Identifying Infection Sources and Regions in Large Networks

Wuqiong Luo, *Student Member, IEEE*, Wee Peng Tay, *Member, IEEE* and Mei Leng, *Member, IEEE*

arXiv:1204.0354v1 [cs.DM] 2 Apr 2012

Abstract

Identifying the infection sources in a network, including the individuals who started a rumor in a social network, the computers that introduce a virus into a computer network, or the index cases of a contagious disease, plays a critical role in limiting the damage caused by the infection through timely quarantine of the sources. We consider the problem of estimating the infection sources and the infection regions (subsets of nodes infected by each source) in a network, based only on knowledge of the underlying network connections, and when the number of sources is unknown a priori. We derive estimators for the infection sources and their infection regions based on approximations of the infection sequences counts. We prove that if there are at most two infection sources in a geometric tree, our estimator identifies the true source or sources with probability going to one as the number of infected nodes increases. When there are more than two infection sources, and when the maximum possible number of infection sources is known, we propose an algorithm with quadratic complexity to estimate the actual number and identities of the infection sources. Simulations are conducted on various kinds of networks, including tree networks, small-world networks and real world power grid networks, to verify the performance of our algorithms. Our simulation results show that with high probability, our proposed estimators are within a few hops from the true infection sources.

Index Terms

Source estimation, infection graphs, inference algorithms, security, sensor networks, social networks.

I. Introduction

Online social networks have grown exponentially in complexity over the last few years. A modern social network like Twitter have millions of active users [1]. A rumor started by a few individuals can spread quickly through the network [2]–[8]. In many cases, we are interested in finding the sources of the rumor. For example, law enforcement agencies may be interested in identifying the perpetrators who fabricate false information to manipulate the market prices of certain stocks. In a similar vein, a computer virus on a few servers of a computer network can quickly spread to other servers or computers in the network. Without prompt identification and isolation of the source servers, significant damage can result [9], [10]. Identifying the servers in the network that are first infected also

This research was supported by the MOE AcRF Tier 1 Grant M52040000. W. Luo, W. P. Tay and M. Leng are with the Nanyang Technological university, Singapore. E-mail: `wluo1@e.ntu.edu.sg, {wptay,lengmei}@ntu.edu.sg`

allows us to detect the latent points of weaknesses in the computer network so that preventive measures can be taken to enhance the protection at these points. The source identification problem also arises in the study and control of viral epidemics. The identification of index cases of a contagious disease in a human population allows us to study the causes, and hence facilitate the search for antiviral drugs and efficacious therapies. Moreover, by inferring the infection region of each source, potential containments can be implemented to prevent further spreading of the disease [11], [12].

We can model all the above examples as an infection spreading in a network of nodes. In a social network, an infection can be a rumor or a piece of information that is communicated between individuals. In the example of a computer virus spreading in a network, the infection is the computer virus, while for the case of a disease spreading in a population, the infection is the disease, which is transmitted between individuals. We consider the problem of estimating the infection sources in a network of infected nodes. We are interested in the scenario where the only given information is the set of infected nodes and the underlying network connections. This is because typically, complete data about the infection spreading process, like the first times when the infection is detected at each node, is not available. Even when such detection times are available, the naive method of declaring the first detected node in the network as the sole infection source is often incorrect, as the infection may have a random dormant period, the length of which varies from node to node. For example, the spreading of a disease in a population with individuals having varying degrees of resistance, and hence exhibiting symptoms not necessarily in the order in which they are infected, presents such a problem. Our goal is to construct estimators for both the infection sources and their infection regions, i.e., the subset of nodes likely to be infected by each source, when the number and locations of the sources are unknown a priori.

A. Related Works

Existing works related to infection spreading in a social network have primarily focused on the identification of influential nodes in the network. Each node in a network has a probability of influencing or “infecting” its neighbors. The references [13]–[16] consider the problem of identifying a subset of nodes to maximize the total *expected* influence of the subset, where the expectation is taken over all possible realizations of the infection process. In this paper, we consider a related but different problem. Our aim is to identify a set of nodes most likely to be the infection sources, given a *particular* realization of the infection process. The case where there is a single infection source has been studied in [17]. Based only on the knowledge of which nodes are infected and the underlying network structure, an estimator based on the linear extensions count of a poset or number of infection sequences (cf. Section II) was derived in [17] to identify the most likely infection source. Although finding the infection sources is much easier than solving the influence maximization problem, which is NP-hard, it was shown in [17] that finding the most likely infection source in a general network is nevertheless a #P-complete problem. Therefore, a simplistic homogeneous diffusion model, where the infection from an infected node is equally likely to be transmitted to any of its neighbors at each time step, was adopted. The infection spreading model is based on the classical *susceptible-infected-recovered* (SIR) model [18], which has been widely used in modeling viral epidemics. An algorithm for evaluating the single source estimator was proposed in [17], and it was shown to have complexity[1] $O(n)$ for tree networks, where n is the total number of infected nodes. Furthermore, it was shown that this estimator performs well in a very general class of tree networks known as the geometric trees (cf. Section III-D), and identifies the infection source with probability going to one as n increases. In this paper, we generalize and extend the results in [17] to the cases where there may be multiple infection sources, and when the number of infection sources is unknown a priori. We also consider the problem of estimating the infection regions, and show that a direct application of the algorithm in [17] performs significantly worse than our proposed algorithms if there are more than one infection sources. We also note that [17] provides theoretical performance measures for several classes of tree networks, which we are unable to do here except for the class of geometric trees, because of the greater complexity of our proposed algorithms. Instead, we provide simulation results to verify the performance of our algorithms.

A related problem is the detection and localization of diffusive sources using wireless sensor networks [19]–[24]. The diffusion models used under this framework are based on spatio-temporal diffusion models [19] or state-space models with linear dynamics [20], where information like the physical positions of sensors are typically assumed. There is no natural translation of the source detection and localization problem in a sensor network to other networks like a social network, without performing discretization and introducing a combinatorial aspect to the problem, as is done in [13], [25]. Similarly, inference of viral epidemic processes in populations has been studied in [18], [26], [27], where various features related to the propagation of a viral epidemic, such as the rates of infection and the length of latency periods are investigated. These works' focus is on specific viral infection processes with assumptions that do not naturally hold for infection processes in other networks like a social network. Moreover, there is little work on determining the sources or index cases of a disease.

On the other hand, the infection source estimation algorithms we consider in this paper can be useful in applications like pollution source localization, where we are limited to inexpensive sensors capable only of detecting the presence or absence of a pollutant, and the identities of its neighbors. In this case, spatio-temporal diffusion models are not applicable as we only have knowledge of which nodes are "infected" and each node's neighbors. Moreover, the algorithms we study in this paper are also applicable to inferring infection sources in viral epidemics, when little information about the epidemic propagation characteristics is available.

B. Our Contributions

In this paper, we consider the estimation of multiple infection sources when the number of infection sources is unknown a priori. We adopt the same diffusion model as in [17], and show that unlike the single source estimation problem, the multiple source estimation problem is much more complex and cannot be solved exactly even for

[1] A function $f(n) = O(g(n))$ if $f(n) \leq cg(n)$ for some constant c and for all n sufficiently large.

regular trees. In addition, we derive an estimator to estimate the infection region of each infection source, i.e., the set of nodes infected by that source. Our main contributions are the following.

(i) For the case of a tree network, and when it is known that there are two infection sources, we derive an estimator for the infection sources based on the infection sequences count. The estimator can be calculated in $O(n^2)$ time complexity, where n is the number of infected nodes.

(ii) When there are at most two infection sources that are at least two hops apart, we derive an estimator for the class of geometric trees based on approximations of the estimator in (i), and we show that our estimator correctly estimates the number of infection sources and correctly identifies the source nodes, with probability going to one as the number of infected nodes increases.

(iii) We derive an estimator for the infection regions of every infection source under a simplifying technical condition.

(iv) For general graphs, when there are at most $k_{\max}$ infection sources, we provide an estimation procedure for the infection sources and infection regions. Simulation results show that our estimation procedure produces estimators that are within a few hops of the true infection sources with high probability.

The rest of the paper is organized as follows. In Section II, we present the system model and problem formulation. In Section III, we derive estimators for infection sources and regions for tree networks, and present algorithms to evaluate them. We also show asymptotic results for geometric tree networks. We discuss estimation algorithms for general graphs in Section IV. In Section V, we present simulation results to to verify the performance of our proposed estimators. Finally we conclude and summarize in Section VI.

II. Problem Formulation

In this section, we describe our model and assumptions, introduce some notations, and present some preliminary results. Consider an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. If there is an edge connecting two nodes, we say that they are neighbors. The neighborhood $\mathcal{N}(v)$ of a node v is the set of all neighbors of v. The length of the shortest path between u and v (excluding u and v) is denoted as $d(u, v)$. In a computer network, the graph G models the interconnections between computers in the network. In the example of a population or a social network, V is the set of individuals, while an edge in E represents a relationship between two individuals. We define an **infection** to be a property that a node in G possesses, and can be transmitted to another node. When a node has an infection, we say that it is infected. An infected node can pass its infection to its neighbors in the graph G. The neighbors of an infected node is said to be susceptible. We assume the susceptible-infected model [18], where once a node has been infected, it will not lose its infection. We adopt the same infection spreading process as in [17], where the time taken for an infected node to infect a susceptible neighbor is exponentially distributed with rate 1. All infections are independent of each other. Therefore, if a susceptible node has more than one infected neighbors and subsequently becomes infected, its infection is transmitted by one of its infected neighbors, chosen uniformly at random. For mathematical convenience, we also assume that G is large so that boundary effects can be ignored in our analysis.

Suppose that at time 0, there are $k \geq 1$ nodes in the infected node set $S = \{s_1, \ldots, s_k\} \subset V$. These are the **infection sources** from which all other nodes get infected. Suppose that after the infection process has run for some time, n nodes are observed to be infected, where n is typically much larger than k. These nodes form an **infection graph** $G_n = (V_n, E_n)$, which is a subgraph of G. Let $\mathcal{A}_n = \cup_{i=1}^{k} A_{n,i}$ be a partition of the infected nodes V_n so that $A_{n,i} \cap A_{n,j} = \emptyset$ for $i \neq j$, with each partition $A_{n,i}$ being connected in G_n, and consisting of the nodes whose infection can be traced back to the source node s_i. The set $A_{n,i}$ is called the **infection region** of s_i, and we say that $\mathcal{A}_n$ is the **infection partition**. Given G_n, our objective is to infer the sources of infection S and to estimate $\mathcal{A}_n$. In addition, if we do not have prior knowledge of the number of infection sources k, we also aim to infer the number of infection sources. Without loss of generality, we assume that G_n is connected, otherwise the same estimation procedure can be performed on each of the components of the graph. We also assume that there are at most $k_{\max}$ infection sources, i.e., the number of infection sources $k \leq k_{\max}$.

