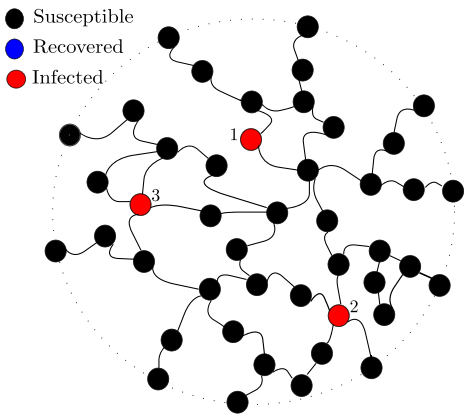
This paper considers the problem of detecting multiple information sources under the SIR model. It is not uncommon to have multiple information sources. For example, confidential information can be leaked from different sources and an infectious disease can start from multiple locations. We study this multi-source detection problem under the SIR model. Motivated by the sample-path-based estimator proposed in [8], we first present a clustering and localization (CL) algorithm for tree networks, where the number of real sources is assumed to be known. We then prove that on g-regular trees, the distances between the estimators given by the algorithm and the real sources are upper bounded by a constant with a high probability. We further present a heuristic algorithm for general networks and an algorithm for estimating the number of sources when the number of sources is unknown.

2 Basic Model

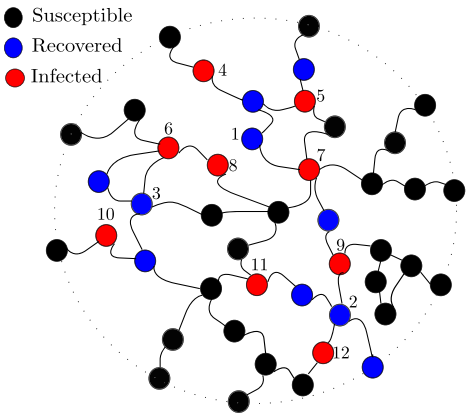
In this section, we introduce the SIR model and the multi-source detection problem.

2.1 SIR Model

The network is defined to be an undirected graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node in graph G may represent a person, a computer or a mobile device. An edge represents a communication channel such that information can be transmitted from one node to another if there is an edge between these two nodes.



(a) Network state when time t = 0.



(b) Network state when the snapshot is taken.

Fig. 1. An example of multi-source detection.

Define $s(t)$ $(i(t), r(t))$ to be the fraction of nodes that are susceptible (infected, recovered) at time t. In the classical SIR model, with the uniform contact assumption and fully mixed approximation, the dynamic of the SIR system can be expressed by the following set of differential equations:

$$\frac{\mathrm{d}s}{\mathrm{d}t} = -\beta si, \quad \frac{\mathrm{d}i}{\mathrm{d}t} = \beta si - \gamma i, \quad \frac{\mathrm{d}r}{\mathrm{d}t} = \gamma i, \tag{1}$$

where β is the infection rate and γ is the recovery rate.

Since the network structure is ignored in those equations, these differential equations can only be used for a rough approximation of the state of the network. Therefore, in this paper, we consider the following multi-source Susceptible-Infected-Recovered (SIR) model for information diffusion in networks. In the SIR model, every node has three states: susceptible (S), infected (I) and recovered (R) such that:

- a susceptible node may be infected by his/her infected neighbors,
- an infected node may recover, and
- a recovered node cannot be infected again.

We consider a time slotted system. At the beginning of each time slot, a susceptible node is infected by each of its infected neighbors with probability q, and each infected node recovers with probability p. Assuming the number of infected neighbors of a susceptible node is n, the probability the node becomes infected is $1-(1-q)^n$. Initially, at $t=0$, all nodes are in the susceptible state except a set of source nodes, denoted by $\mathcal{S}$.

TABLE 1
Notations

$G(\mathcal{V},\mathcal{E})$	the tree that our detection problem is based on.
$\mathcal{V}_G$	the set of vertices of G.
$\mathcal{V}_S$	the set of actual sources.
S	$\lvert\mathcal{V}_S\rvert$, the number of original sources.
$\mathcal{V}_I$	the set of infected nodes when we take the snapshot.
$d(a,b)$	the distance between node a and b on tree G.
(a,b)	the path between node a and b, and also it represents the set of nodes on that path.
ζ_i $(i=1,\ldots,S)$	The actual information sources.
γ_i $(i=1,\ldots,S)$	The estimators we find.
$\mathcal{V}_\gamma$	the set of estimators.
$d_{i,j}$	$d_{i,j}=d(\zeta_i,\zeta_j)$.
d	$d=\min_{1\le i,j\le S} d_{i,j}$.
$K^a_{b,c}$	Node satisfies $(a,c)\cap(a,b)=(a,K^a_{b,c})$.
$(a,b)\subset(c,d)$	path (a,b) is contained in path (c,d).
$a\in(c,d)$	node a is on path (c,d).
$d(a,\mathcal{N})$	$\mathcal{N}$ is a set of nodes and $\min_{b\in\mathcal{N}} d(a,b)$
t^I_a	The time that node a got infected
t^R_a	The time node a recovered

2.2 Problem Description

The objective of this paper is to locate the set of sources $\mathcal{S}$ given a snapshot of the network in which we can distinguish infected nodes from other nodes, but cannot distinguish susceptible nodes and recovered nodes. We say a node is healthy if the node is in either the susceptible state or recovered state.

Define X_v to be the state of node v in the given snapshot and $\mathbf{X}=\{X_v; v\in\mathcal{V}\}$, where

$$X_v = \begin{cases} 1, & \text{if } v \text{ is in the infected state;} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Denote by $\mathcal{V}_I$ the set of observed infected nodes in the snapshot. We further assume the number of sources ($S=|\mathcal{S}|$) is known.

Consider Fig. 1 as an example. Fig. 1a is the snapshot at $t=0$, in which there are three information sources, node 1, node 2 and node 3. When we take a snapshot at some time, the network state may look like Fig. 1b. Define the set of nodes to be $\mathcal{V}$ and $\mathcal{V}_I=\{3,4,5,6,7,8,9,10,11,12\}$. Since we cannot distinguish recovered nodes and susceptible nodes, the information we have is

$$\mathbf{X}=\{\forall i\in\mathcal{V}_I, X_i=1, \text{ and } \forall j\in\mathcal{V}\setminus\mathcal{V}_I, X_j=0\}.$$

Then we need to use $\mathbf{X}$ to identify three nodes in the network as our estimators for the sources.

3 Main Results

We summarize the main results in this section and the notations in this paper is in Table 1.

3.1 Multi-Source Detection on Tree Neworks

In this section, we present a multi-source detection algorithm for tree networks, named clustering and localization. The algorithm is presented in Algorithm 1.

Algorithm 1. Clustering and Localization

1: Select two infected nodes e_1 and e_2 with the maximum distance, i.e.,

$$d(e_1, e_2) = \max_{a,b \in \mathcal{V}_I} d(a, b),$$

and let $\mathcal{B} = \{e_1, e_2\}$.

2: Let $i = |\mathcal{B}|$ and select an infected node $e_{i+1} \in \mathcal{V}_I \setminus \mathcal{B}$ such that

$$d(e_{i+1}, \mathcal{B}) = \max_{a \in \mathcal{V}_I \setminus \mathcal{B}} d(a, \mathcal{B}),$$

i.e., selecting an infected node from $\mathcal{V}_I \setminus \mathcal{B}$ that is furthest away from set $\mathcal{B}$. Here $d(a, \mathcal{B}) = \min_{u \in \mathcal{B}} d(a, u)$, where $d(a, v)$ is the distance between node a and v on graph $G(\mathcal{V}, \mathcal{E})$. Repeat this step until $|\mathcal{B}| = \min\{S, |\mathcal{V}_I|\}$.

3: Without loss of generality, assume $|\mathcal{B}| = S$. Partition the set of infected nodes into S sets: $\mathcal{V}_I^{(s)}$ for $s = 1, \ldots, S$. An infected node a is assigned to set $\mathcal{V}_I^{(s)}$ if

$$d(a, e_s) = \min_{j=1,\ldots,S} d(a, e_j).$$

Ties are broken arbitrarily.

4: For each $\mathcal{V}_I^{(s)}$, compute the infection radius

$$r_s = \left\lfloor \max_{a,b \in \mathcal{V}_I^{(s)}} \frac{d(a, b)}{2} \right\rfloor.$$

Furthermore, compute the maximum infection radius

$$r_{\max} = \max_{s=1,\ldots,S} r_s.$$

5: Consider the tree $\mathcal{T}$ formed by the set of nodes in $\mathcal{B}$ and paths between each two nodes in $\mathcal{B}$ on graph G. For each $e_i \in \mathcal{B}$, find a node γ_i on tree $\mathcal{T}$ such that $d(e_i, \gamma_i) = r_{\max}$, and add node γ_i into $\tilde{\mathcal{S}}$.

6: $\tilde{\mathcal{S}}$ is the set of source estimators.

In the first step of Algorithm 1, we select a pair of infected nodes with the maximum distance because most likely these two nodes are associated with two different information sources, where we say an infected node a is "associated" with source s if node a is on the information spreading tree starting from node s. The second step of the algorithm is to select S infected nodes in a greedy fashion to maximize the pairwise distances of these S nodes. These S infected nodes are likely to be associated with different sources, and are likely to be the leaf nodes of the corresponding information spreading trees. The third step divides the set of infected nodes into S sets according to their distances to the selected S nodes. The purpose is to cluster the infected nodes according to their associated sources. The fourth step estimates the maximum distance $r_{\max}$ from a source to any observed infected node associated with the source, which can be used to approximate the depth of the information spreading trees. In the final step, a tree $\mathcal{T}$ with nodes in $\mathcal{B}$ as the leaf nodes is constructed; and for each $e_i \in \mathcal{B}$, we select a node $\gamma_i \in \mathcal{T}$ that is $r_{\max}$ hops away from e_i as the corresponding source estimator. Note that γ_i is close to the real source associated with infected node e_i when e_i is close to a leaf node on the corresponding information spreading tree and $r_{\max}$ is close to the depth of the tree.

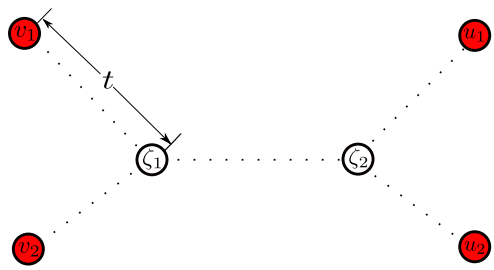


Fig. 2. An simple example of our algorithm.