From a practical point of view, if two infection sources are close to each other, we can ignore either one of them and treat the infection as spreading from a single source. Therefore, we are interested in cases where the infection sources are separated by a minimum distance. We make the following assumption throughout this paper.

Assumption 1. *For all $s_i, s_j \in S$, the length of the shortest path between them $d(s_i, s_j) \geq \tau$, where τ is a constant greater than 1.*

Suppose that our priors for S and $\mathcal{A}_n$ are uniform over all possible realizations, and let $\mathbb{P}$ be the probability measure of the infection process. We seek to maximize the posterior probability

$$\mathbb{P}(S, \mathcal{A}_n \mid G_n) \propto \mathbb{P}(G_n \mid S)\mathbb{P}(\mathcal{A}_n \mid S, G_n). \tag{1}$$

Let an **infection sequence** $\sigma = (\sigma_1, \ldots, \sigma_{n-k})$ be a sequence of the nodes in G_n, excluding the sources S, arranged in ascending order of their infection times (note that with probability one, no two infection times are the same). For any sequence to be an infection sequence, a necessary and sufficient condition is that any infected node σ_i, $i = 1, \ldots, n-k$, has a neighbor in $S \cup \{\sigma_1, \ldots, \sigma_{i-1}\}$. We call this the *infection sequence property*. An example is shown in Figure 1. Let $\Omega(G_n, S)$ be the set of infection sequences for an infection graph G_n and source set S. We have

$$\mathbb{P}(G_n \mid S) = \sum_{\sigma \in \Omega(G_n, S)} \mathbb{P}(\sigma \mid S). \tag{2}$$

Evaluating the expression (2) and maximizing (1) for a general G_n is a computationally hard problem as it involves combinatorial quantities. As shown in [17], if G is a regular tree and $|S| = 1$, $\mathbb{P}(G_n \mid S)$ is proportional to $|\Omega(G_n, S)|$, which is equivalent to the number of linear extensions of a poset. It is known that evaluating the linear extensions count is a hard problem [28]. When there are multiple infection sources, the complexity is even

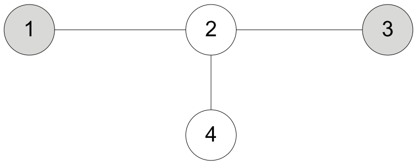


Fig. 1. Example of an infection sequence. The shaded nodes are the infection sources. The sequence $(2,4)$ is an infection sequence, but $(4,2)$ is not.

higher. As such, we will make a series of approximations to simplify the problem, and present numerical results in Section V to verify our algorithms. The first approximation we make is to evaluate the estimators

$$\hat{S} = \arg \max_{\substack{S \subset V_n \\ |S| \leq k_{\max}}} \mathbb{P}(G_n \mid S), \tag{3}$$

$$\hat{\mathcal{A}}_n(\hat{S}) = \arg\max_{\mathcal{A}} \mathbb{P}(\mathcal{A} \mid \hat{S}, G_n), \tag{4}$$

instead of the exact maximum a posteriori (MAP) estimators for (1). Even with this approximation, the optimal estimators are difficult to compute exactly, and may not be unique in general. Therefore, our goal is to design algorithms that are approximately optimal.

For any graph G, let $\deg_G(u)$ denote the number of neighbors of u in the graph G. For any infection sequence σ, and $u \in \sigma$, let $N_\sigma(u, G)$ be the number of infected neighbors of u in the graph G immediately after u becomes infected. For example, if $u = \sigma_j$, then $N_\sigma(u, G)$ is the number of nodes in $S \cup \{\sigma_1, \ldots, \sigma_{j-1}\}$ that are neighbors of u in G. If $u \in S$, then $N_\sigma(u, G)$ is the number of infection sources that are neighbors of u. For any infection sequence σ of the nodes in G, and for $l = 1, \ldots, |\sigma|$, where $|\sigma|$ is the number of infected nodes, let

$$p_l(\sigma \mid H, S) = \left(\sum_{s \in S} (\deg_H(s) - N_\sigma(s, H)) + \sum_{j=1}^{l-1} \deg_H(\sigma_j) - 2 \sum_{j=1}^{l-1} N_\sigma(\sigma_j, H) \right)^{-1}. \tag{5}$$

We have the following general characterizations for the conditional probabilities of interest in (3) and (4), whose proof is provided in Appendix A.

Lemma 1. *For any graph G, source node set S with $|S| = k$, and infection graph G_n, we have*

$$\mathbb{P}(G_n \mid S) = \sum_{\sigma \in \Omega(G_n, S)} \prod_{l=1}^{n-k} \left(N_\sigma(\sigma_l, G) \cdot p_l(\sigma \mid G, S) \right). \tag{6}$$

Furthermore, suppose that $\mathcal{A}_n = \cup_{i=1}^k A_{n,i}$ is an infection partition for G_n. Let H_n be the minimal subgraph of G_n that contains all non-cyclic paths between any pair of nodes in S, and let [2]

$$\Omega(H_n, S, \mathcal{A}_n) = \{\sigma \in \Omega(H_n, S) : \sigma \cap A_{n,i} \textit{ is an infection sequence, for all } i = 1, \ldots, k.\}.$$

Then, we have

$$\mathbb{P}(\mathcal{A}_n \mid S, G_n) = \sum_{\sigma \in \Omega(H_n, S, \mathcal{A}_n)} \prod_{l=1}^{|H_n|-k} p_l(\sigma \mid H_n, S). \tag{7}$$

[2] For a sequence of nodes σ and a set A, the notation $\sigma \cap A$ represents the subsequence of σ containing only nodes that are in A.

The characterizations in Lemma 1 are computationally hard to evaluate. In Section III, we make further approximations and design algorithms to evaluate the estimators $\hat{S}$ and $\hat{\mathcal{A}}_n$ when G is a tree. In Section IV, we consider the case when G is a general graph.

III. Identifying Infection Sources and Regions for Trees

In this section, we consider the problem of estimating the infection sources and regions when the underlying network G is a tree. We first derive an estimator for the infection partition in (4), given any source node set S and G_n. Then, under simplifying approximations, we show that the estimator (3) is closely related to $C(S \mid G_n) = |\Omega(G_n, S)|$, the number of infection sequences. Our derivation for $\hat{S}$ is similar to [17], which considers only the single source estimation problem. Next, we consider the case where there are two infection sources, propose approximations that allow us to compute $\hat{S}$ with reasonable complexity, and show that our proposed estimator works well in an asymptotically large geometric tree. In most practical applications, the number of infection sources is not known a priori. We present a heuristic algorithm for general trees to estimate the infection sources when the number of infection sources is unknown, but bounded by $k_{\max}$.

A. Infection Partition with Multiple Sources

In this section, we derive an infection partition estimator for (4) given any S, under a simplifying technical condition. We show that the MAP estimator can be approximated by a Voronoi partition of G_n, where the distance measure is taken to be the path length. This is intuitively correct as nodes closer to a particular source are more likely to be infected by that source. The proof of the following result is provided in Appendix B.

Theorem 1. *Suppose that G is a tree with infection sources S. Let H_n be the minimal connected subgraph of G_n that spans S.*[3] *If any two paths in H_n do not intersect except possibly at nodes in S, then the MAP estimator $\hat{\mathcal{A}}_n(S)$ for the infection partition is a Voronoi partition of the graph G_n, where the centers of the partitions are the infection sources S.*

A Voronoi partition may not produce the MAP estimator for the infection partition in a general tree. However, for simplicity, we will henceforth approximate the MAP estimator with a Voronoi partition of the infection graph G_n, and present simulation results in Section V to verify the performance of this estimator.

B. Estimation of Infection Sources

We now consider the problem of estimating the set of infection sources S. When $|S| = 1$, our estimation problem reduces to that in [17], which considers only the single source infection problem. In the following, we introduce some notations, and briefly review some relevant results from [17].

[3] A connected subgraph $H = (V', E')$ of G_n spans S if $S \subset V'$. The subgraph H is minimal if it has the least number of nodes amongst all connected subgraphs that span S.

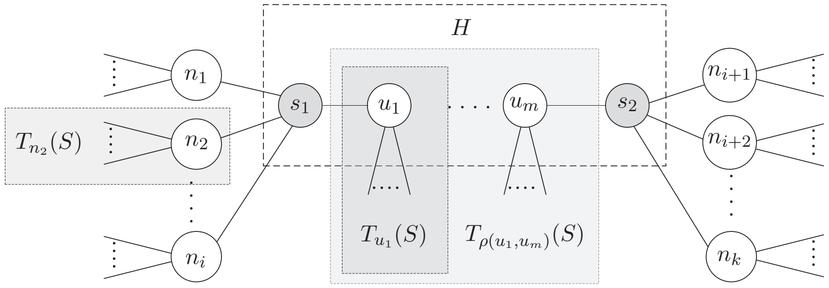


Fig. 2. An example infection graph with $S = \{s_1, s_2\}$

A path between any two nodes u and v in G_n is denoted as $\rho(u, v)$. For any set of nodes S in G_n, consider the minimal connected subgraph $H \subset G_n$ that spans S. Treat this subgraph has a "super" node, with the tree G_n rooted at this "super" node. For any node $v \in G_n \backslash H$, we define $T_v(S)$ to be the tree rooted at v with the path from v to H removed. For $v \in H$, we define $T_v(S)$ to be the tree rooted at v so that all edges between v and its neighbors in H are removed.[4] We say that $T_v(S)$ is the tree rooted at v with respect to (w.r.t.) S. For any subset of nodes $M \subset G_n$, we let $T_M(S) = \cup_{v \in M} T_v(S)$. An illustration of these definitions is shown in Figure 2. If $S = \{s_1, \ldots, s_k\}$, we will sometimes use the notation $T_v(s_1, \ldots, s_k)$ instead.

Let $C(S \mid G_n) = |\Omega(G_n, S)|$ be the number of infection sequences. Suppose that G is a regular tree where each node has the same degree. If $S = \{s\}$, we have for any infection sequence σ, $N_\sigma(\sigma_j, G) = 1$ and from (5), we have $p_l(\sigma \mid G, s)$ is identical for all $\sigma \in \Omega(G_n, s)$. Lemma 1 then shows that $\mathbb{P}(G_n \mid s) = C(s \mid G_n)$. Therefore, when there is a single source node, the MAP estimator for the infection source is obtained by evaluating $\hat{S} = \arg\max_{v \in G_n} C(v \mid G_n)$, which seeks to maximize $C(v \mid G_n)$ over all nodes. The following result is shown in [17].

Lemma 2. *For any node $v \in G_n$, we have*

$$C(v \mid G_n) = n! \prod_{u \in G_n} |T_u(v)|^{-1}. \tag{8}$$

We observe that each term $|T_u(v)|$ in the product on the right hand side (R.H.S.) of (8) is the number of nodes in the sub-tree $T_u(v)$ (and which appears when we account for the number of permutations of these nodes). We can think of the terms in the product being ordered according to the infection spreading sequence, i.e., each time we reach a particular node u, we include terms corresponding to the nodes u can potentially infect. This interpretation is useful in helping us understand the characterization in Lemma 3 for the case when there are two infection sources.

To compute $C(v \mid G_n)$, an $O(n)$ algorithm based on Lemma 2 was provided in [17]. We call this algorithm the Single Source Estimation (SSE) algorithm. We refer the reader to [17] for details about the implementation of the algorithm. Although finding $\hat{S}$ by maximizing $C(s \mid G_n)$ is exact only for regular trees, it was shown in [17] that this estimator has good performance for other classes of trees. In particular, if G is a geometric tree (cf.

[4] As $T_v(S)$ is defined on G_n, its notation should include G_n. However, in order to avoid cluttered expressions, we drop G_n in our notations. Confusion will be avoided through the context in which these trees are referenced.

Section III-D), then the probability, conditioned on $S = \{s\}$, of correctly identifying s using $C(s \mid G_n)$ goes to one as $n \to \infty$. Inspired by this result, we propose estimators based on quantities related to $C(S \mid G_n)$ for cases where $|S| > 1$. In the following, we first discuss the case where $|S| = 2$, and extend the results to the general case where $|S|$ is unknown in Section III-E. We then numerically compare our proposed algorithms with a modified SSE algorithm adapted for finding multiple sources in Section V.

C. Two Infection Sources

In this section, we assume that there are two infection sources, and propose approximations that allow us to compute $\hat{S}$ in (3) with reasonable complexity.

Let $S = \{s_1, s_2\}$. If G is a regular tree with node degree m, we have from (5) that for any infection sequence σ,

$$p_l(\sigma \mid G, S) = \left(2m + (l-1)(m-2) - 2\mathbf{1}_{\{l \geq \chi\}}\right)^{-1},$$

where χ is the index of the last node in the path between s_1 and s_2 to be infected. We note that χ varies from sequence to sequence. However, the probabilities $p_l(\sigma \mid G, S)$ are dominated by those sequences with small values of χ. Therefore, in the same spirit as [17], we approximate the estimator $\hat{S}$ by maximizing $C(S \mid G_n)$ over all two node sets S.

Given two nodes u and v in G_n, suppose that $|\rho(u,v)| = m$. For any permutation $\xi = (\xi_1, \ldots, \xi_m)$ of the nodes in $\rho(u,v)$, let

$$I_i(\xi; s_1, s_2) = \sum_{j \leq i} |T_{\xi_j}(s_1, s_2)| \tag{9}$$

be the total number of nodes in the trees rooted at the first i nodes in the permutation ξ. For $w \in \rho(u,v)$, recall that $\Omega(\rho(u,v), w)$ is the set of infection sequences in the graph $\rho(u,v)$ with w as the infection source. Let $\bar{\Omega}(\rho(u,v), w) = \{\xi = (w, \sigma) : \sigma \in \Omega(\rho(u,v), w)\}$ be the set of infection sequences augmented with w as the first node in the sequence. We have the following characterization for $C(s_1, s_2 \mid G_n)$. The proof can be found in Appendix C.

Lemma 3. *Consider any two nodes s_1 and s_2 in G_n, and suppose that $\rho(s_1, s_2) = (s_1, u_1, \ldots, u_m, s_2)$ with nodes $u_1, \cdots, u_m$ in G_n. We have*

$$C(s_1, s_2 \mid G_n) = (n-2)! \cdot q(u_1, u_m; s_1, s_2) \prod_{u \in G_n \backslash \rho(s_1, s_2)} |T_u(s_1, s_2)|^{-1}, \tag{10}$$

where

$$q(u_1, u_m; s_1, s_2) = \sum_{\xi \in \Gamma} \prod_{i=1}^{m} I_i(\xi; s_1, s_2)^{-1}, \tag{11}$$

and

$$\Gamma = \bigcup_{w \in \rho(u_1, u_m)} \bar{\Omega}(\rho(u_1, u_m), w). \tag{12}$$

Furthermore, for $1 \leq i \leq j \leq m$, $q(u_i, u_j; s_1, s_2)$ *satisfies the following recursive relationships*

$$q(u_i, u_j; s_1, s_2) = |T_{\rho(u_i,u_j)}(s_1, s_2)|^{-1} \left(q(u_{i+1}, u_j; s_1, s_2) + q(u_i, u_{j-1}; s_1, s_2) \right) \text{ for } i < j, \tag{13}$$

and

$$q(v, v; s_1, s_2) = |T_v(s_1, s_2)|^{-1} \qquad \forall\ v \in \rho(u_1, u_m). \tag{14}$$

The characterization for $C(s_1, s_2 \mid G_n)$ is similar to that for the single source case in (8), except for the additional $q(u_1, u_m; s_1, s_2)$ term. Each sequence in the set Γ can be interpreted as the *reverse* infection sequence of the nodes in $\rho(u_1, u_m)$ due to the sources s_1 and s_2. The set Γ then consists of all possible infection sequences for the graph $\rho(u_1, u_m)$. For each sequence $\xi = (\xi_1, \ldots, \xi_m) \in \Gamma$, each term $I_i(\xi; s_1, s_2)$ in the product in the R.H.S. of (11) corresponds to the number of nodes in the graph $T_{(\xi_1,\ldots,\xi_i)}(s_1, s_2)$, i.e., as the infection spreads to ξ_{i+1}, ξ_i is a potential node that ξ_{i+1} can infect, and $T_{(\xi_1,\ldots,\xi_i)}(s_1, s_2)$ is the subgraph that will be infected by ξ_i if ξ_i is the only source. The other terms corresponding to $T_u(s_1, s_2)$, where $u \in T_{\xi_{i+1}}(s_1, s_2)$ appear in the product term on the R.H.S. of (10).

By utilizing Lemma 3, we can compute $C(u, v \mid G_n)$ for any two nodes u and v in G_n by evaluating $|T_w(u, v)|$ for all nodes $w \in G_n$, and the quantity $q(u_1, u_m; u, v)$, where $\rho(u, v) = (u, u_1, \ldots, u_m, v)$. In order to evaluate $|T_w(u, v)|$, we make the assumption that the degree of every node in G is bounded.

Assumption 2. *Every node in* G *has bounded degree.*

For each node w and its neighbor u, recall that $T_w(u)$ is the subtree rooted at w w.r.t. u. With Assumption 2, Algorithm 1 allows us to compute $f_w(u) = |T_w(u)|$ and $g_w(u) = \prod_{v \in T_w(u)} |T_v(u)|$ for all neighbors u of w, and for all $w \in G_n$ in $O(n)$ time complexity. Choose any node $r \in G_n$, and consider G_n as a directed tree with r as the root node, and with edges in G_n pointing away from r. Let $\mathrm{pa}(w)$ be the parent of w in the directed tree G_n. Starting from the leaf nodes, let each non-root node $w \in G_n$ pass two messages containing $f_w(\mathrm{pa}(w))$ and $g_w(\mathrm{pa}(w))$ to its parent. Each node stores the values of these two messages from each of its children in a local database, and computes its two messages to be passed to its parent from this database. When r has received all messages from its children, a reverse sweep down the tree is done so that at the end of the algorithm, every node $w \in G_n$ has stored the values $\{f_u(w), g_u(w) : u \in \mathcal{N}(w)\}$. The algorithm is formally described in Algorithm 1, where $\mathrm{ch}(w)$ is the set of child nodes of w in the directed tree G_n. The last product term on the R.H.S. of (10) can then be computed using

$$g(s_1, s_2) = \prod_{w \in \rho(s_1, s_2)} \prod_{x \in \mathcal{N}(w) \backslash \rho(s_1, s_2)} g_x(w), \tag{15}$$

and taking its reciprocal.

To compute $C(s_1, s_2 \mid G_n)$ in (10), we still need to compute $q(u_1, u_m; s_1, s_2)$. The recursive relationships (13)-(14) allow us to compute $q(u_1, u_m; s_1, s_2)$ for all $s_1, s_2 \in G_n$ in $O(n^2 d_*^2)$ complexity, where d_* is the maximum node degree. The computation proceeds by first considering each pair of neighbors (u, v). Both nodes have at most

Algorithm 1 Tree Sizes and Products Computation

```
**Inputs**: $G_n$
Choose any node $r \in G_n$ as the root node.
**for** $w \in G_n$ **do**
    Store received messages $f_x(w)$ and $g_x(w)$, for each $x \in \mathrm{ch}(w)$.
    **if** $w$ is a leaf **then**
        $f_w(\mathrm{pa}(w)) = 1$
        $g_w(\mathrm{pa}(w)) = 1$
    **else**
        $f_w(\mathrm{pa}(w)) = \sum_{x \in \mathrm{ch}(w)} f_x(w) + 1$
        $g_w(\mathrm{pa}(w)) = f_w(\mathrm{pa}(w)) \cdot \prod_{x \in \mathrm{ch}(w)} g_x(w)$
    **end if**
    Store $f_{\mathrm{pa}(w)}(w) = n - f_w(\mathrm{pa}(w))$.
    Pass $f_w(\mathrm{pa}(w))$ and $g_w(\mathrm{pa}(w))$ to $\mathrm{pa}(w)$.
**end for**
Set $g_{\mathrm{pa}(r)}(r) = 1$.
**for** $w \in G_n$ **do**
    Store received message $g_{\mathrm{pa}(w)}(w)$ from $\mathrm{pa}(w)$.
    **if** $w$ is not a leaf **then**
        **for** $x \in \mathrm{ch}(w)$ **do**
            $g_w(x) = f_w(x) \cdot g_{\mathrm{pa}(w)}(w) \cdot \prod_{y \in \mathrm{ch}(w) \backslash \{x\}} g_y(w)$
            Pass $g_w(x)$ to $x$.
        **end for**
    **end if**
**end for**
```

d_* neighbors each, so that we need to evaluate $q(u, v; s_1, s_2)$ for all $s_1 \in \mathcal{N}(u) \backslash \rho(u, v)$ and $s_2 \in \mathcal{N}(v) \backslash \rho(u, v)$. This requires $O(d_*^2)$ computations. The computed values and $T_{\rho(u,v)}(s_1, s_2)$ are stored in a hash table. In the next step, we repeat the same procedure for node pairs that are two hops apart, and so on until we have considered every pair of nodes in G_n. Note that for a path $(u_1, \ldots, u_m)$ and s_1, s_2 neighbors of u_1 and u_m respectively, $q(u_1, u_m; s_1, s_2)$ can be computed in constant time from (13) as $q(u_2, u_m; s_1, s_2) = q(u_2, u_m; u_1, s_2)$ and $q(u_1, u_{m-1}; s_1, s_2) = q(u_1, u_{m-1}; s_1, u_m)$. A similar remark applies for the computation of $|T_{\rho(u_1,u_m)}(s_1, s_2)|$. In addition, each lookup of the hash table takes $O(1)$ complexity since G_n is known and collision-free hashing can be used. Therefore, the overall complexity is $O(n^2 d_*^2)$. The algorithm to compute the infection sources estimator is formally given in Algorithm 2. We call this the Two Source Estimation (TSE) algorithm, and it forms the basis of our algorithm for