Next, we present an example to illustrate how our algorithm works. Assuming the probability of infection $q = 1$ and the probability of recovery $p = 1$, which means each infected node, right after it is infected, will infect all of its susceptible neighbors and then become recovered at the end of the time slot. Given a simple tree structure as in Fig. 2, we use the dotted lines to represent the paths. There are two original sources, ζ_1 and ζ_2. The snapshot includes four infected nodes, v_1, v_2, u_1 and u_2, and we have $d(\zeta_1, v_1) = d(\zeta_1, v_2) = t$ and $d(\zeta_2, u_1) = d(\zeta_2, u_2) = t$. Under our algorithm, a pair of infected nodes that are farthest away from each other are selected, which could be $\{v_1, u_1\}$, $\{v_1, u_2\}$, $\{v_2, u_1\}$ and $\{v_2, u_2\}$. Without loss of generality, assume the two nodes are v_1 and u_1. Then after step 3, we have $\mathcal{V}_I^{(1)} = \{v_1, v_2\}$ and $\mathcal{V}_I^{(2)} = \{u_1, u_2\}$. Then, we have the maximum infection radius $r_{\max}$ is exactly equal to t and the tree $\mathcal{T}$ formed by the infected nodes and paths between them is the original tree $G(\mathcal{V}, \mathcal{E})$. Therefore, in step 5, the two estimators, say γ_1 and γ_2, on tree $\mathcal{T}$ that satisfy $d(v_1, \gamma_1) = t$ and $d(u_1, \gamma_2) = t$ are the sources ζ_1 and ζ_2.

3.2 Performance Analysis

The following theorem shows that for a g-regular tree, the distance between a detected source produced by Algorithm 1 and its closest real source is bounded by a constant with a high probability, where the constant is independent of the size of the infected subnetwork.

Theorem 1. *Consider a $(g+1)$-regular tree with infinite number of levels where $g > 2$. Assume that $gq > 1$ and the distance between any two sources is larger than C for some large enough constant C. Then given any $\epsilon > 0$, there exists a constant d_ϵ such that the distance between each estimator and its closest real source is upper bounded by d_ϵ with a probability at least $1 - \epsilon$, where d_ϵ is independent of the size of the infected subnetwork.*

The detailed proof is presented in Section 5, which consists of the following key steps:

1) We define one-time-slot branching process to be an infection spreading tree such that each infected node on the tree was infected in the immediate next time slot after the infection of the node's parent. A one-time-slot branching process is a subsequence of the infection process where an infected node is included

in the one-time-slot branching process if and only if it was infected at the immediate next time slot after her parent was infected. Because of that, the radius of the one-time-slot branching process increases by one in every time step until it terminates. Then for each source ζ_i, we define event $\mathcal{A}_{\zeta_i}$ which includes two cases: Case 1: the source has at least $(S+1)$ one-time-slot branching processes survived after time t_0. This means there exist $(S+1)$ survived one-time-slot branching processes whose roots are nodes that were infected before or at time slot t_0, where a one-time-slot branching process starting from an infected node is said to survive if it never dies out, which occurs with a non-zero probability. Case 2, the infection process from the source terminates at time t_0. We will prove that event $\mathcal{A} = \bigcap_i \mathcal{A}_{\zeta_i}$ occurs with a high probability.

2) The next step is to show that under event $\mathcal{A}$, each estimator produced by the algorithm is within a constant distance to its closest original source. The analysis includes the following three cases:
   (a) Infection processes from all original sources die out at time t_0.
   (b) At least two sources have survived $(S+1)$ one-time-slot branching processes.
   (c) Only one source have survived $(S+1)$ one-time-slot branching processes after time t_0.

In case (a), since infection processes from all original sources die out at time t_0, the maximum distance between a source and its associated infected nodes is t_0. Since any two sources are sufficiently far away from each other, the infected nodes in each set $\mathcal{V}_I^{(i)}$ are associated with the same source. Then in step 4, $r_{\max} \leq t_0$, which means the distance between each estimator found in step 5 and its closest original source is no larger than $2t_0$, which further means the distance between each estimator and its closest source is bounded by a constant. For case (b) and case (c), the idea is to study the leaf-nodes of the survived one-time-slot branching processes. The distance between the leaf-nodes of two one-time-slot branching processes from the same source is at least $2t - 2t_0$. Note that each survived source has at least $(S+1)$ one-time-slot branching processes. For simplicity, assume the nodes in set $\mathcal{B}$ after step 2 are the leaf-nodes of one-time-slot branching processes (This may not be true in general and we will discuss the general case in the proof). Then after step 3, there are at least two leaf-nodes of one-time-slot branching processes from the same source in $\mathcal{V}_I^{(i)}$, which implies $t - t_0 \leq r_{\max} \leq t$. Then the distance between the estimator and its closest survived source is equal to or smaller than $r_{\max} - (t - t_0) + t_0$, which is smaller than $2t_0$.

We can finally conclude that under event $\mathcal{A}$, the distance between each estimator and its closest original source is bounded by a constant, so Theorem 1 holds.

3.3 Heuristic for General Network Topologies

Locating multiple information sources in general networks is a much more complicated problem. Algorithm 1 is not directly applicable to a general network because after obtaining set $\mathcal{B}$, multiple trees can be constructed with the nodes in $\mathcal{B}$ as leaf nodes. Therefore, we propose the following heuristic algorithm, which use the Jordan infection center defined by $\mathcal{V}_I^{(s)}$ as the estimator associated with e_s.

In Algorithm 2, if the distance between any two original sources is sufficiently large, it is likely that the infected nodes in each set $\mathcal{V}_I^{(s)}$ are associated with the same source. The reverse infection algorithm was proposed in [8] to localize the source in the single-source SIR model. If the infected nodes in $\mathcal{V}_I^{(s)}$ are associated with the same source, we can expect the reserve infection algorithm restricted to $\mathcal{V}_I^{(s)}$ to output a good estimator. Therefore, in Algorithm 2, we use the reverse infection algorithm in each set $\mathcal{V}_I^{(s)}$ to get our estimators.

Algorithm 2. Clustering and Reverse Infection (CRI)

1: Step 1 to 3 of Algorithm 1.
2: For $\mathcal{V}_I^{(s)}$, use the reverse infection algorithm in [8] to find a Jordan infection center for $\mathcal{V}_I^{(s)}$, named γ_s, and then add γ_s to $\tilde{\mathcal{S}}$. A Jordan infection center γ_s for $\mathcal{V}_I^{(s)}$ is defined to be

$$\gamma_s \in \arg\min_{b \in \mathcal{V}} \left(\max_{a \in \mathcal{V}_I^{(s)}} d(b,a) \right).$$

3.4 Heuristic for Estimating S

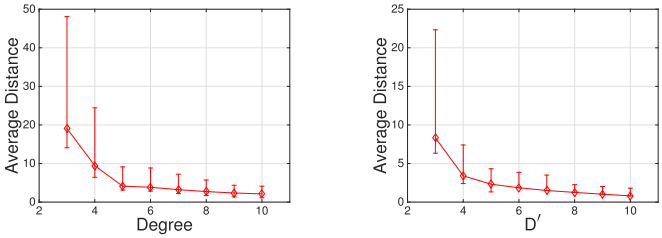
Both Algorithms 1 and 2 require the knowledge of the number of real sources, which may be difficult to know in practice. We further propose the following heuristic for estimating the number of real sources when the number of real sources is unknown.

Assume the information spreading trees never die out. Consider the case where k is smaller than the number of real sources. Then after the clustering step of Algorithm 1, there exists a set $\mathcal{V}_I^{(s)}$ which contains infected nodes from at least two different real sources. In such a set, the maximum distance between two nodes will be significantly larger than the distance of two infected nodes that are associated with the same source, assuming the real sources are not close to each other. Then under Algorithm 2, we will observe a significant decrease in w_k when the value of k changes from $S-1$ to S. Based on this observation, we use k that maximizes

$$w_k - w_{k+1} - (w_{k+1} - w_{k+2})$$

as the estimator of S.

Theorem 1 was established assuming an infinite regular tree network, in which case, the diameter of the network is infinite and the infection process can never reach the "edge" of the network. Therefore, if the time when the snapshot is taken is large enough, we can utilize the survived one-time-slot branching processes to identify the sources. However, in reality, networks are of finite size, which means the infection processes from different sources may hit the "edge" of the network and get completely "mixed" when the spreading time is large enough. Therefore, in practice, the algorithm may not perform well when the infection time is close to the diameter of the network. In fact, from Table 3 and Fig. 6, the normalized average distance becomes larger as the decrease of the diameter of the small-world network.



(a) Binomial random trees. (b) Regular trees.

Fig. 3. The average distance from sources to the estimators with 25 and 75 percentile on tree networks.

4 Simulations

In this section, we evaluate the performance of our algorithm using simulations.

4.1 Tree Networks

In this set of simulations, we assumed the number of real sources is known. We evaluated Algorithm 1 on g-regular trees and binomial random trees, in which the number of children of each node follows a binomial distribution with number of trials, D', and success probability β. In this simulation, we choose 4 original sources randomly and set $\beta = 0.6$, the probability of infection, q, is uniformly chosen from $(0, 0.3)$ and the probability of recovery, p, is uniformly chosen from $(0, 0.2)$.

In Figs. 3a and 3b, we plotted the average distance between real sources and the estimators versus the degree of the trees. To calculate the distance between an estimator and its related source, we maintain an estimator list, which contains all estimators, and a source list, which contains all sources. Then, we select the (estimator, source) pair with the smallest distance between them among all possible pairs formed by nodes from these two lists. Then we assign the estimator to the source in the pair we have selected. Next, we remove the source and the corresponding estimator from the source list and estimator list. The previous three steps are repeated until the estimator list or the source list is empty. After that, we can use the distance between the two nodes in each selected pair to calculate the average distance.

From Figs. 3a and 3b, we can see that as the degree of the tree becomes larger, the performance of our algorithm improves. For regular trees, the average distance is smaller than 3 when the degree is 5 or larger; and for binomial random trees, the average distance is smaller than 4 when D' is 5 or larger.

In Fig. 4, we plotted the detection rate of Algorithm 1, which is the fraction of estimators being real sources. As the degree becomes bigger, detection rates of both regular trees and binomial trees improves.

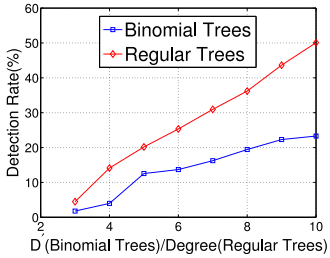


Fig. 4. True detection rate on binomial trees and regular trees.

TABLE 2
Diameters for Random Graphs with Different p

p	0.001	0.0015	0.002	0.0025
Diameter	26	17	11	10

4.2 General Networks

In this set of simulations, we tested the performance of our algorithm on the Erdős-Rényi (ER) model [16] and the small-world network model proposed in [17]. We compared our algorithm with random guessing and a heuristic algorithm based on k-means clustering. In k-means clustering, the initial centroids are randomly chosen. During the clustering step of each iteration in the k-means heuristic, we used distance centrality to select the centroid of each cluster. We assumed that the number of sources is unknown so we used Algorithm 3 to estimate the number of sources.

Algorithm 3. An Algorithm for Approximating S

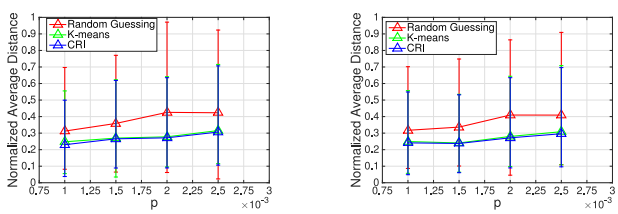
1: Choose a large number $\bar{S}$. For each k such that $1 \leq k \leq \bar{S}$, use the steps 1-4 in Algorithm 1 to compute $w_k = r_{\max}$ by assuming the number of real sources is k.
2: Set

$$\tilde{S} = \arg \max_{k:1 \leq k \leq \bar{S}-2} w_k - w_{k+1} - (w_{k+1} - w_{k+2}),$$

and claim $\tilde{S}$ to be the number of real sources.

4.2.1 The ER Random Graph

The ER random graphs generated in this section contains 2,000 nodes with wiring probability p, i.e., every pair of nodes is connected with probability p. We varied p to generate graphs with different diameters to test the algorithms. The diameters of the random graphs with different values of p are listed in Table 2. In Fig. 5, we used the normalized average distance between the estimator and its associated original source, which is the average distance divided by the diameter of the network, to measure the performance. From Figs. 5a and 5b, we can see that both the CRI algorithm and k-means algorithm performs much better than the random guessing algorithm, while the CRI algorithm outperforms k-means. As p becomes larger, which means the number of edges becomes larger, the normalized average distances of all these algorithms increase, which means these algorithms perform better in sparse networks than dense networks in terms of the normalized average distance.



(a) The number of original sources is 4. (b) The number of original sources is 5.

Fig. 5. The average distance between estimators and original sources versus random graph parameter probability p with 25 and 75 percentile.

TABLE 3
Diameters for the Small-World Networks with Different q

q	0	1	2	3
Diameter	98	26	19	15

4.2.2 The Small-World Network

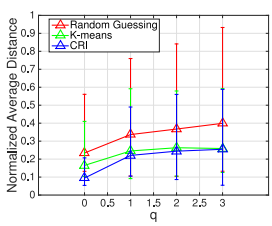
The small-world model proposed in [17] is based on a two-dimensional $n \times n$ grid, where the nodes are the lattice points. There are three parameters in this model, p, q, and r. For each node in the grid, it has an edge to every other node within lattice distance p. Then for each node u, q edges between u and other nodes are constructed. For example, the ith edge from u has endpoint v with probability proportional to $(d(u,v))^{-r}$. When we generated the network, we chose 50×50 grid with $p = 1$ and $r = 3$, while q was varied from 0 to 3. The diameters of these networks used in section are listed in Table 3. The results are shown in Figs. 6a and 6b, we can see that the CRI algorithm outperforms k-means algorithm and random guessing algorithm. As q increases, the normalized average distances increase under all three algorithms, similar to those in the ER random graphs.

5 Proof of Theorem 1

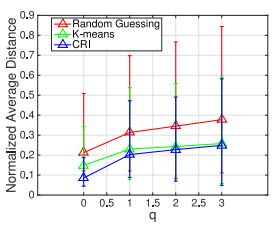
Consider a $(g+1)$-regular tree $G(\mathcal{V}, \mathcal{E})$ with S different information sources, named $\zeta_1, \zeta_2, \ldots, \zeta_S$. These S original sources and the paths between each pair form a tree, named $G_s = (\mathcal{V}_{G_s}, \mathcal{E}_{G_s})$. Define event $\mathcal{A} = \bigcap_{i=1}^{S} \mathcal{A}_{\zeta_i}$, where $\mathcal{A}_{\zeta_i} (i = 1, \ldots, S)$ is the event that includes the following cases:

- Case 1: At least $(S+1)$ one-time-slot branching processes from source ζ_i survive after time t_0, where these $(S+1)$ one-time-slot branching processes do not overlap.
- Case 2: The infection spreading tree starting from ζ_i terminates at or before time t_0. We describe this as the infection process from source ζ_i dies out at time t_0 on tree G.

We remark that t_0 is a constant. The one-time-slot branching process is defined to be the process that starting from an infected node, at each time slot, at least one node at the the next level, $l+1$, gets infected by an infected node from the current level l. The level of a node is defined to be the distance between that node and the starting node of the process, while the current level is defined to be the largest level among all infected nodes associated with the starting node of the one-time-slot branching process at the beginning of the current time slot. In Fig. 7, an example is used to explain the definition of the one-time-slot branching process.

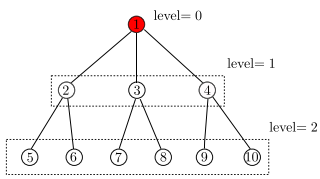


(a) The number of original sources is 4.

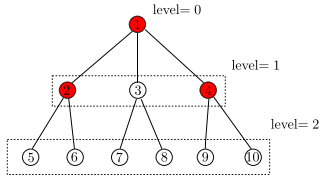


(b) The number of original sources is 5.

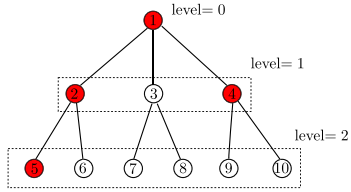
Fig. 6. The average distance between estimators and original sources versus q with 25 and 75 percentile.



(a) Time t = 0.



(b) Time t = 1.



(c) Time t = 2.

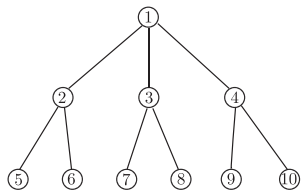
Fig. 7. An example of one-time-slot branching process starting from node 1. For simplicity, we assume the time when node 1 gets infected is 0. From Fig. 7b, at $t = 1$, the current level is 0 and there are two nodes, 2 and 4, from level 1 infected by node 1 from level 0. Fig. 7c shows that at $t = 2$, node 5 from next level 2 is infected by node 2 from the current level 1. Therefore, at each time slot, there is at least one node infected at level, $l+1$, by infected node from level l.

Then, we define a node set

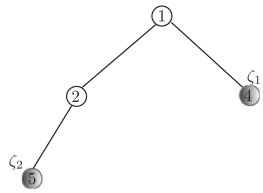
$$\mathcal{V}_\alpha = \{\alpha |\ \alpha \in \mathcal{V}_{G_s} \text{ and } \min_{i=1,\ldots,S} d(\alpha, \zeta_i) \leqslant C_1\},$$

in which each node is the on tree G_s and within C_1 ($C_1 > 0$ and $C_1 \in \mathbb{N}$) distance from at least one source, and $\mathcal{V}_\beta = \mathcal{V}_{G_s} \backslash \mathcal{V}_\alpha$, in which each node is on the tree G_s and in a distance larger than C_1 from any source. Define set $\mathcal{V}_S = \{\zeta_1, \zeta_2, \ldots, \zeta_S\}$ to be the set of original sources and we have $\mathcal{V}_S \subset \mathcal{V}_\alpha$. After that, define $m = |\mathcal{V}_\alpha|$ and $n = |\mathcal{V}_\beta|$. Without loss of generality, we assume $\mathcal{V}_\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ and define $\mathcal{V}_\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$.

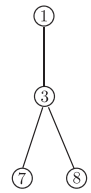
Define T_a to be a subtree on G, which starts from root a without edges on G_s. To better understand the definition of tree G_s and T_a, an example is provided in Fig. 8. We further



(a) Tree G.



(b) Tree G_s.



(c) Tree T_1.

Fig. 8. An example of tree G, tree G_s and tree T_1. In this example, the original tree is the tree in Fig. 8a. Assume the original sources are $\zeta_1 = 4$ and $\zeta_2 = 5$. Then tree G_s formed by the orginal sources and paths between each pair of them is shown in Fig. 8b. And Tree T_1, which starts from root 1 without edges on G_s, is shown in Fig. 8c.

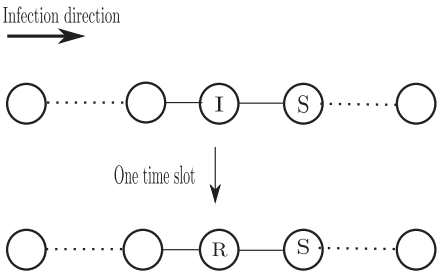


Fig. 9. The situation of the stop of infection process.

define $\mathcal{A}_a$ to be the event that on the tree T_a there are at least $(S+1)$ one-time-slot branching processes survived after time t_0 or all infection processes die out at time t_0. Then we have

$$\begin{aligned} &P(\mathcal{A}) \\ &= P\left(\bigcap_{i=1}^{S}\mathcal{A}_{\zeta_i}\right) \\ &\geq P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}) \\ &= \sum_{\mathbf{t}^I,\mathbf{t}^R} P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)P(\mathbf{t}^I,\mathbf{t}^R) \\ &> \sum_{(\mathbf{t}^I,\mathbf{t}^R)\in M_1} P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)P(\mathbf{t}^I,\mathbf{t}^R), \end{aligned} \tag{3}$$

where

$$\begin{aligned} \mathbf{t}^I &= (t^I_{\alpha_1},\dots,t^I_{\alpha_m},t^I_{\beta_1},\dots,t^I_{\beta_n}), \\ \mathbf{t}^R &= (t^R_{\alpha_1},\dots,t^R_{\alpha_m},t^R_{\beta_1},\dots,t^R_{\beta_n}) \end{aligned}$$

and $M_1 = \{(\mathbf{t}^I,\mathbf{t}^R)|\ \forall a\in\mathcal{V}_{G_s}, t^I_a \leq C_1 \text{ or } t^I_a=\infty\}$. Define event $\tilde{\mathcal{A}} = \bigcap_{a\in V_{G_s}}\mathcal{A}_a \bigcap\{(\mathbf{t}^I,\mathbf{t}^R)\in M_1\}$, which is the event of $\mathcal{A}$ restricted to M_1.

Lemma 2. *For any $\epsilon>0$, there exist some constants C_1 and t_0 such that $P(\tilde{\mathcal{A}})>1-\epsilon$, if the distance between any two sources is larger than $2C_1$.*

Proof. According to the definition of $\tilde{\mathcal{A}}$, we have

$$P(\tilde{\mathcal{A}}) = \sum_{(\mathbf{t}^I,\mathbf{t}^R)\in M_1} P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)P(\mathbf{t}^I,\mathbf{t}^R). \tag{4}$$

To obtain a lower bound on $P(\tilde{\mathcal{A}})$, we need to analyze $P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)$ when $(\mathbf{t}^I,\mathbf{t}^R)\in M_1$ and $P((\mathbf{t}^I,\mathbf{t}^R)\in M_1)$ separately.