Algorithm 2 Two Source Estimation (TSE)

```
Inputs: $G_n$
Let $(s_1^*, s_2^*)$ be the maximizer of $C(\cdot,\cdot \mid G_n)$. Set $C^* = 0$.
for $d = 1$ to diameter of $G_n$ do
    for each $s_1 \in G_n$ do
        for each $s_2$ such that $d(s_1, s_2) = d$ do
            Let $\rho(s_1, s_2) = (s_1, u_1, \ldots, u_{d-1}, s_2)$.
            if $d = 1$ then
                $q(u_1, u_{d-1}; s_1, s_2) = 1$.
            else if $d = 2$ then
                Store $q(u_1, u_1; s_1, s_2) = |T_{u_1}(s_1, s_2)|^{-1}$ and $|T_{u_1}(s_1, s_2)|$.
            else
                Lookup $|T_{\rho(u_1, u_{d-2})}(s_1, u_{d-1})|$, $q(u_2, u_{d-1}; u_1, s_2)$, and $q(u_1, u_{d-2}; s_1, u_{d-1})$.
                Store
                    $$|T_{\rho(u_1, u_{d-1})}(s_1, s_2)| = |T_{\rho(u_1, u_{d-2})}(s_1, u_{d-1})| \cdot |T_{u_{d-1}}(s_1, s_2)|.$$
                Store
                    $$q(u_1, u_{d-1}; s_1, s_2) = \frac{q(u_2, u_{d-1}; u_1, s_2) + q(u_1, u_{d-2}; s_1, u_{d-1})}{|T_{\rho(u_1, u_{d-1})}(s_1, s_2)|}.$$
            end if
            Compute $g(s_1, s_2)$ from (15).
            $C(s_1, s_2 \mid G_n) = (n-2)! q(u_1, u_{d-1}; s_1, s_2)/g(s_1, s_2)$.
            Update $(s_1^*, s_2^*)$ and $C^*$ if $C(s_1, s_2 \mid G_n) > C^*$.
        end for
    end for
end for
```

multiple sources estimation in the sequel.

D. Geometric Trees with Two Sources

In this section, we study the special case of geometric trees, propose an approximate MAP estimator for geometric trees, and provide theoretical analysis for its performance. First, we give the definition of geometric trees and prove some of its key properties. Then, we derive a lower bound for $C(S \mid G_n)$, and propose an estimator based on this lower bound. We show that our proposed estimator is asymptotically correct, i.e., it identifies the actual infection sources with probability (conditioned on the infection sources) going to one as the infection graph G_n becomes large. For mathematical convenience, instead of letting the number of infected nodes n grow large, we let the time t from the start of the infection process to our observation time become large.

Our definition of a geometric tree generalizes from that in [17]. Let $S = \{s_1, s_2\}$ be the infection sources. Let $M = \rho(s_1, s_2)\backslash S$, and define the set A to consist of the element M, and all neighbors of the infection sources except those on the path $\rho(s_1, s_2)$. For each $u \in A$, let $T'_u(s_1, s_2)$ be defined in the graph G in the same way as $T_u(s_1, s_2)$ is defined for G_n. For each $u \in A$, and each $v \in T'_u(s_1, s_2)$, let $n_u(v, r)$ be the number of nodes in $T'_u(s_1, s_2)$ that are at a distance r from v. Then we say that G is a geometric tree if for all $u \in A$, and for all $v \in T'_u(s_1, s_2)$, we have

$$br^\alpha \leq n_u(v, r) \leq cr^\alpha, \tag{16}$$

where α, b, and c are fixed positive constants with $b \leq c$. The condition (16) implies that all trees defined w.r.t. the infection sources are growing polynomially fast at about the same rate. As we have assumed that the infection rates are homogeneous for every node, the resulting infection graph G_n will also be approximately regular with high probability. We have the following properties for a geometric tree, whose proofs are in Appendix D.

Lemma 4. *Suppose that s_1 and s_2 are the infection sources, and G is a geometric tree. For any $\epsilon \in (0, 1)$, let $\mathcal{E}_t$ be the event that all nodes within distance $t(1-t^{-1/2+\epsilon})$ of either source nodes are infected, and no nodes greater than distance $t(1 + t^{-1/2+\epsilon})$ of either source nodes are infected. Then, there exists t_0 such that for all $t \geq t_0$, $\mathbb{P}(\mathcal{E}_t) \geq 1 - \epsilon$. Furthermore, conditioned on $\mathcal{E}_t$, we have for all $u \in A$,*

$$N_{\min}(t) \leq |T_u(s_1, s_2)| \leq N_{\max}(t), \tag{17}$$

where

$$N_{\min}(t) = \frac{b}{1+\alpha}\left(t - t^{\frac{1}{2}+\epsilon} - 2\right)^{\alpha+1}, \tag{18}$$

and

$$N_{\max}(t) = \frac{c}{1+\alpha}\left(t + t^{\frac{1}{2}+\epsilon}\right)^{\alpha+1}. \tag{19}$$

In addition, for $t \geq t_0$, we have

$$\frac{N_{\min}(t)}{N_{\max}(t)} \geq \frac{b}{c}(1-\epsilon).$$

The infection sequences count in (10) is not amendable to analysis. In the following, we seek an approximation to simplify our analysis. For $s_1, s_2 \in G_n$, suppose that $\rho(s_1, s_2) = (s_1, u_1, \ldots, u_m, s_2)$, with $p = |\rho(s_1, s_2)| = m + 2$. Instead of computing $C(s_1, s_2 \mid G_n)$, we consider a new infection graph G'_n with two “virtual” nodes x_i, $i = 1, 2$ added, where x_i is attached to s_i (see Figure 3). We now consider $C(x_1, x_2 \mid G'_n) \geq C(s_1, s_2 \mid G_n)$. Since the trees rooted at x_i are single node trees, we have

$$\begin{aligned} C(x_1, x_2 \mid G'_n) &= C(s_1, x_2 \mid G'_n) + C(x_1, s_2 \mid G'_n) \\ &\leq 2(n-1)C(s_1, s_2 \mid G_n), \end{aligned}$$

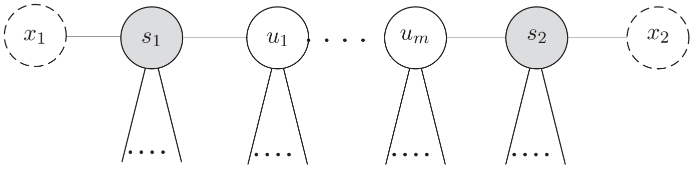


Fig. 3. Addition of virtual nodes x_1 and x_2.

where the last inequality follows because if s_1 and x_2 are sources, then s_2 can be inserted in any of at most $n-1$ positions in an infection sequence from $\Omega(G_n, \{s_1, s_2\})$, so that $C(s_1, x_2 \mid G'_n) \leq (n-1)C(s_1, s_2 \mid G_n)$. A similar argument holds for $C(x_1, s_2 \mid G'_n) \leq (n-1)C(s_1, s_2 \mid G_n)$.

Let $\xi^* = (\xi^*_1, \ldots, \xi^*_p)$ be a permutation of the nodes in $\rho(s_1, s_2)$ such that $|T_{\xi^*_i}(s_1, s_2)| \leq |T_{\xi^*_j}(s_1, s_2)|$ for all $1 \leq i \leq j \leq p$. Let $I^*_i(s_1, s_2) = I_i(\xi^*; s_1, s_2)$ (cf. the definition in (9)). Then, $I^*_i(s_1, s_2)$ is the total number of nodes in the i biggest trees in $\{T_u(s_1, s_2) : u \in \rho(s_1, s_2)\}$. From Lemma 3, we have

$$C(x_1, x_2 \mid G'_n) \geq n! \cdot 2^{p-1} \prod_{i=1}^{p} I^*_i(s_1, s_2)^{-1} \prod_{u \in G_n \setminus \rho(s_1, s_2)} |T_u(s_1, s_2)|^{-1}, \tag{20}$$

where the inequality holds because $|\Gamma| = 2^{p-1}$, and each term in the sum on the R.H.S. of (11) is lower bounded by $\prod_{i=1}^{p} I^*_i(s_1, s_2)^{-1}$. We use the lower bound in (20) as a proxy for $C(s_1, s_2 \mid G_n)$. However, we have used a very loose lower bound in (20), so we propose the estimator

$$\tilde{S} = \arg \max_{s_1, s_2 \in G_n} \tilde{C}(s_1, s_2 \mid G_n), \tag{21}$$

where

$$\tilde{C}(s_1, s_2 \mid G_n) = n! \cdot Q(s_1, s_2) \prod_{u \in G_n \setminus \rho(s_1, s_2)} |T_u(s_1, s_2)|^{-1}, \tag{22}$$

$$Q(s_1, s_2) = [2(1+\delta)]^{p-1} \prod_{i=1}^{p} I^*_i(s_1, s_2)^{-1},$$

and δ is a fixed positive constant, to be chosen based on prior knowledge about the graph G. Algorithm 2 can be modified to find the maximizer for $\tilde{C}(\cdot, \cdot \mid G_n)$. We call this the geometric tree TSE algorithm. The following result provides a way to choose δ, and shows that our proposed estimator $\tilde{S}$ is asymptotically correct in a geometric tree. A proof is provided in Appendix E.