For $P((\mathbf{t}^I,\mathbf{t}^R)\in M_1)$, we have

$$P((\mathbf{t}^I,\mathbf{t}^R)\in M_1) = 1-P((\mathbf{t}^I,\mathbf{t}^R)\in M_1^c), \tag{5}$$

where M_1^c is the complementary set of M_1. Define event $\mathcal{C}=\{(\mathbf{t}^I,\mathbf{t}^R)|\ \exists\beta\in\mathcal{V}_\beta, t^I_\beta>0 \text{ and } t^I_\beta\neq\infty\}$. Since the probability of event $\mathcal{M}$ is difficult to analyze directly, we define $\mathcal{M}_1 = M_1^c\cap\mathcal{C}$ and $\mathcal{M}_2 = M_1^c\cap\mathcal{C}^c$, i.e.,

$$P(M_1^c) = P(\mathcal{M}_1)+P(\mathcal{M}_2). \tag{6}$$

Before discussing $P(M_1^c)$, we analyze when the infection process on a path is going to stop. From Fig. 9, we know that during each time slot the infection process on a path stops only when the recently infected node is recovered before the node next to it becomes infected. The probability for this event to happen is $p(1-q)$. Use $(\zeta_j\to a)$ to represent the event that node a is associated with ζ_j and $a_{\{t=k,\zeta_j\}}$ to represent the event that infection process from source ζ_j to a does not stop at time slot k. Define $p_s = 1-p(1-q)$ and we have

$$\begin{aligned} &P(t^I_a>C_1 \text{ and } t^I_a\neq\infty|\zeta_j\to a) \\ &\leq P(a_{\{t=1,\zeta_j\}}\cap a_{\{t=2,\zeta_j\}}\cap\dots\cap a_{\{t=C_1,\zeta_j\}}|\zeta_j\to a) \\ &= P(a_{\{t=1,\zeta_j\}}|\zeta_j\to a)P(a_{\{t=2,\zeta_j\}}|a_{\{t=1,\zeta_j\}},\zeta_j\to a)\dots \\ &\quad P(a_{\{t=C_1,\zeta_j\}}|a_{\{t=C_1-1,\zeta_j\}},\zeta_j\to a) \\ &= (p_s)^{C_1}. \end{aligned} \tag{7}$$

Therefore, we have

$$\begin{aligned} &P(t^I_a>C_1 \text{ and } t^I_a\neq\infty) \\ &= \sum_{i=1}^{S} P(t^I_a>C_1 \text{ and } t^I_a\neq\infty|\zeta_i\to a)P(\zeta_i\to a) \\ &\leq \sum_{i=1}^{S}(p_s)^{C_1}P(\zeta_i\to a) \\ &\leq (p_s)^{C_1}. \end{aligned} \tag{8}$$

Next, we analyze $P(\mathcal{M}_1)$. Define set

$$\mathcal{V}_b = \{b|b\in\mathcal{V}_\beta \text{ and } \exists j,\ \text{s.t. } d(\zeta_j,b)=C_1+1\},$$

and for any $\beta\in\mathcal{V}_\beta$, since $d(\beta,\zeta_i)>C_1$ for any $i=1,\dots,S$, we have $t^I_\beta>C_1$ as long as $t^I_\beta\neq\infty$. Thus, $\mathcal{C}\subset M_1^c$, and we have

$$\begin{aligned} P(\mathcal{M}_1) &= P(M_1^c\cap\mathcal{C}) \\ &= P(\mathcal{C}) \\ &\overset{(a)}{=} P\Big(\bigcup_{b\in\mathcal{V}_b}\{t^I_b>C_1 \text{ and } t^I_b\neq\infty\}\Big) \\ &\leq \sum_{b\in\mathcal{V}_b}P(t^I_b>C_1 \text{ and } t^I_b\neq\infty) \\ &\leq |\mathcal{V}_b|(p_s)^{C_1} \\ &\leq S(S-1)(p_s)^{C_1}, \end{aligned} \tag{9}$$

where (a) holds because $\beta\in\mathcal{V}_\beta$ infected by the information from source ζ_i along the path that contains node b and satisfies $d(b,\zeta_i)=C_1+1$.

For $P(\mathcal{M}_2)$, we have

$$\begin{aligned} P(\mathcal{M}_2) &= P(M_1^c\cap\mathcal{C}^c) \\ &< P(\{(\mathbf{t}^I,\mathbf{t}^R)|\exists\alpha\in\mathcal{V}_\alpha,\ t^I_\alpha>C_1 \text{ and } t^I_\alpha\neq\infty\}) \\ &= P\Big(\bigcup_{\alpha\in\mathcal{V}_\alpha}\{t^I_\alpha>C_1 \text{ and } t^I_\alpha\neq\infty\}\Big) \\ &\leq \sum_{\alpha\in\mathcal{V}_\alpha}P(t^I_\alpha>C_1 \text{ and } t^I_\alpha\neq\infty) \\ &\overset{(a)}{<} \sum_{\alpha\in\mathcal{V}_\alpha}(p_s)^{C_1} \\ &= m(p_s)^{C_1}, \end{aligned} \tag{10}$$

where (a) comes from (8). Based on inequalities (9) and (10), we conclude

$$\begin{aligned}
&P((\mathbf{t}^I,\mathbf{t}^R)\in M_1)\\
&=1-P((\mathbf{t}^I,\mathbf{t}^R)\in M_1^c)\\
&=1-P(\mathcal{M}_1)-P(\mathcal{M}_2)\\
&\quad>1-m(p_s)^{C_1}-S(S-1)(p_s)^{C_1}\\
&\quad\overset{(a)}{>}1-S(S-1)C_1(p_s)^{C_1}-S(S-1)(p_s)^{C_1}\\
&=1-S(S-1)(C_1+1)(p_s)^{C_1},
\end{aligned}\tag{11}$$

where (a) holds because of $m\leq S(S-1)C_1$. Then we can choose a constant C_1 such that for any $\epsilon_1>0$,

$$P((\mathbf{t}^I,\mathbf{t}^R)\in M_1)>1-\epsilon_1.\tag{12}$$

Now, we need to discuss $P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)$ when $(\mathbf{t}^I,\mathbf{t}^R)\in M_1$. In this case, we have

$$\begin{aligned}
&P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)\\
&\quad=\prod_{\alpha\in\mathcal{V}_\alpha}P(\mathcal{A}_\alpha|t_\alpha^I,t_\alpha^R)\prod_{\beta\in V_\beta}P(\mathcal{A}_\beta|t_\beta^I,t_\beta^R).
\end{aligned}\tag{13}$$

According to the definition of $\mathcal{V}_\beta$, we have for any $\beta\in V_\beta$ and $1\leq i\leq S$, $d(\beta,\zeta_i)>C_1$, which implies that for any $(\mathbf{t}^I,\mathbf{t}^R)\in M_1$ and any $\beta\in\mathcal{V}_\beta$, we have $t_\beta^I=\infty$. So $P(\mathcal{A}_\beta|t_\beta^I,t_\beta^R)=1$.

For $P(\mathcal{A}_\alpha|t_\alpha^I,t_\alpha^R)$ when $\alpha\in\mathcal{V}_\alpha$, we define $u_1,\dots,u_k$ to be the child nodes of root α on the tree T_α when $\alpha\in\mathcal{V}_\alpha$ and $\mathcal{A}_{u_i}$ $(i=1,\dots,k)$ to be the event that on the tree $T_{u_i}^{-\alpha}$ ($T_{u_i}^{-\alpha}$ is a subtree of T_α rooted at u_i but without the branch from α), there are at least $(S+1)$ one-time-slot branching processes survived after time t_0 or all infection processes die out before or at time t_0. On a $(g+1)$-regular tree, we have $k\leqslant g$. Then we have

$$\begin{aligned}
P(\mathcal{A}_\alpha|t_\alpha^I,t_\alpha^R)&>P(\mathcal{A}_{u_1}\dots\mathcal{A}_{u_k}|t_\alpha^I,t_\alpha^R)\\
&=\sum_{\mathbf{t}_u^I}P(\mathcal{A}_{u_1}\dots\mathcal{A}_{u_k}|\mathbf{t}_u^I,t_\alpha^I,t_a^R)P(\mathbf{t}_u^I|t_\alpha^I,t_\alpha^R)\\
&>\sum_{\mathbf{t}_u^I\in M_2}P(\mathcal{A}_{u_1}\dots\mathcal{A}_{u_k}|\mathbf{t}_u^I)P(\mathbf{t}_u^I|t_\alpha^I,t_\alpha^R),
\end{aligned}\tag{14}$$

where $\mathbf{t}_u^I=\{t_{u_1}^I,\dots,t_{u_k}^I\}$ and $M_2=\{\vec{t_u^I}|$ for $i=1,\dots,k$, $t_{u_i}^I-t_\alpha^I\leq C_2,$ or $t_{u_i}^I=\infty\}$, and $C_2\in\mathbb{N}$ is a constant.

To compute $P(\mathbf{t}_u^I\in M_2|t_\alpha^I,t_\alpha^R)$, we consider the following three cases:

1) When $t_\alpha^I=\infty$, we have $t_{u_i}^I=\infty$ for $i=1,\dots,k$, which means $P(\mathbf{t}_u^I\in M_2|t_\alpha^I,t_\alpha^R)=1$.
2) When $t_\alpha^R-t_\alpha^I\leq C_2$ and $t_\alpha^I\neq\infty$, we always have $t_{u_i}^I-t_\alpha^I\leq C_2$ or $t_{u_i}^I=\infty$ for $i=1,..,k$, which means $\mathbf{t}_u^I\in M_2$. Thus, we have $P(\mathbf{t}_u^I\in M_2|t_\alpha^I,t_\alpha^R)=1$.
3) When $t_\alpha^R-t_\alpha^I>C_2$ and $t_\alpha^I\neq\infty$, we have

$$\begin{aligned}
&P(\mathbf{t}_u^I\in M_2|t_\alpha^I,t_\alpha^R)\\
&=\prod_{i=1}^k P(t_{u_i}^I-t_\alpha^I\leq C_2\text{ or }t_{u_i}^I=\infty|t_\alpha^I,t_\alpha^R)\\
&=\prod_{i=1}^k\left(\sum_{t_{u_i}^I=1+t_\alpha^I}^{C_2+t_\alpha^I}q(1-q)^{t_{u_i}^I-t_\alpha^I-1}+(1-q)^{t_\alpha^R-t_\alpha^I}\right)\\
&=\prod_{i=1}^k(1-(1-q)^{C_2}+(1-q)^{t_\alpha^R-t_\alpha^I})\\
&\quad>\prod_{i=1}^k(1-(1-q)^{C_2})\\
&=(1-(1-q)^{C_2})^k.
\end{aligned}\tag{15}$$

In summary, we always have

$$P(\mathbf{t}_u^I\in M_2|t_\alpha^I,t_\alpha^R)>(1-(1-q)^{C_2})^k.\tag{16}$$

We also have

$$P(\mathcal{A}_{u_1}\dots\mathcal{A}_{u_k}|\mathbf{t}_u^I,t_\alpha^I,t_\alpha^R)=\prod_{i=1}^k P(\mathcal{A}_{u_i}|t_{u_i}^I)\overset{(a)}{>}(1-\epsilon_2)^k,\tag{17}$$

where (a) can be proved by following the proof of Lemma 6 in [8].

By substituting (16) and (17) into (14), we have

$$\begin{aligned}
&P(\mathcal{A}_\alpha|t_\alpha^I,t_\alpha^R)\\
&>\sum_{\mathbf{t}_u^I\in M_2}P(\mathcal{A}_{u_1}\dots\mathcal{A}_{u_k}|\mathbf{t}_u^I,t_\alpha^I,t_\alpha^R)P(\mathbf{t}_u^I|t_\alpha^I,t_\alpha^R)\\
&>(1-\epsilon_2)^k(1-(1-q)^{C_2})^k.
\end{aligned}\tag{18}$$

Since $k\leq g$, we can choose a constant C_2 such that for any $\epsilon_3>0$, $(1-(1-q)^{C_2})^k>1-\epsilon_3$. Then we have

$$P(\mathcal{A}_\alpha|t_\alpha^I,t_\alpha^R)>(1-\epsilon_2)^k(1-\epsilon_3),$$

and

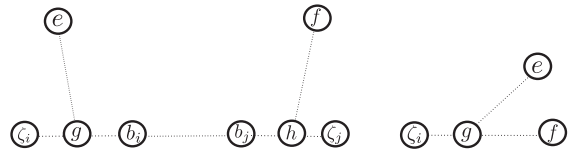
$$\begin{aligned}
P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)&>\prod_{\alpha\in\mathcal{V}_\alpha}(1-\epsilon_2)^k(1-\epsilon_3)\\
&=((1-\epsilon_2)^k(1-\epsilon_3))^m.
\end{aligned}\tag{19}$$

Substituting (19) and (12) into (3), we have

$$\begin{aligned}
P(\mathcal{A})&>P(\tilde{\mathcal{A}})\\
&=\sum_{\mathbf{t}^I,\mathbf{t}^R\in M_1}P(\mathcal{A}_{\alpha_1}\dots\mathcal{A}_{\alpha_m}\mathcal{A}_{\beta_1}\dots\mathcal{A}_{\beta_n}|\mathbf{t}^I,\mathbf{t}^R)P(\mathbf{t}^I,\mathbf{t}^R)\\
&>((1-\epsilon_2)^k(1-\epsilon_3))^m(1-\epsilon_1)\\
&>1-\epsilon.
\end{aligned}\tag{20}$$

□

Assuming that the time when the snapshot is taken, t, satisfies $t\gg t_0$ and $t\gg d_{i,j}$ for any $i,j=1,\dots,S$, where $d_{i,j}$ is defined to be $d(\zeta_i,\zeta_j)$, and $d>3St_0+4SC_1$, where $d=\min_{1\leq i,j\leq S}d_{i,j}$. We need to prove that the distance between each estimator and its closest original source is



(a) e and f are infected by two sources.

(b) e and f are infected by the same source.

Fig. 10. In Fig. 10a, we consider the infection process of two sources, ζ_i and ζ_j. Assume b_i and b_j are the nodes on path (ζ_i, ζ_j) that satisfy $d(b_i, \zeta_i) = C_1$ and $d(b_j, \zeta_j) = C_1$. Under event $\tilde{\mathcal{A}}$, on path (ζ_i, ζ_j) only nodes on (ζ_i, b_i) and (ζ_j, b_j) can be infected. If e and f are endpoints of survived one-time-slot branching processes, we have $d(\zeta_i, e) \geq t - t_0$ and $d(\zeta_j, f) \geq t - t_0$. Therefore, $d(e, f) \geq d(\zeta_i, \zeta_j) - d(\zeta_i, b_i) - d(\zeta_j, b_j) + d(\zeta_i, e) - d(\zeta_i, b_i) + d(\zeta_j, f) - d(\zeta_j, b_j)$, which means $d(e, f) \geq d(\zeta_i, \zeta_j) + 2(t - t_0) - 4C_1$. In Fig. 10b, we consider the situation that e and f are infected by the same source, ζ_i. If e and f are endpoints of survived one-time-slot branching process, we have $d(e, f) = d(\zeta_i, e) + d(\zeta_i, f) - 2d(\zeta_i, g)$. Since the survived one-time-slot branching processes do not overlap after t_0, we have $d(\zeta_i, g) \leq t_0$. Therefore, we have $d(e, f) \geq 2(t - t_0) - 2t_0 = 2t - 4t_0$.

bounded by some constants under event $\tilde{\mathcal{A}}$. We say a source ζ_i has infected nodes if there are some infected nodes associated with ζ_i when the snapshot was taken. Since in the following proof, we will use the distance between two infected nodes in the graph frequently, here in Fig. 10, we discuss some cases of distances between two infected nodes commonly used later.

We divide event $\tilde{\mathcal{A}}$ into three subevents:

1) Infection processes from all original sources die out at time t_0.

There are two cases under this condition:

- The number of infected nodes is less than or equal to S. All those infected nodes will be treated as our estimators according to Algorithm 1. Because there is no node getting infected after time t_0, each of these nodes is within the distance of t_0 from the source that infects it. Therefore, the distance between each estimator and its closest source is bounded by t_0, which is a constant.
- The number of infected nodes is greater than S.

Claim 1. *After step 2, the set $\mathcal{B}$ contains infected nodes associated with all sources who have infected nodes when the snapshot is taken.*

Proof of Claim 1. Assume set $\mathcal{B}$ does not contain any infected node associated with some source ζ_k that has infected nodes observed in the snapshot, which means $\mathcal{B}$ contains at least two nodes associated with same source. Assume $a, b \in \mathcal{B}$ are associated with source ζ_j and $c \notin \mathcal{B}$ to be an infected node associated with ζ_k.

Then we have $d(a, b) \leq 2t_0$. For any $e \in \mathcal{B}$, we have

$$\begin{aligned} d(e, c) &\geq d - 2C_1 \\ &> 3St_0 + 4SC_1 - 2C_1 \\ &> 2t_0 \\ &> d(a, b), \end{aligned} \tag{21}$$

which contradicts the step 2 in Algorithm 1.

Claim 2. *After step 3, in each set $\mathcal{V}_I^{(i)}$ $(i = 1, \ldots, S)$, the nodes are infected by the same source.*

Proof of Claim 2. Without loss of generality, we assume $\mathcal{B} = \{e_1, \ldots, e_S\}$ and $e_i \in \mathcal{V}_I^{(i)}$. Assume node a in set $\mathcal{V}_I^{(i)}$ is associated with a source which is different from the source that e_i is associated with. Then we have

$$\begin{aligned} d(a, e_i) &\overset{(a)}{\geq} d - 2C_1 \\ &> 3St_0 + (4S - 2)C_1, \end{aligned} \tag{22}$$

where (a) is true because of the definition of event $\tilde{\mathcal{A}}$ and set M_1. According to Claim 1, set $\mathcal{B}$ contains nodes associated with the sources that have infected nodes associated with them in the snapshot. Therefore, we can assume in another set $\mathcal{V}_I^{(j)}$, node e_j is associated with the same source as node a. Then we have $d(e_j, a) < 2t_0$. Therefore, we have $d(e_j, a) < d(e_i, a)$, which is in contradiction with step 3. Thus, Claim 2 holds.

According to Claim 2, in each set $\mathcal{V}_I^{(i)}$, all the nodes are infected by the same source. Therefore, the maximum infection radius $r_{\max}$ after step 4 should satisfy that $r_{\max} \leq t_0$. Assume γ_i to be the estimator generated by $\mathcal{V}_I^{(i)}$ that contains e_i and ζ_i to be the actual source who infected the nodes in $\mathcal{V}_I^{(i)}$. Then we have $d(\gamma_i, e_i) = r_{max}$ and $d(e_i, \zeta_i) \leq t_0$. Therefore, $d(\gamma_i, \zeta_i) \leq 2t_0$, which means the distance between each estimator and its closest source is bounded by $2t_0$, which is a constant.

2) There are at least two sources surviving $(S + 1)$ one-time-slot branching processes after time t_0.

Assume the number of sources who survive at least $(S + 1)$ one-time-slot branching processes is n_0. And we have $n_0 \geq 2$. We will then prove the following conclusions:

a) $e_1, e_2, \ldots, e_{n_0}$ are infected by different sources that survive at least $(S + 1)$ one-time-slot branching processes after t_0.

b) For any $i \geq 2$,

$$d(e_i, \hat{\zeta}_i) \geq t - it_0 - 4(i - 1)C_1, \tag{23}$$

and when $i = 1$,

$$d(e_1, \hat{\zeta}_1) \geq t - 2t_0 - 4C_1, \tag{24}$$

where $\hat{\zeta}_i$ represents the source that e_i is associated with.

c) For any $1 \leq i, j \leq n_0$, we have

$$d(e_i, e_j) > 2t. \tag{25}$$

Proof of the Conclusions.

- $e_1, e_2 (i = 1, 2)$: Assume η_1 and η_2 are two leaf-nodes of one-time-slot processes associated with two different sources. Since for any $a, b \in \mathcal{V}_I$, $d(e_1, e_2) \geq d(a, b)$, we have

$$d(e_1,e_2) \geq d(\eta_1,\eta_2) \overset{(a)}{\geq} d + 2(t-t_0) - 4C_1 \; > \; 2t,$$

where (a) is true because of Fig. 10a. Thus, e_1 and e_2 have to be infected by two different sources. Without loss of generality, we assume ζ_1 and ζ_2 are the two sources who infect e_1 and e_2. Then we have

$$\begin{aligned} d(e_1,\zeta_1) + d_{1,2} + d(e_2,\zeta_2) &\geq d(e_1,e_2) \\ &\geq d_{1,2} + 2(t-t_0) - 4C_1, \end{aligned} \tag{26}$$

where $d_{i,j} = d(\zeta_i,\zeta_j)$ for any $i,j = 1,\ldots,S$ and because $d(e_1,\zeta_1) \leq t$ and $d(e_2,\zeta_2) \leq t$, we have $d(e_1,\zeta_1) \geq t - 2t_0 - 4C_1$ and $d(e_2,\zeta_2) \geq t - 2t_0 - 4C_1$. Because $t \gg t_0$ and $t \gg C_1$, we may consider $d(e_1,\zeta_1) > t_0$ and $d(e_2,\zeta_2) > t_0$, which implies ζ_1 and ζ_2 must be two surviving sources. Therefore, e_1 and e_2 satisfy these three conclusions.

- $e_k (2 \leq k \leq n_0 - 1)$: Assume that e_k satisfies the three conclusions, which means that e_k is infected by another source ζ_k, who survives at least $(S+1)$ one-time-slot branching processes at time t_0, $d(e_i,\zeta_k) \geq t - kt_0 - 4(k-1)C_1$, and for any $1 \leq i,j \leq k$, $d(e_i,e_j) > 2t$.
- e_{k+1}: If e_{k+1} is infected by $\zeta_1,\ldots,\zeta_k$ or any source that dies out at time t_0, we have $\min_{i=1}^{k} d(e_i,e_{k+1}) \leq 2t$. If e_{k+1} is infected by another source ζ_{k+1}, who survives at least $(S+1)$ one-time-slot branching processes at time t_0, assume η_{k+1} is the leaf-node of one survived one-time-slot branching process infected by ζ_{k+1} and we have

$$\begin{aligned} d(\eta_{k+1},e_1) &\geq (t - 2t_0 - 4C_1) + d_{1,k+1} + t - t_0 - 4C_1 \\ &\overset{(a)}{>} 2t, \end{aligned} \tag{27}$$

where (a) holds according to $d_{1,k+1} \geq d > 3St_0 + 4SC_1$, and

$$\begin{aligned} d(\eta_{k+1},e_i) &\geq (t - it_0 - 4(i-1)C_1) \\ &\quad + d_{i,k+1} + t - t_0 - 4C_1 \\ &> 2t, \end{aligned} \tag{28}$$

for $i = 2,\ldots,k$. Thus, we have

$$\begin{aligned} d(e_{k+1},e_i) &\geq \min_{i=1}^{k} d(\eta_k,e_i) \\ &> 2t. \end{aligned} \tag{29}$$

Therefore, we know that when e_{k+1} is infected by ζ_{k+1}, we have $\min_{i=1}^{k} d(e_{k+1},e_i) > 2t$. Then, e_{k+1} has to be infected by another source who survives at least $(S+1)$ one-time-slot branching processes at time t_0 according to Algorithm 1.