Theorem 2. *Suppose that G is a geometric tree with infection sources $S = \{s_1, s_2\}$. Let $d_{\min}$ and $d_{\max}$ be constants such that $\deg_G(s_i) \in [d_{\min}, d_{\max}]$ for $i = 1, 2$. Suppose that*

$$d_{\min} \geq \frac{3}{2} + \frac{c}{b}\sqrt{2d_{\max}}. \tag{23}$$

Then, for any δ in the non-empty interval

$$\left(\frac{cd_{\max}}{b(d_{\min}-1)} - 1, \frac{b(d_{\min}-2)}{2c} - 1\right), \tag{24}$$

we have

$$\lim_{t\to\infty} \mathbb{P}(\tilde{S} = S \mid S) = 1.$$

Theorem 2 implies that if we know the constants governing the regularity condition (16) for G, we can choose a δ so that our estimator $\tilde{S}$ gives the true infection sources with high probability if the infection graph G_n is large. The class of geometric trees as defined by (16) can be used to model various scenarios in practice, e.g., a tree spanning a wireless sensor network with nodes randomly scattered. However, the assumption (16) may also be overly strong for other applications. In Section V, we perform numerical studies to gain insights into the performance of our proposed estimator for different classes of tree networks.

E. Unknown Number of Infection Sources

In most practical applications, the number of infection sources is not known a priori. However, typically we may be able to guess the maximum number of infection sources $k_{\max}$, or we can choose a reasonable value of $k_{\max}$ depending on the size of the infection graph G_n. In this section, we present a *heuristic* algorithm that allows us to estimate the infection sources with a given $k_{\max}$.

We first consider the instructive case where $k_{\max} = 2$ and G is a geometric tree. In this case, the number of infection sources can be either one or two. Suppose we run the geometric tree TSE algorithm on G_n. We have the following result, whose proof is in Appendix F.

Theorem 3. *Suppose that there is a single infection source s and G is a geometric tree with* (16) *holding for all nodes u that are neighbors of s. Suppose that s has degree $\deg_G(s) \in [d_{\min}, d_{\max}]$, where $d_{\min}$ and $d_{\max}$ are positive constants satisfying* (23)*. Then, for any δ in the interval* (24)*, the geometric tree TSE algorithm estimates as sources s and one of its neighbors with probability (conditioned on s being the infection source) going to* 1 *as* $t \to \infty$.

Theorem 3 implies that when there exists only one source, the geometric tree TSE algorithm finds two neighboring nodes, one of which is the true source. From Theorem 2 and Assumption 1, if there are two sources, our algorithm identifies the two source nodes, which are at least two hops from each other, with high probability. Therefore, by checking the distance between the two nodes identified by the geometric tree TSE algorithm, we can estimate the number of source nodes in the infection graph. This observation together with Theorem 1 suggest the following heuristic.

(i) Randomly choose $k_{\max}$ nodes satisfying Assumption 1 as the infection sources and find a Voronoi partition for G_n. Use the SSE algorithm to find a source node for each infection region. Repeat these steps until for every region, the distance between estimated source nodes between iterations is below a fixed threshold or a maximum number of iterations is reached. We call this the Infection Partition (IP) Algorithm (see Algorithm 3).

Algorithm 3 Infection Partitioning (IP)

```
**Inputs**: An infection source set $S^{(0)} = \{s_i^{(0)} : i = 1, \ldots, m\}$ in $G_n$.
**Iterations**:
**for** $l = 1$ to MaxIter **do**
    Run the Voronoi partitioning algorithm with centers in $S^{(l-1)}$ to obtain the infection partition $\mathcal{A}^{(l)} = \cup_{i=1}^{m} A_i^{(l)}$.
    **for** $i = 1$ to $m$ **do**
        Run SSE algorithm in $A_i^{(l)}$ to obtain
            $$s_i^{(l)} = \arg \max_{s \in A_i^{(l)}} C(s \mid A_i^{(l)}).$$
    **end for**
    $S^{(l)} := \{s_i^{(l)} : i = 1, \ldots, m\}$
    **if** $\max_{1 \le i \le m} d(s_i^{(l)}, s_i^{(l-1)}) \le \eta$ for some fixed small positive $\eta$ **then**
        break
    **end if**
**end for**
**return** $(S^{(l)}, \mathcal{A}^{(l)})$
```

(ii) For any two regions in the partition obtained from step (i) that can be connected by adding an edge in G_n, run the TSE algorithm in the combined region to determine if there are indeed two infection sources. If it is determined that there is only one infection source in the combined region, we decrement the number of source nodes, and repeat step (i). These two steps are repeated until no two pairs of regions in the Voronoi partition can be combined. The formal algorithm is given as the Multiple Sources Estimation and Partitioning (MSEP) algorithm in Algorithm 4.

To compute the complexity of the MSEP algorithm, we note that since the IP algorithm is based on the SSE algorithm, it has complexity $O(n)$. For each value of $k = 1, \ldots, k_{\max}$ in the MSEP algorithm, there are $O(k^2)$ pairs of neighboring regions in the infection partition. For each pair of region, the TSE algorithm makes $O(n^2)$ computations. Summing over all $k = 1, \ldots, k_{\max}$, the time complexity of the MSEP algorithm can be shown to be $O(k_{\max}^3 n^2)$. On the other hand, to compute $C(S \mid G_n)$ for $|S| = k_{\max}$ would require $O(n^{k_{\max}})$ computations.

IV. Identifying Infection Sources and Regions for General Graphs

In this section, we generalize the MSEP algorithm to identify multiple infection sources in general graphs G. Such network structures are frequently encountered in practical applications. Examples include small-world networks [29] and power grid networks [29]. In [17], the SSE algorithm is extended to general graphs when it is known that there is only a single infection source in the network using a heuristic. The algorithm first chooses a node s of G_n

Algorithm 4 Multiple Sources Estimation and Partitioning (MSEP)

```
1:  **Inputs**: $G_n$ and $k_{\max}$.
2:  **Initialization**:
3:  $k := k_{\max}$ and choose $S := \{s_1, \ldots, s_k\}$ randomly in $G_n$.
4:  **Iterations**:
5:  **while** $k > 1$ **do**
6:      $(S, \mathcal{A}) =$ Algorithm IP($S$)
7:      $S' := S$
8:      **for all** regions $A_i$ and $A_j$ in the partition $\mathcal{A}$ such that there exists an edge $(u, v)$ in $G_n$ with $u \in A_i$ and $v \in A_j$ **do**
9:          Set $(u, v) =$ Algorithm TSE($A_i \cup A_j$).
10:         **if** $d(u, v) < \tau$ **then**
11:             Merge $A_i$ and $A_j$, set $s_i = u$ and discard $s_j$
12:             $k := k - 1$
13:             break
14:         **end if**
15:     **end for**
16:     **if** $S = S'$ **then**
17:         break
18:     **end if**
19: **end while**
20: **return** $(S, \mathcal{A})$
```

as the root node, and generates a spanning tree $T_{\rm bfs}(s, G_n)$ of G_n rooted at s using the breadth-first-search (BFS) procedure. The SSE algorithm is then applied on this spanning tree to compute $C(s \mid T_{\rm bfs}(s, G_n))$. In addition, the infection sequences count is weighted by the likelihood of the BFS tree. This is repeated using every node in G_n as the root node, and the node $\hat{s}$ with the maximum weighted infection sequences count is chosen as the source estimator, i.e.,

$$\hat{s} = \arg\max_{v \in G_n} \mathbb{P}(\sigma_v \mid v) C(s \mid T_{\rm bfs}(v, G_n)),$$

where σ_v is the sequence of nodes that corresponds to an infection spreading from v along the BFS tree. It can be shown that this algorithm has complexity $O(n^2)$. For further details, the reader is referred to [17]. We call this algorithm the SSE-BFS algorithm in this paper.

We adapt the MSEP algorithm for general graphs using the same BFS heuristic. Specifically, we replace the SSE algorithm in line 6 of the IP algorihm with the SSE-BFS algorithm. In addition, in line 9, we run the TSE

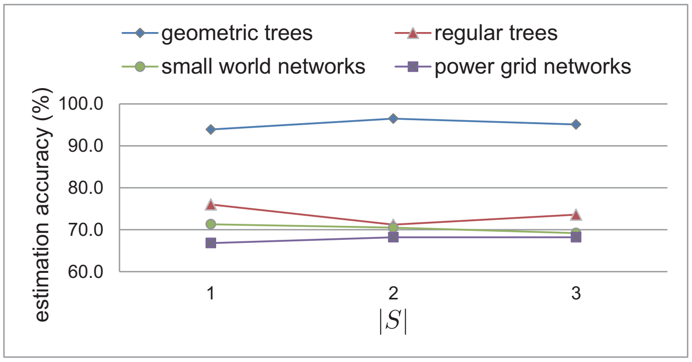


Fig. 4. Estimating the number of infection source nodes.

algorithm on $T_{\rm bfs}(s_i, A_i) \cup T_{\rm bfs}(s_j, A_j)$, where the two BFS trees are connected by randomly selecting an edge (u, v) in G_n with $u \in T_{\rm bfs}(s_i, A_i)$ and $v \in T_{\rm bfs}(s_j, A_j)$. We call this modified algorithm the MSEP-BFS algorithm. Since the worst case complexity for the SSE-BFS algorithm is $O(n^2)$, the complexity of the MSEP-BFS algorithm can be shown to be $O(k_{\max}^3 n^2)$, which is the same complexity as the MSEP algorithm. To verify the effectiveness of the MSEP-BFS algorithm, we conduct simulations on both synthetic and real world networks in Section V.

V. Simulation Results and Discussions

In this section, we present simulation results on different network structures to verify our proposed algorithms. We first consider geometric tree networks and regular tree networks with various numbers of infection sources, and then we present results on small-world networks and a real world power grid network. We also apply our algorithms to the contact tracing data obtained during the Severe Acute Respiratory Syndrome (SARS) outbreak in Singapore in 2003 [30].