Assuming $j = \arg\min_{i\in\{1,\ldots,k\}} d(\eta_{k+1},e_i)$, we have

$$\begin{aligned} &d(e_{k+1},\zeta_{k+1}) + d_{j,k+1} + d(e_j,\zeta_j) \\ &\geq d(e_{k+1},e_j) \\ &\geq d(\eta_{k+1},e_j) \\ &\geq d(e_j,\zeta_j) + d_{j,k+1} + t - t_0 - 4C_1 \\ &\geq t - kt_0 - 4(k-1)C_1 + d_{j,k+1} - 4C_1 + t - t_0 \\ &= 2t - (k+1)t_0 - 4kC_1 + d_{j,k+1}. \end{aligned} \tag{30}$$

Because $d(e_j,\zeta_j) \leq t$, we have $d(e_{k+1},\zeta_{k+1}) \geq t - (k+1)_0 - 4kC_1$. Therefore, e_{k+1} satisfies the three conclusions mentioned before.

Then we complete the proof the conclusions.

Since $e_1,\ldots,e_{n_0}$ are infected by different sources who survive at least $(S+1)$ one-time-slot branching processes, we can assume $e_i (i = 1,\ldots,n_0)$ is infected by ζ_i, while ζ_i survives at least $(S+1)$ one-time-slot branching processes after time t_0. According to inequalities (23), (24) and (25), we have

$$d(e_i,\zeta_i) \geq t - n_0t_0 - 4(n_0-1)C_1 \tag{31}$$

and for any $1 \leq i,j \leq n_0$, $d(e_i,e_j) > 2t$.

Define $\hat{\zeta}_i$ to be the source whom e_i $(i = n_0+1,\ldots,S)$ is associated with. Then we will prove the following conclusion:

a) For $e_i (i = n_0+1,\ldots,S)$, we have

$$d(e_i,\hat{\zeta}_i) \geq t - (3i - 2n_0)t_0 - 4(n_0-1)C_1, \tag{32}$$

where $\hat{\zeta}_i$ is one of $\zeta_1,\ldots,\zeta_{n_0}$.

b) For $j = 1,\ldots,i-1$, we have

$$d(e_i,e_j) \geq 2t - (3i - 2n_0)t_0 - 4(n_0-1)C_1. \tag{33}$$

Proof of the Conclusions.

- e_{n_0+1}: If e_{n_0+1} is infected by a source ζ_{n_0+1}, whose infection process dies out at time t_0, for any $i = 1,\ldots,n_0$, we have $d(e_{n_0+1},e_i) \leq d_{n_0+1,i} + t + t_0$, which means

$$\min_{i\in\{1,\ldots,n_0\}} d(e_{n_0+1},e_i) \leq d' + t + t_0, \tag{34}$$

where $d' = \min_{i\in\{1,\ldots,n_0\}} d_{n_0+1,i}$.

If e_{n_0+1} is infected by ζ_j, where $j = 1,2,\ldots,n_0$, we have

$$\begin{aligned} &d(e_{n_0+1},e_i) \\ &\geq d(e_i,\zeta_i) + d_{i,j} - 4C_1 + d(e_{n_0+1},\zeta_j) \\ &\geq t - n_0t_0 + 4(n_0-1)C_1 + d_{i,j} - 4C_1 + d(e_{n_0+1},\zeta_j) \\ &= t - n_0t_0 - 4n_0C_1 + d_{i,j} + d(e_{n_0+1},\zeta_j) \\ &> t + d(e_{n_0+1},\zeta_j), \end{aligned} \tag{35}$$

where $i \neq j$, $i = 1,\ldots,n_0$, and

$$d(e_{n_0+1},e_j) \leq t + d(e_{n_0+1},\zeta_j). \tag{36}$$

Therefore, assuming η_{n_0+1} is the leaf-node of a survived one-time-slot branching process generated by ζ_j, we have

$$\min_{i\in\{1,\ldots,n_0\}} d(e_{n_0+1},e_i) = d(e_{n_0+1},e_j) \geq d(\eta_{n_0+1},e_j). \tag{37}$$

Since every surviving source has at least $(S+1)$ one-time-slot branching processes survived after time t_0, we can always find a leaf-node ζ_{n_0+1}, whose one-time-slot branching process doesn't overlap with path (e_{n_0+1},ζ_j). Therefore, we have

$$\begin{aligned} d(e_{n_0+1},e_j) &\geq d(\eta_{n_0+1},e_j) \\ &\geq d(e_j,\zeta_j)+d(\eta_{n_0+1},\zeta_j)-2t_0 \\ &\geq t-n_0t_0-4(n_0-1)C_1+t-t_0-2t_0 \\ &= 2t-(n_0+3)t_0-4(n_0-1), \end{aligned} \tag{38}$$

which is bigger than $d'+t+t_0$ of (34) according to $t \gg t_0$ and $t \gg d_{i,j}$ for any $i,j=1,\ldots,S$. This means e_{n_0+1} has to be infected by a source that survives at least $(S+1)$ one-time-slot branching processes at time t_0.

According to (35), (36) and (38), we have $d(e_{n_0+1},\ e_i) \geq 2t-(n_0+3)t_0-4(n_0-1),$ where $i=1,\ldots,n_0$. Because $d(e_{n_0+1},\zeta_j)+d(e_j,\zeta_j) \geq d(e_{n_0+1},e_j)$ and $d(e_j,\zeta_j)\leq t,$ we have

$$d(e_{n_0+1},\zeta_j) \geq t-(n_0+3)t_0-4(n_0-1)C_1.$$

Therefore, the conclusions holds for e_{n_0+1}.

- $e_k (n_0+1 \leq k < S)$: Assume that it's infected by one of $\zeta_i (i=1,\ldots,n_0)$ and we have

$$d(e_k,\hat{\zeta}_k) \geq t-(3k-2n_0)t_0-4(n_0-1)C_1 \tag{39}$$

and for $i=1,\ldots,k-1$,

$$d(e_k,e_i) \geq 2t-(3k-2n_0)t_0-4(n_0-1)C_1. \tag{40}$$

- e_{k+1}: If e_{k+1} is infected by a source ζ_{k+1}, whose infection process dies out at time t_0, for any $i=1,\ldots,k$, we have $d(e_{k+1},e_i) \leq d(\zeta_{k+1},\hat{\zeta}_i)+t+t_0,$ which means

$$\min_{i\in\{1,\ldots,k\}} d(e_{k+1},e_i) \leq d''+t+t_0, \tag{41}$$

where $d''=\min_{i=\{1,\ldots,k\}} d(\zeta_{k+1},\hat{\zeta}_i)$.

According to the assumption for e_k and inequality (31), we have

$$d(e_i,\hat{\zeta}_i) \geq t-(3k-2n_0)t_0-4(n_0-1)C_1, \tag{42}$$

for $i\in\{1,\ldots,k\}$.

If e_{k+1} is associated with ζ_j, where $j\in\{1,\ldots,n_0\}$, for $\zeta_j\neq\hat{\zeta}_i$, we have

$$\begin{aligned} d(e_{k+1},e_i) &\geq d(e_{k+1},\zeta_j)+d(\hat{\zeta}_i,e_i)+d(\zeta_j,\hat{\zeta}_i)-4C_1 \\ &\geq t-(3k-2n_0)t_0-4(n_0-1)C_1 \\ &\quad +d(\zeta_j,\hat{\zeta}_i)+d(e_{k+1},\zeta_j)-4C_1 \\ &> t+d(e_{k+1},\zeta_j). \end{aligned} \tag{43}$$

When $\zeta_j=\hat{\zeta}_i$, we have $d(e_{k+1},e_i)\leq t+d(e_{k+1},\zeta_j)$. Assume $e'=\text{argmin}_{\{e_1,\ldots,e_k\}} d(e_{k+1},e_i)$, where e' is one of $e_1,\ldots,e_k$ and is associated with ζ_j, and η_{k+1} is the leaf-node of a survived one-time-slot branching process infected by ζ_j. Similarly, we assume $e''=\arg\min_{e_i} d(\eta_{k+1},e_i),$ where e'' is one of $e_1,\ldots,e_k$ and is associated with ζ_j.

According to Algorithm 1, we have

$$\min_{i=1}^{k} d(e_{k+1},e_i) = d(e_{k+1},e') \geq d(\eta_{k+1},e''). \tag{44}$$

Since every survived source has at least $(S+1)$ one-time-slot branching processes survived at time t_0, we can always find an leaf-node η_{k+1}, whose one-time-slot branching process doesn't overlap with (e_{k+1},ζ_j). Therefore, we have

$$\begin{aligned} &d(e_{k+1},e') \\ &\geq d(\eta_{k+1},e'') \\ &\geq d(\eta_{k+1},\zeta_j)+d(e'',\zeta_j)-2t_0 \\ &\geq t-t_0+t-(3k-2n_0)t_0-4(n_0-1)C_1-2t_0 \\ &= 2t-(3k+3-2n_0)t_0-4(n_0-1)C_1, \end{aligned} \tag{45}$$

which is bigger than $d''+t+t_0$ of (41) because of $t\gg t_0$ and $t\gg d_{i,j}$ for any $i,j=1,\ldots,k+1$. This means e_{k+1} has to be infected by a surviving source. Since for $i=1,\ldots,k$, the inequalities

$$d(e_{k+1},e_i) \geq 2t-(3k+3-2n_0)t_0-4(n_0-1)C_1,$$
$$d(e_{k+1},\zeta_j)+d(e',\zeta_j) \geq d(e_{k+1},e'),$$

and $d(e',\zeta_j)\leq t$, hold, we have

$$d(e_{k+1},\zeta_j) \geq t-(3k+3-2n_0)n_0t_0 - 4(n_0-1)C_1, \tag{46}$$

which satisfies the condition $d(e_i,\hat{\zeta}_i)\geq t-(3k+32n_0)t_0-4(n_0-1)C_1,$ when $i=k+1$.

Then we complete the proof of the conclusions.

According to inequalities (31) and (42), we have that for $i=1,\ldots,n_0$, $d(e_i,\hat{\zeta}_i)\geq t-it_0-4(i-1)C_1$ and for $i=n_0+1,\ldots,S$, $d(e_i,\hat{\zeta}_i)\geq t-(3i-2n_0)t_0-4(n_0-1)C_1$ $(i=n_0+1,\ldots,S)$. Thus, when $n_0\geq 2$, if we set $C_3=(3S-2n_0)t_0+4(n_0-1)C_1,$ then we have $d(e_i,\hat{\zeta}_i)\geq t-C_3$. And for $1\leq i,j\leq S$, we have

$$d(e_i,e_j) \geq 2t-(3S-2n_0)t_0-4(n_0-1)C_1, \tag{47}$$

which means $d(e_i,e_j)\geq 2t-C_3$.

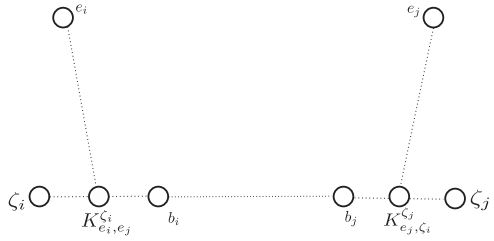


Fig. 11. The positions of ζ_i, ζ_j, e_i and e_j when e_i and e_j are associated with different sources, where $d(b_i, \zeta_i) = C_1$ and $d(b_j, \zeta_j) = C_1$.

Claim 3. *If $d > 3St_0 + 4SC_1$ and there are at least two sources having $(S+1)$ one-time-slot branching processes survived, the leaf-nodes of all survived one-time-slot branching processes in the same set $\mathcal{V}_I^{(i)}$ $(i = 1, \ldots, S)$ are associated with the same source that e_i is associated with.*

Proof of Claim 3. Assume $e_i \in \mathcal{V}_I^{(i)}$ is associated with source ζ_i and $a \in \mathcal{V}_I^{(i)}$ is the leaf-node of an survived one-time-slot branching process generated by another source ζ_j, which means in set $\mathcal{V}_I^{(i)}$, e_i and a are associated with different sources.

Assume $e_j \in \mathcal{V}_I^{(i)}$ is associated with source ζ_j. Then we have

$$\begin{aligned} d(a, e_i) &\geq d(e_i, \zeta_i) + d(a, \zeta_j) + d_{i,j} - 4C_1 \\ &\geq (t - C_3) + (t - t_0) + d_{i,j} - 4C_1. \\ &= 2t + d_{i,j} - C_3 - t_0 - 4C_1. \end{aligned} \tag{48}$$

Because $C_3 = (3S - 2n_0)t_0 + 4(n_0 - 1)C_1$ and $d_{i,j} > 3St_0 + 4SC_1$, we have

$$\begin{aligned} &d_{i,j} - C_3 - t_0 - 4C_1 \\ &\geq (2n_0 - 1)t_0 + 4(S - n_0)C_1 \\ &> 0, \end{aligned} \tag{49}$$

which means $d(a, e_i) > 2t$. However, we have $d(a, e_j) \leq 2t$, which means $d(a, e_i) > d(a, e_j)$, which is in contradiction to step 3. This completes the proof Claim 3.

Next, we need to prove that the distance between each estimator and its closest source is bounded by a constant. Since there are S clusters and each surviving source has at least $(S+1)$ one-time-slot branching processes, there is at least one cluster containing two leaf-nodes of one-time-slot branching processes from one source. Then in step 4, the infection radius r_i of set $\mathcal{V}_I^{(i)}$ that contains at least two leaf-nodes of survived one-time-slot branching processes from a single source satisfies that $t - t_0 \leqslant r_i \leqslant t$. Then $r_{\max} = \max\{r_i\}$ in step 4 should satisfy that $t - t_0 \leqslant r_{\max} \leqslant t$.

Then, we need to prove that, in step 5, on the tree $\mathcal{T}$, the node, which is in distance $r_{\max}$ from e_i, is in a constant distance to the surviving original source with which e_i is associated. Define node h_i to be the node on path $(\hat{\zeta}_i, e_i)$ that satisfies $d(\hat{\zeta}_i, h_i) = C_3$. Then we the following claim.

Claim 4. *Node h_i is on the tree $\mathcal{T}$ for all $i = 1, \ldots, S$.*

Proof of Claim 4. There are two cases:

a) $e_i, e_j \in \mathcal{B}$ are associated with the same source:
Without loss of generality, we assume e_i and e_j are associated with ζ_i. Then we know that $K^{\zeta_i}_{e_i,e_j}$ (the definition is in Table 1) is on the path (e_i, e_j) and we have

$$d(e_i, e_j) = d(e_i, \zeta_i) + d(e_j, \zeta_i) - 2d(\zeta_i, K^{\zeta_i}_{e_i,e_j}) \tag{50}$$

and

$$d(e_i, e_j) \geq 2t - C_3, \tag{51}$$

which means

$$\begin{aligned} 2d(\zeta_i, K^{\zeta_i}_{e_i,e_j}) &\leq d(e_i, \zeta_i) + d(e_j, \zeta_i) - 2t + C_3 \\ &\leq C_3. \end{aligned} \tag{52}$$

Therefore, we have

$$d(\zeta_i, K^{\zeta_i}_{e_i,e_j}) \leq C_3/2. \tag{53}$$

b) $e_i, e_j \in \mathcal{B}$ are associated with different sources:
Without loss of generality, we assume e_i is associated with ζ_i, while e_j is associated with ζ_j. Fig. 11 is the description of the relations among nodes e_i, e_j, ζ_i and ζ_j. According to the definition, both $K^{\zeta_i}_{e_i,e_j}$ and $(K^{\zeta_j}_{e_j,\zeta_i}, e_j)$ are on the path (ζ_i, e_j). More precisely, $K^{\zeta_i}_{e_i,e_j}$ is on the path $(\zeta_i, K^{\zeta_j}_{e_j,\zeta_i})$. Since path $(\zeta_i, K^{\zeta_j}_{e_j,\zeta_i})$ is part of path (ζ_i, ζ_j), which means $K^{\zeta_i}_{e_i,e_j}$ is on path (ζ_i, ζ_j). Because $K^{\zeta_i}_{e_i,e_j}$ is associated with ζ_i, under event $\tilde{\mathcal{A}}$, we have $d(\zeta_i, K^{\zeta_i}_{e_i,e_j}) \leq C_1$

Therefore, for all $e_i, e_j \in \mathcal{B}$, we have

$$\begin{aligned} d(\hat{\zeta}_i, K^{\hat{\zeta}_i}_{e_i,e_j}) &\leq \max\{C_3/2, C_1\} \\ &= C_3/2. \end{aligned} \tag{54}$$

Then for each e_i, if we choose h_i which satisfies $h_i \in (\hat{\zeta}_i, e_i)$ and $d(h_i, \hat{\zeta}_i) = C_3$. We have $h_i \in (K^{\hat{\zeta}_i}_{e_i,e_j}, e_i)$. Since path $(K^{\hat{\zeta}_i}_{e_i,e_j}, e_i)$ is on the tree $\mathcal{T}$, h_i is on the tree $\mathcal{T}$. This completes the proof of Claim 4.

Therefore, we have $d(e_i, \hat{\zeta}_i) = d(e_i, h_i) + d(h_i, \hat{\zeta}_i)$ and $d(\hat{\zeta}_i, h_i) = C_3$. Assume γ_i to be the estimator to $\hat{\zeta}_i$ generated by e_i. Because $r_{\max} \geq t - t_0$, $(e_i, h_i) \subset (e_i, e_j)$ for all $e_j \in \mathcal{B}$ and

$$\begin{aligned} d(e_i, h_i) &= d(e_i, \hat{\zeta}_i) - d(h_i, \hat{\zeta}_i) \\ &\leq t - C_3 \\ &< t - t_0 \\ &\leq r_{\max}, \end{aligned} \tag{55}$$

we have $r_{\max} = d(e_i, \gamma_i) = d(e_i, h_i) + d(h_i, \gamma_i)$.

Therefore,

$$\begin{aligned} d(\gamma_i, \hat{\zeta}_i) &\leqslant d(\gamma_i, h_i) + d(\hat{\zeta}_i, h_i) \\ &= d(e_i, \gamma_i) - d(e_i, h_i) + d(\hat{\zeta}_i, h_i) \\ &= d(e_i, \gamma_i) - (d(e_i, \hat{\zeta}_i) - d(h_i, \hat{\zeta}_i)) + d(\hat{\zeta}_i, h_i) \\ &\leq t - (t - C_3 - C_3) + C_3 \\ &\leq 3C_3. \end{aligned} \tag{56}$$

Finally, we have $d(\gamma_i, \hat{\zeta}_i) \leqslant 3C_3$, which means the estimator we find is in a constant distance with the surviving original source.

3) There is only one source surviving $(S+1)$ one-time-slot branching processes after time t_0.

Assuming $\hat{\zeta}_i$ to be the source that infects e_i, we will prove the following conclusions:

a) For $i = 1, \ldots, S$, $\hat{\zeta}_i$ must be the same source that survives at least $(S+1)$ one-time-slot branching processes at time t_0.

b) For e_1, we have

$$d(e_1, \hat{\zeta}_1) \geq t - 4t_0. \tag{57}$$

c) For $e_i (i = 2, \ldots, S)$, we have

$$d(e_i, \hat{\zeta}_i) \geq t - (3i-2)t_0. \tag{58}$$

d) For any $i = 1, \ldots, S$ and $j = 1, \ldots, i-1$, we have

$$d(e_i, e_j) \geq 2t - (3i-2)t_0. \tag{59}$$

Proof of the conclusions. At first we assume that the only source who survives at least $(S+1)$ one-time-slot branching processes at time t_0 is ζ_j.

- e_1, e_2: If e_1 and e_2 are associated with two other sources, let's say ζ_{i_1} and ζ_{i_2}, except ζ_j, we have $d(e_1, e_2) \leq 2t_0 + d_{i_1,i_2}$.

  If e_1 and e_2 are associated with the same source except ζ_j, we have $d(e_1, e_2) \leq 2t_0$.

  If e_1 and e_2 are associated with ζ_j, assuming η_1 and η_2 are two leaf-nodes of survived one-time-slot branching processes of ζ_j, we have

$$\begin{aligned} d(e_1, e_2) &\geq d(\eta_1, \eta_2) \\ &\geq d(\eta_1, \zeta_j) + d(\eta_2, \zeta_j) - 2t_0 \\ &\geq t - t_0 + t - t_0 - 2t_0 \\ &= 2t - 4t_0. \end{aligned} \tag{60}$$

  Because $t \gg t_0$, we know that when e_1 and e_2 are associated with the same source ζ_j, we can get $d(e_1, e_2)$ maximized. According to Algorithm 1, e_1 and e_2 are associated with ζ_j. Since $d(e_2, \zeta_j) \leq t$ and

$$\begin{aligned} d(e_1, \zeta_j) + d(e_2, \zeta_j) &\geq d(e_1, e_2) \\ &\geq 2t - 4t_0, \end{aligned} \tag{61}$$

  we have $d(e_1, \zeta_j) \geq t - 4t_0$. In a similar way, we have $d(e_2, \zeta_j) \geq t - 4t_0$.

  Therefore, these conclusions hold for e_1 and e_2.

- $e_k (2 < k \leq S-1)$: Assume that e_k is infected by ζ_j and it satisfies $d(e_k, \zeta_j) \geq t - (3k-2)t_0$ and $d(e_k, e_i) \geq 2t - (3k-2)t_0$, where $i = 1, \ldots, k-1$.
- $e_{k+1} (2 \leq k \leq S-1)$: If e_{k+1} is associated with another source $\zeta_{i_{k+1}}$, we have $d(e_{k+1}, e_i) \leq d(\zeta_{i_{k+1}}, \zeta_j) + t + t_0$.