A. Synthetic Networks

We perform simulations on geometric trees, regular trees, and small-world networks. The number of infection sources are chosen to be 1, 2, or 3, and we set $k_{\max} = 3$. For each type of network and each number of infection sources, we perform 1000 simulation runs with 500 infected nodes each. We randomly choose infection sources satisfying Assumption 1 and obtain the infection graph by simulating the infection spreading process using the SIR model. Finally, the MSEP or MSEP-BFS algorithm for tree networks and small-world networks respectively, is applied to the infection graph to estimate the number and locations of the infection sources. The estimation results for the number of infection sources in different scenarios are shown in Figure 4. It can be seen that our algorithm correctly finds the number of infection sources more than 93% of the time for geometric trees, and more than 71% of the time for regular trees. The accuracy of about 69.2% for small-world networks is worse than that for the tree networks, as the infection tree for a small-world network has to be estimated using the BFS heuristics, thus additional errors are introduced into the procedure.

simulation settings		average	average error distances		minimum infection region
network topology	$\|S\|$	diameter	MSEP/MSEP-BFS	nSSE	covering percentage (%)
geometric trees	2	63.7	0.56	12.86	98.6
	3	66.2	0.89	15.12	94.4
regular trees	2	40.5	0.94	6.08	97.0
	3	43.7	1.06	6.53	89.6
small-world networks	2	35.5	2.99	8.27	93.8
	3	40.9	2.9	10.36	87.7
power grid network	2	27.3	3.86	7.94	92.9
	3	30.8	3.38	9.01	87.9

TABLE I

AVERAGE ERROR DISTANCES AND MINIMUM INFECTION REGION COVERING PERCENTAGE FOR VARIOUS NETWORKS, CONDITIONED ON CORRECT SOURCE NUMBER ESTIMATION.

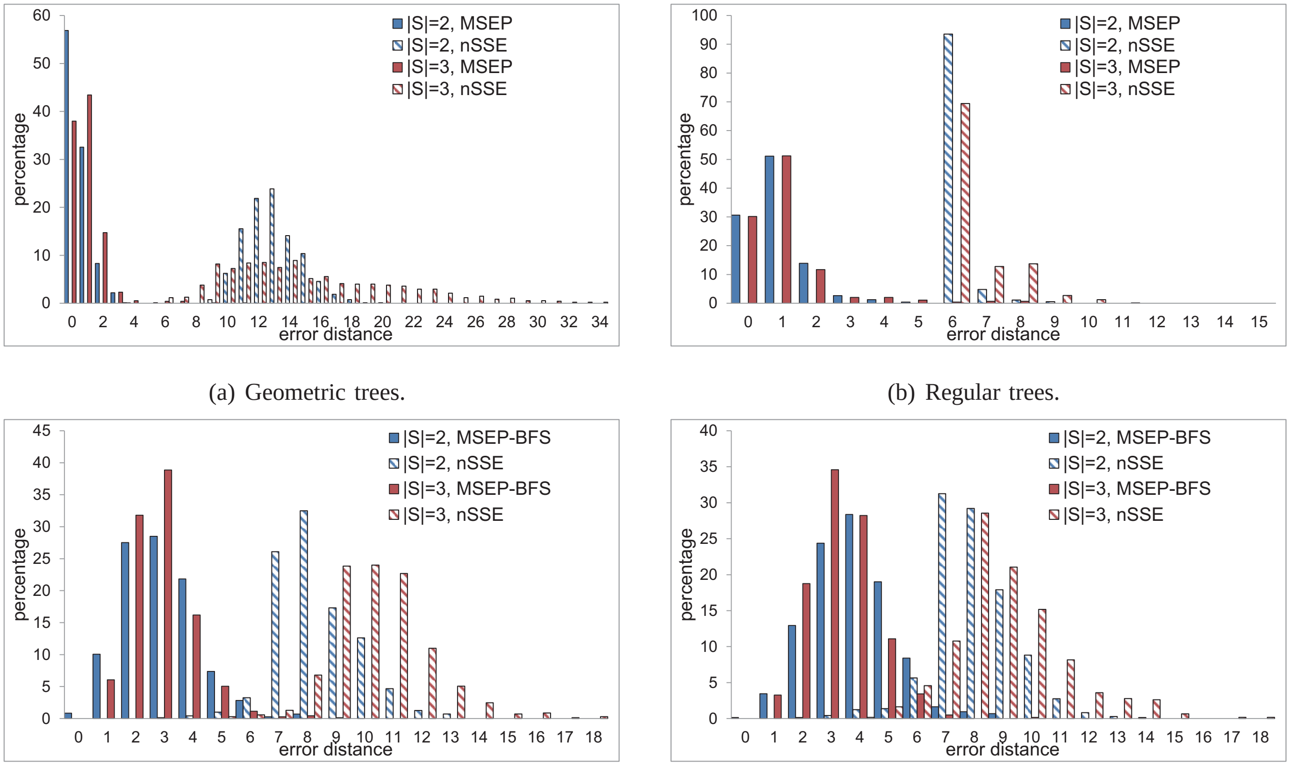
When it is known that there are more than one infection sources, we compare the performance of the MSEP algorithm with a naive estimator based on the SSE algorithm. In the estimator for tree networks, we first compute $C(u \mid G_n)$ for all nodes $u \in G_n$, and choose the $|S|$ nodes with the largest counts as the source nodes. In the small-world networks, we use the SSE-BFS algorithm. We call this the nSSE algorithm. In comparison, the MSEP or MSEP-BFS algorithm does not require us to know $|S|$ a priori. However, to perform a fair comparison, we consider only those simulation runs in which the MSEP or MSEP-BFS algorithm correctly estimates the number of infection source nodes. The error distance is found by first matching the estimated source nodes with the actual sources so that the sum of the distance between each estimated source and its match is minimized. We then divide this sum by the number of source nodes to obtain the error distance.

The histogram of the error distances for the different types of networks are shown in Figure 5. The error distances averaged over all simulation runs are provided in Table I. Clearly, the MSEP/MSEP-BFS algorithm outperforms the nSSE algorithm in every case. Moreover, the performance of the nSSE algorithm deteriorates with increasing $|S|$. This is to be expected as the SSE algorithm assumes that the node with the largest infection sequence count is the only source, and this node tends to be close to the distance center [31] of the infection graph.

The MSEP/MSEP-BFS algorithm also estimates the infection region of each source. To evaluate its accuracy, suppose that the infection sources are $S = \{s_1, \ldots, s_k\}$, and let the true infection region of source s_i be $A_{n,i}$. Let the MSEP/MSEP-BFS estimated infection region of s_i be $\hat{A}_{n,i}$. We define the correct infection region covering percentage for s_i as the ratio between $|\hat{A}_{n,i} \cap A_{n,i}|$ and $|A_{n,i}|$, and we compute the minimum (or worst case) infection region covering percentage as

$$\min_{i \in \{1,\cdots,k\}} \frac{|\hat{A}_{n,i} \cap A_{n,i}|}{|A_{n,i}|}.$$

We find that the minimum infection region covering percentage is more than 87% for all networks, as shown in Table I.



(a) Geometric trees.

(b) Regular trees.

(c) Small-world networks.

(d) US power grid network.

Fig. 5. Histogram of the error distances for various networks.

B. *Real World Networks*

In this section, we verify the performance of the MSEP-BFS algorithm on real world networks. We consider the western states power grid network of the United States [29], and contact tracing data collected during the SARS epidemic in Singapore in the year 2003 [30].

We simulate the infection spreading process on the power grid network, which contains 4941 nodes. For each simulation run, 1, 2 or 3 infection sources are randomly chosen from the power grid network under Assumption 1, and the spreading process is simulated so that a total of 500 nodes are infected. For each value of $|S|$, 1000 simulation runs are performed. The simulation results are shown in Figures 4 and 5 (d), and Table I. We see that the MSEP-BFS algorithm significantly outperforms the nSSE algorithm, with an average error distance of less than 4 compared to an error of more than 7.9 for the nSSE algorithm. The minimum infection region covering percentage is also above 87%.

In our final numerical study, we apply the MSEP algorithm to to a network of nodes that represent the individuals who were infected with the SARS virus during an epidemic in Singapore. The data is collected using contact tracing of patients [30], where an edge between two nodes indicate that there is some form of interaction or relationship between the individuals (e.g., they are family members, classmates, colleagues, or commuters who shared the same public transport system). A fragment of the SARS infection network is shown in Figure 6. The arrows indicate the chain of transmission, and the index node is the infection source. We test the MSEP algorithm on a network

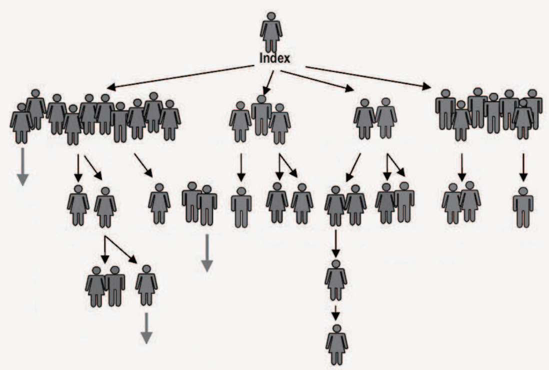


Fig. 6. Part of the SARS infection network with a single source (abstracted from Figure 1 of [30]).

of 193 nodes, assuming that there are at most $k_{\max} = 3$ infection sources. It turns out that the MSEP algorithm correctly estimates the number of infection source to be one, and correctly identifies the real infection source.

VI. Conclusion

We have derived estimators for the infection sources and regions when the number of infection sources is bounded but unknown a priori. The estimators are based only on knowledge of the infected nodes and their underlying network connections. We provide an approximation for the infection source estimator for the class of geometric trees, and when there are at most two sources in the network. We show that this estimator asymptotically correctly identifies the infection sources when the number of infected nodes grows large. We also propose an algorithm that estimates the number of source nodes, and identify them and their respective infection regions for general infection graphs. Simulation results on geometric trees, regular trees, small-world networks, the US power grid network, and the SARS infection network show that our proposed estimation procedure performs well in general, with an average error distance of less than 4 when the number of source nodes is correctly estimated. The estimation accuracy of the number of source nodes is over 65% in all the networks we consider, with the geometric tree networks having an accuracy of over 90%. Furthermore, the minimum infection region covering percentage is more than 87% for all networks. Our estimation procedure assumes only knowledge of the underlying network connections. In practical applications where more information about the infection process is available, a more accurate and intelligent guess of the number of infection sources can be made.

In this paper, we have adopted a simple SIR infection model with homogeneous spreading rates, allowing us to derive analytical results that provide useful insights into infection source estimation for practical networks. However, this simplistic diffusion model does not adequately capture the real world dynamics of many networks. Future research includes the use of richer diffusion models that allow the inclusion of drifts and other dynamics in the infection spreading process, and tools from statistics to approximate optimal estimators for the infection sources. Our proposed algorithms find a set of nodes most likely to infect or influence a network, and are thus potentially useful for various practical applications. For example, our algorithm may be integrated with non-model-based consensus methods [32], [33] to design multi-agent control systems that uses only a small subset of agents as controllers. In cloud-centric media platforms [34], [35], variants of our proposed algorithm may be used for intelligent content cache management. These are all areas of future research.