If e_{k+1} is associated with ζ_j, assuming η_{k+1} is the leaf-node of a survived one-time-slot branching process of ζ_j who dose not overlap with (ζ_j, e_i) for $i = 1, \ldots, k$ after time t_0, we have

$$\begin{aligned} d(e_i, \eta_{k+1}) &\geq d(e_i, \zeta_j) + d(\eta_{k+1}, \zeta_j) - 2t_0 \\ &\geq t - (3i-2)t_0 + t - t_0 - 2t_0 \\ &\geq t - (3k-2)t_0 + t - t_0 - 2t_0 \\ &= 2t - (3k+1)t_0, \end{aligned} \tag{62}$$

which means for any $i = 1, \ldots, k$, we have

$$\begin{aligned} d(e_i, e_{k+1}) &\geq \min_{i=1}^{k} d(e_i, \eta_{k+1}) \\ &\geq 2t - (3k+1)t_0. \end{aligned} \tag{63}$$

Since we hypothesize $t \gg t_0$ and $t \gg d_{i,j}$ for any $i, j = 1, \ldots, k+1$, we have $2t - (3k+1)t_0 > d(\zeta_{i_{S+1}}, \zeta_j) + t + t_0$, which means e_{k+1} has to be associated with ζ_j.

Assuming

$$e' = \arg \min_{e_i, i=1,} d(e_i, e_{S+1}),$$

because $d(e', \zeta_j) \leq t$ and

$$\begin{aligned} d(e', \zeta_j) + d(e_{k+1}, \zeta_j) &\geq d(e', e_{k+1}) \\ &\geq 2t - (3k+1)t_0, \end{aligned} \tag{64}$$

we have $d(e_{k+1}, \zeta_j) \geq t - (3k+1)t_0$.

The we have finished the proof of these conclusions. Therefore, according to inequalities (57), (58) and (59), for any $i = 1, \ldots, S$, we have

$$d(e_i, \hat{\zeta}_i) \geq t - C_4, \tag{65}$$

$$d(e_i, e_j) \geq 2t - C_4, \forall 1 \leq i, j \leq S, \tag{66}$$

where we define $C_4 = (3S-2)t_0$, and $e_1, \ldots, e_S$ are infected by the only source who survives at least $(S+1)$ one-time-slot branching processes at time t_0.

Because there are S clusters and each surviving source has at least $(S+1)$ one-time-slot branching processes, we have at least one cluster containing two leaf-nodes of one-time-slot branching processes. Then in step 4, the infection radius r_i of the set $\mathcal{V}_I^{(i)}$, which contains at least two leaf-nodes of survived one-time-slot branching processes from a single source satisfies that $t - t_0 \leqslant r_i \leqslant t$. Then $r_{\max} = \max\{r_i\}$ in step 4 should also satisfy that $t - t_0 \leqslant r_{\max} \leqslant t$.

Next, we need to consider the tree $\mathcal{T}$, which is formed by nodes in $\mathcal{B}$ and paths between every two nodes in $\mathcal{B}$. According to the conclusions we have proved before, the nodes in $\mathcal{B}$ are associated with the same source. Without loss of generality, we assume that nodes in set $\mathcal{B}$ are infected by the source ζ_1. Then for any two nodes e_i, e_j in $\mathcal{B}$ we have

$$\begin{aligned} &d(e_i, e_j) \\ &= d(e_i, \zeta_1) + d(e_j, \zeta_1) - 2d(\zeta_1, K^{\zeta_1}_{e_i,e_j}) \\ &\geq 2t - C_4, \end{aligned} \tag{67}$$

which means

$$\begin{aligned} &2d(\zeta_1, K^{\zeta_1}_{e_i,e_j}) \\ &\leq d(e_i, \zeta_1) + d(e_j, \zeta_1) - 2t + C_4 \\ &\leq C_4. \end{aligned} \tag{68}$$

Therefore, we have $d(\zeta_1, K^{\zeta_1}_{e_i,e_j}) \leq C_4/2$.

According to the definition of $\mathcal{T}$, we know that path $(e_i, K^{\zeta_1}_{e_i,e_j})$ is on the tree $\mathcal{T}$. Therefore, for node e_i, if we define another node h_i, which is on the path (ζ_1, e_i) and satisfies $d(h_i, \zeta_1) = C_4$, h_i is on the tree $\mathcal{T}$. In a similar way, we can define h_j.

Thus, for each $e_i \in \mathcal{B}(i = 1, \ldots, S)$, we can define another node h_i which is on the path (ζ_1, e_i) and satisfies $d(\zeta_1, h_i) = C_4$ so that it is on the tree $\mathcal{T}$. And we have $d(e_i, \zeta_1) = d(e_i, h_i) + d(h_i, \zeta_1)$ and $d(\zeta_1, h_i) = C_4$.

Assume γ_i to be the estimator to ζ_1 generated by e_i. Because $r_{\max} \geq t - t_0$, we have

$$r_{\max} = d(e_i, \gamma_i) = d(e_i, h_i) + d(h_i, \gamma_i).$$

We have

$$\begin{aligned} &d(\gamma_i, \zeta_1) \\ &\leqslant d(\gamma_i, h_i) + d(\zeta_1, h_i) \\ &= d(e_i, \gamma_i) - d(e_i, h_i) + d(\zeta_1, h_i) \\ &= d(e_i, \gamma_i) - (d(e_i, \zeta_1) - d(h_i, \zeta_j)) + d(\zeta_1, h_i) \\ &\leq t - (t - C_4 - C_4) + C_4 \\ &\leq 3C_4. \end{aligned} \tag{69}$$

Finally, we have $d(\gamma_i, \zeta_1) \leq 3C_4$, which means the estimator we find is in a constant distance to the surviving original source.

6 Conclusion

In this paper, we studied the multi-source detection problem in the SIR model with the observation of states of nodes in the network. We provided an algorithm for tree network to detect multiple information sources when the number of sources is known. And we also proved that with a fairly general condition, each estimator is within a constant distance to its closest original source for tree networks, which can guarantee our algorithm. Then we proposed another algorithm for general networks and a heuristic algorithm to decide the number of sources, which make our CL and CRI algorithms more general and applicable. The simulation results showed that our algorithm performs well on multi-source detection problem.

Appendix A
Reverse Infection Algorithm [8]

The key idea of the reverse infection algorithm in [8] is to let each observed infected node to broadcast its identity (ID) to its neighbors. The set of nodes who first receive all IDs of the infected nodes are declared to be Jordan infection centers. The pseudocode is described in Algorithm 4.

Algorithm 4. Reverse Infection Algorithm

for $i \in \mathcal{V}_I$ **do**
  i sends its ID w_i to its neighbors.
end for
while $t \geq 1$ and STOP=0 **do**
  for $u \in \mathcal{V}$ **do**
    if u receives w_i for the first time **then**
      set $t_{ui} = t$, where t is the current time slot, and then broadcast the message w_i to its neighbors.
      if there exists a node who received $\mathcal{V}_I$ distinct messages **then**
        set STOP=1.
      end if
    end if
  end for
end while
return $u^\star = \arg\min_{s \in \mathcal{S}} \sum_{i \in \mathcal{V}_I} t_{ui}$, where $\mathcal{S}$ is the set of nodes who receive $\mathcal{V}_I$ distinct messages when the algorithm terminates. Ties are broken at random.

Acknowledgments

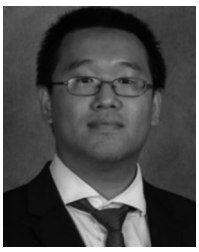
Research supported in part by ARO grant W911NF-13-1-0279.

References

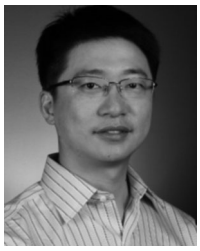
[1] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: Theory and experiment," in *Proc. ACM SIGMETRICS Int. Conf. Measure. Model. Comput. Syst.*, 2010, pp. 203–214.
[2] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, Aug. 2011.
[3] D. Shah and T. Zaman, "Rumor centrality: A universal source detector," in *Proc. 12th ACM SIGMETRICS/PERFORM. Joint Int. Conf. Measure. Model. Comput. Syst.*, 2012, pp. 199–210.
[4] W. Luo and W. P. Tay, "Identifying multiple infection sources in a network," in *Proc. 46th Asilomar Conf. Signals, Syst. Comput.*, 2012, pp. 1483–1489.
[5] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2850–2865, Jun. 2013.
[6] N. Karamchandani and M. Franceschetti, "Rumor source detection under probabilistic sampling," in *Proc. IEEE Int. Symp. Inform. Theory*, Jul. 2013, pp. 2184–2188.
[7] W. Dong, W. Zhang, and C. W. Tan, "Rooting out the rumor culprit from suspects," in *Proc. IEEE Int. Symp. Inform. Theory*, 2013, pp. 2671–2675.
[8] K. Zhu and L. Ying, "Information source detection in the SIR model: A sample path based approach," in *Proc. Inform. Theory Appl. Workshop*, Feb. 2013, pp. 1–9.
[9] K. Zhu and L. Ying, "A robust information source estimator with sparse observations," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr.–May 2014, pp. 2211–2219.
[10] W. Luo and W. P. Tay, "Finding an infection source under the SIS model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2930–2934.

[11] W. Luo and W. P. Tay, "Estimating infection sources in a network with incomplete observations," in *Proc. IEEE Global Conf. Signal Inform. Process.*, 2013, pp. 301–304.
[12] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 11–20.
[13] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, "Inferring the origin of an epidemy with dynamic message-passing algorithm," *Phys. Rev. E*, vol. 90, p. 012801, Jul. 2014.
[14] A. Agaskar and Y. M. Lu, "A fast Monte Carlo algorithm for source localization on graphs," in *Proc. SPIE Optical Eng. Appl.*, 2013, p. 88581N.
[15] E. Seo, P. Mohapatra, and T. Abdelzaher, "Identifying rumors and their sources in social networks," in *Proc. SPIE*, 2012, vol. 8389, pp. 83891I-1–83891I-13.
[16] A. Rényi and P. Erdős, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.
[17] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. Annu. ACM Symp. Theory Comput.*, 2000, pp. 163–170.

Zhen Chen received the BE degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2011. He is currently working toward the PhD degree at the School of Electrical, Computer and Energy Engineering, Arizona State University. His research interest is in social networks.



Kai Zhu received the BE degree in electronic engineering from Tsinghua University, Beijing, China, and the PhD degree in electrical engineering from Arizona State University. His research interest is in the areas of diffusion processes, random graphs, and social networks.



Lei Ying (M'08) received the BE degree from Tsinghua University, Beijing, China, and the MS and PhD degrees in electrical and computer engineering from the University of Illinois, Urbana-Champaign. He currently is an associate professor at the School of Electrical, Computer and Energy Engineering, Arizona State University, and an associate editor of the *IEEE/ACM Transactions on Networking*. His research interest is broadly in the area of stochastic networks, including cloud computing, communication networks, and social networks. He is coauthor with R. Srikant of the book *Communication Networks: An Optimization, Control and Stochastic Networks Perspective*, Cambridge University Press, 2014. The book has been selected as a notable book in the Computing Reviews' 19th Annual Best of Computing list. He received the Young Investigator Award from the Defense Threat Reduction Agency (DTRA) in 2009 and NSF CAREER Award in 2010. He was the Northrop Grumman assistant professor in the Department of Electrical and Computer Engineering, Iowa State University from 2010 to 2012. He received the Best Paper Award at the IEEE INFOCOM 2015. He is a member of the IEEE.