Appendix A
Proof of Lemma 1

The proof of (6) is the similar to that of (5) in [17]. For completeness, we repeat the argument here. The posterior probability of any infection sequence $\sigma \in \Omega(G_n, S)$ is given by

$$\mathbb{P}(\sigma \mid S) = \prod_{l=1}^{n-k} \mathbb{P}(\sigma_l \mid S, \sigma_1, \ldots, \sigma_{l-1}).$$

Since infection times are independent exponential random variables with the same rate, the next infected node is chosen from the set of susceptible nodes with probability proportional to the number of infected neighbors. We have $\mathbb{P}(\sigma_l \mid S, \sigma_1, \ldots, \sigma_{l-1}) = N_\sigma(\sigma_l, G)/n_l$, where $n_l = \sum_{u \in J_l} N_l(u)$, with J_l being the set of susceptible nodes and $N_l(u)$ the number of infected neighbors of u, immediately before the lth infection. We show by induction on l that $n_l = 1/p_l(\sigma \mid G_n, S)$. The claim trivially holds for $l = 1$. Suppose that it holds for $l - 1$. Consider an urn containing n_l balls, with each ball colored in one of $|J_l|$ colors. Each color corresponds to one node in J_l, and the number of balls of the same color corresponds to the number of infected neighbors of that node. When σ_{l-1} becomes infected, $\deg_G(\sigma_{l-1}) - N_\sigma(\sigma_{l-1}, G)$ new balls are added to the urn, and $N_\sigma(\sigma_{l-1}, G)$ balls are removed from the urn. Therefore, $n_l = n_{l-1} + \deg_G(\sigma_{l-1}) - 2N_\sigma(\sigma_{l-1}, G)$, and the claim follows by induction. The proof of (6) now follows from (2).

To show (7), let nodes that are infected by source s_i be colored with color i, where $i = 1, \ldots, k$. The color of a node that is not on a path between two infection sources is uniquely determined with probability one when conditioned on the coloring of H_n, therefore it suffices to evaluate $\mathbb{P}(\mathcal{A}_n \cap H_n \mid S, H_n)$. The set $\Omega(H_n, S, \mathcal{A}_n)$ is the set of infection sequences compatible with the coloring imposed by $\mathcal{A}_n$. Therefore, the same argument as in the proof of (6) yields (7). The proof of the lemma is now complete.

Appendix B
Proof of Theorem 1

Let nodes that are infected by source s_i be colored with color i, with $i = 1, \ldots, k$. Then a partition $\mathcal{A}_n$ is a coloring of the graph H_n. For any infection sequence σ, and for any path in H_n connecting two infection sources, we can find the index of the last node on this path that is infected by either source. Let J_σ be the set of such indices. We have $\deg_{H_n}(\sigma_l) = 2$ for all l. We also have $N_\sigma(\sigma_l, H_n) = 1$ or 2 depending on whether l belongs to J_σ or not, respectively. From (5), we have

$$p_l(\sigma \mid H_n, S) = \left(\sum_{s \in S} \deg_{H_n}(s) - 2 \sum_{j \in J_\sigma} \mathbf{1}_{\{l \geq j\}} \right)^{-1}. \tag{25}$$

Choose two sources s_i and s_j and let m be the number of nodes in the path connecting s_i and s_j, excluding the source nodes. Suppose that $r > \lceil m/2 \rceil$ nodes in this path have color i. Construct a new coloring $\mathcal{A}'_n$ so that $\lceil m/2 \rceil$ nodes in $\rho(s_i, s_j)$ closest to s_i have color i and the rest have color j. The rest of the nodes in $\mathcal{A}'_n$ have the same colors as that in $\mathcal{A}_n$. Each infection sequence $\sigma \in \Omega(H_n, S, \mathcal{A}_n)$ corresponds to an infection sequence $\sigma' \in \Omega(H_n, S, \mathcal{A}'_n)$, where the last $x = r - \lceil m/2 \rceil$ color i nodes in σ become the last x color j nodes in σ'. From (25), we have $p_l(\sigma \mid H_n, S) = p_l(\sigma' \mid H_n, S)$ for all l. Since $\binom{m}{\lceil m/2 \rceil} \geq \binom{m}{r}$, we have $|\Omega(H_n, S, \mathcal{A}'_n)| \geq |\Omega(H_n, S, \mathcal{A}_n)|$, therefore Lemma 1 yields $\mathbb{P}(\mathcal{A}'_n \mid S, G_n) \geq \mathbb{P}(\mathcal{A}_n \mid S, G_n)$.

The same argument can be repeated a finite number of times for all paths in H_n connecting infection sources. This shows that the MAP estimator $\hat{\mathcal{A}}_n(S)$ is a Voronoi partition of G_n, and the proof is complete.

Appendix C
Proof of Lemma 3

To simplify notations, we write $T_u(s_1, s_2)$ as T_u, with the implicit understanding that all trees are defined w.r.t. $\{s_1, s_2\}$. The number of infection sequences can be found by counting the number of ways to form such a sequence. The $n-2$ slots in a sequence are occupied by nodes from $T_{s_i} \backslash \{s_i\}$, $i = 1, 2$, and $T_{\rho(u_1, u_m)}$. Therefore, we have

$$\begin{aligned} C(s_1, s_2 \mid G_n) &= (n-2)! \prod_{i=1}^{2} \frac{C(s_i \mid T_{s_i})}{(|T_{s_i}| - 1)!} \cdot \frac{R(u_1, u_m)}{|T_{\rho(u_1, u_m)}|!} \\ &= \frac{(n-2)!}{|T_{\rho(u_1, u_m)}|!} \prod_{\substack{v \in T_{s_i}, i=1,2 \\ v \neq s_1, s_2}} \frac{1}{|T_v|} \cdot R(u_1, u_m), \end{aligned}$$

where $R(u_i, u_j)$ for $i \leq j$ is the number of ways of permuting the nodes in $T_{\rho(u_i, u_j)}$ such that the infection sequence property is maintained, and the last equality follows from Lemma 2. In the following, we show that for $1 \leq i \leq j \leq m$,

$$R(u_i, u_j) = |T_{\rho(u_i, u_j)}|! \prod_{v \in T_{\rho(u_i, u_j)} \backslash \rho(u_i, u_j)} \frac{1}{|T_v|} \cdot q(u_i, u_j; s_1, s_2). \tag{26}$$

The proof proceeds by induction on $j - i$. If $j = i$, we have $R(u_i, u_i) = C(u_i \mid T_{u_i})$ and the claim follows from Lemma 2. Suppose that the claim (26) holds for all nodes u_k and u_p such that $p - k < j - i$. The number of permutations $R(u_i, u_i)$ can be computed by considering a sequence with $m = |T_{\rho(u_i, u_j)}|$ slots. The first slot can be filled with either u_i or u_j. Therefore, we have

$$\begin{aligned} R(u_i, u_j) &= (m-1)! \left(\frac{C(u_i \mid T_{u_i})}{(|T_{u_i}| - 1)!} \frac{R(u_{i+1}, u_j)}{|T_{\rho(u_{i+1}, u_j)}|!} + \frac{C(u_j \mid T_{u_j})}{(|T_{u_j}| - 1)!} \frac{R(u_i, u_{j-1})}{|T_{\rho(u_i, u_{j-1})}|!} \right) \\ &= (m-1)! \cdot (q(u_{i+1}, u_j; s_1, s_2) + q(u_i, u_{j-1}; s_1, s_2)) \prod_{v \in T_{\rho(u_i, u_j)} \backslash \rho(u_i, u_j)} \frac{1}{|T_v|}, \end{aligned}$$

where the last equality follows from the inductive hypothesis and Lemma 2. The claim (26) now follows from (13). Finally, (11) follows by an inductive argument using (13) and (14), which we omit. The proof is now complete.

Appendix D
Proof of Lemma 4

The proof follows easily from Theorems 5 and 6 of [17]. Consider the infection spreading along a path in G_n. Let $\Pi(t)$ be the counting process of the number of infected nodes in this path. The process $\Pi(t)$ consists of exponentially distributed arrivals with rate 1, and at most one arrival with rate 2 if the path is between the two infection sources. Let $\Pi_1(t)$ be a unit rate Poisson process corresponding to the rate 1 arrivals. Then $\Pi_1(t) \leq \Pi(t) \leq \Pi_1(t) + 1$. From Theorem 6 of [17], we have for any positive $\gamma < 0.2$,

$$\begin{aligned}\mathbb{P}(\Pi(t) \leq t(1-\gamma)) \leq \mathbb{P}(\Pi_1(t) \leq t(1-\gamma) - 1) \leq \exp\left(-\frac{1}{4}t(\gamma + \frac{1}{t})^2\right),\\ \mathbb{P}(\Pi(t) \geq t(1+\gamma)) \leq \mathbb{P}(\Pi_1(t) \geq t(1+\gamma)) \leq \exp\left(-\frac{1}{4}t\gamma^2\right).\end{aligned}$$

The rest of the proof is the same as that of Theorem 5 of [17], and the proof is complete.

Appendix E
Proof of Theorem 2

We first show that under (23), the interval (24) is non-empty. The condition (23) implies that

$$d_{\min} > \frac{3}{2} + \sqrt{2d_{\max}\frac{c^2}{b^2} - \frac{1}{4}},$$

which after some algebraic manipulations yields

$$\begin{aligned}b^2(d_{\min} - 1)(d_{\min} - 2) > 2c^2 d_{\max},\\ 1 \leq \frac{cd_{\max}}{b(d_{\min} - 1)} < \frac{b(d_{\min} - 2)}{2c}.\end{aligned}$$

Therefore (24) is a non-empty interval. Fix a δ in the interval. Then for all $\epsilon > 0$ sufficiently small, we have

$$\begin{aligned}\frac{b(d_{\min} - 1)(1+\delta)}{cd_{\max}} > \frac{1}{1-\epsilon},\\ \frac{b(d_{\min} - 2)}{2(1+\delta)c} > \frac{1}{1-\epsilon}.\end{aligned}$$

Recall that t is the time from the start of the infection spreading to our observation of G_n. From Lemma 4, for each ϵ, there exists t_0 such that if $t \geq t_0$, we have

$$\frac{(d_{\min} - 1)(1+\delta)N_{\min}(t)}{d_{\max}N_{\max}(t)} > 1, \tag{27}$$

$$\frac{(d_{\min} - 2)N_{\min}(t)}{2(1+\delta)N_{\max}(t)} > 1. \tag{28}$$

We will make use of the two inequalities (27) and (28) extensively in the following proof steps. Let $\mathcal{E}_t$ be the event defined in Lemma 4. Then from Lemma 4, we have for $t \geq t_0$,

$$\mathbb{P}(\tilde{S} = S \mid S) \geq \mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t)\mathbb{P}(\mathcal{E}_t \mid S) \geq (1-\epsilon)\mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t). \tag{29}$$

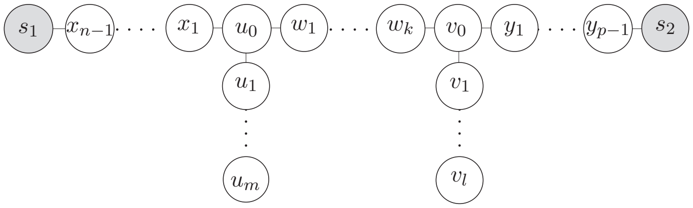


Fig. 7. Illustration of the network structure when $u_0 \neq v_0$. Not all nodes are shown.

In the following, we show that $\mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t) = 1$ for $t \geq t_0$. The proof then follows from (29) as ϵ can be chosen arbitrarily small.

To show that $\mathbb{P}(\tilde{S} = S \mid S, \mathcal{E}_t) = 1$ is equivalent to showing that with probability one, $\tilde{C}(S \mid G_n) > \tilde{C}(u_m, v_l \mid G_n)$, for all node pairs $u_m, v_l \in G_n$ such that at least one of them is not in S. Let u_0 and v_0 be the first nodes in $\rho(s_1, s_2)$ that are connected to u_m and v_l respectively. We divide the proof into two cases, depending on whether u_0 and v_0 are distinct or not, as shown in Figures 7 and 8.

Suppose that $u_0 \neq v_0$. A typical network for this case is shown in Figure 7, where m, l, n, p, and k are non-negative integers, and at least one of u_m and v_l is not in S, i.e., either $m + l > 0$ or $n + p > 0$. We let $u_0 = s_1$ if $n = 0$, and $v_0 = s_2$ if $p = 0$.

We will show that if either $m + l > 0$ or $n + p > 0$, we have for $t \geq t_0$,

$$\frac{\tilde{C}(s_1, s_2 \mid G_n)}{\tilde{C}(u_m, v_l \mid G_n)} = \frac{\tilde{C}(s_1, s_2 \mid G_n)}{\tilde{C}(u_0, v_0 \mid G_n)} \cdot \frac{\tilde{C}(u_0, v_0 \mid G_n)}{\tilde{C}(u_m, v_l \mid G_n)} > 1. \tag{30}$$

The proof follows by showing that $\tilde{C}(u_0, v_0 \mid G_n) \geq \tilde{C}(u_m, v_l \mid G_n)$, where strict inequality holds if $m + l > 0$, and $\tilde{C}(s_1, s_2 \mid G_n) \geq \tilde{C}(u_0, v_0 \mid G_n)$ with strict inequality holding if $n + p > 0$. From (22), we have [5]

$$\begin{aligned}
\frac{\tilde{C}(u_0, v_0 \mid G_n)}{\tilde{C}(u_m, v_l \mid G_n)} &= \frac{Q(u_0, v_0)}{Q(u_m, v_l)} \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\
&= [2(1+\delta)]^{-(m+l)} \cdot \frac{\prod_{i=1}^{m+l+k+2} I_i^*(u_m, v_l)}{\prod_{i=1}^{k+2} I_i^*(u_0, v_0)} \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\
&\geq [2(1+\delta)]^{-(m+l)} \cdot \prod_{i=1}^{m+l} I_i^*(u_m, v_l) \cdot \prod_{w \in \rho(u_m, u_1) \cup \rho(v_l, v_1)} |T_w(u_0, v_0)|^{-1} \\
&\geq \left[\frac{\max\{|T_{u_0}(u_m, v_l)|, |T_{v_0}(u_m, v_l)|\}}{2(1+\delta) \cdot \max\{|T_{u_1}(u_0, v_0)|, |T_{v_1}(u_0, v_0)|\}} \right]^{m+l} \\
&\geq \left[\frac{(d_{\max} - 2) N_{\min}(t) + 1}{2(1+\delta) \cdot N_{\max}(t)} \right]^{m+l} \\
&> 1,
\end{aligned}$$

if $m + l > 0$. The first inequality follows because $I_{m+l+i}^*(u_m, v_l) \geq I_i^*(u_0, v_0)$ for $i = 1, \ldots, k+2$, and the last inequality follows from (28) when $t \geq t_0$.

[5] We define products over empty sets to be 1.

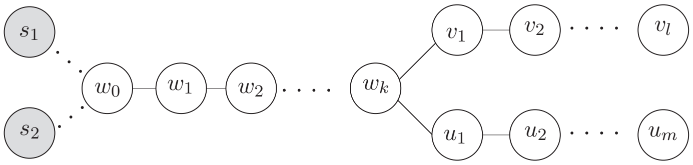


Fig. 8. Illustration of the case where $u_0 = v_0 = w_0$.

Let $\psi = \deg_G(s_1) + \deg_G(s_1)$. We have for $t \geq t_0$,

$$\begin{aligned}
\frac{\tilde{C}(s_1, s_2 \mid G_n)}{\tilde{C}(u_0, v_0 \mid G_n)} &= \frac{Q(s_1, s_2)}{Q(u_0, v_0)} \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\
&= [2(1+\delta)]^{n+p} \cdot \frac{\prod\limits_{i=1}^{k+2} I_i^*(u_0, v_0)}{\prod\limits_{i=1}^{n+p+k+2} I_i^*(s_1, s_2)} \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\
&\geq [2(1+\delta)]^{n+p} \cdot \prod_{i=k+3}^{n+p+k+2} I_i^*(s_1, s_2)^{-1} \cdot \prod_{w \in \rho(s_1, x_1) \cup \rho(y_1, s_2)} |T_w(u_0, v_0)| \\
&\geq \left[\frac{2(1+\delta) \cdot \min\left\{ |T_{s_1}(u_0, v_0)|, |T_{s_2}(u_0, v_0)| \right\}}{\psi N_{\max}(t) + 2} \right]^{n+p} \\
&\geq \left[\frac{(1+\delta)(d_{\min} - 1) \cdot N_{\min}(t) + 1 + \delta}{d_{\max} N_{\max}(t) + 1} \right]^{n+p} \\
&> 1,
\end{aligned}$$

where the first inequality follows because $I_i^*(u_0, v_0) \geq I_i^*(s_1, s_2)$ for $i = 1, \ldots, k+2$, and the last inequality follows from (27) if $n + p > 0$. The bound (30) is now proved.

We next consider the case where $u_0 = v_0 = w_0$ in Figure 8, where k, m and l are non-negative integers. When $t \geq t_0$, we have the following bounds, which are straight forward to verify and whose proofs are omitted here.

(i) $I_i^*(u_m, v_l) \geq (\psi - 2) N_{\min}(t) + 2 \geq (d_{\min} - 2) N_{\min}(t)$ for $i = 1, \ldots, d(u_m, v_l) + 1$,

(ii) $I_i^*(s_1, s_2) \leq \psi N_{\max}(t) + 2 \leq 2 d_{\max} N_{\max}(t) + 2$ for all $i = 1, \ldots, d(s_1, s_2) + 1$,

(iii) $|T_{w_i}(u_m, v_l)| \geq (\psi - 2) N_{\min}(t) + 2 \geq (d_{\min} - 2) N_{\min}(t)$ for all $i = 1, \ldots, k-1$,

(iv) $|T_w(u_m, v_l)| \geq (d_{\min} - 1) N_{\min}(t) + 1$ for all $w \in \rho(s_1, s_2)$,

(v) $|T_{w_i}(s_1, s_2)| \leq N_{\max}(t)$ for all $i = 1, \ldots, k-1$, and

(vi) $|T_w(s_1, s_2)| \leq N_{\max}(t)$ for all $w \in \rho(u_m, v_l)$.

The above bounds yield

$$\begin{aligned}
&\frac{\tilde{C}(s_1, s_2 \mid G_n)}{\tilde{C}(u_m, v_l \mid G_n)} \\
=& \frac{Q(s_1, s_2)}{Q(u_m, v_l)} \frac{\prod_{w \in G_n \backslash \rho(u_m, v_l)} |T_w(u_m, v_l)|}{\prod_{w \in G_n \backslash \rho(s_1, s_2)} |T_w(s_1, s_2)|} \\
=& (2(1+\delta))^{d(s_1, s_2) - d(u_m, v_l)} \frac{\prod_{i=1}^{d(u_m, v_l)+1} I_i^*(u_m, v_l)}{\prod_{i=1}^{d(s_1, s_2)+1} I_i^*(s_1, s_2)} \frac{\prod_{i=1}^{k-1} |T_{w_i}(u_m, v_l)| \prod_{w \in \rho(s_1, s_2)} |T_w(u_m, v_l)|}{\prod_{i=1}^{k-1} |T_{w_i}(s_1, s_2)| \prod_{w \in \rho(u_m, v_l)} |T_w(s_1, s_2)|}
\end{aligned}$$

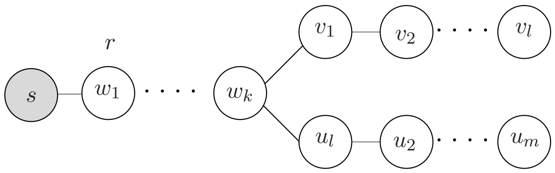


Fig. 9. A typical network for a single source tree.

$$
\begin{aligned}
&=\prod_{i=1}^{k-1}\frac{|T_{w_i}(u_m,v_l)|}{|T_{w_i}(s_1,s_2)|}\cdot(2(1+\delta))^{-d(u_m,v_l)-1}\frac{\prod_{i=1}^{d(u_m,v_l)+1}I_i^*(u_m,v_l)}{\prod_{w\in\rho(u_m,v_l)}|T_w(s_1,s_2)|}\cdot(2(1+\delta))^{d(s_1,s_2)+1}\frac{\prod_{w\in\rho(s_1,s_2)}|T_w(u_m,v_l)|}{\prod_{i=1}^{d(s_1,s_2)+1}I_i^*(s_1,s_2)}\\
&\geq\left[\frac{(d_{\min}-2)N_{\min}(t)}{N_{\max}(t)}\right]^{k-1}\left[\frac{(d_{\min}-2)N_{\min}(t)}{2(1+\delta)N_{\max}(t)}\right]^{d(u_m,v_l)+1}\left[\frac{(1+\delta)((d_{\min}-1)N_{\min}(t)+1)}{d_{\max}N_{\max}(t)+1}\right]^{d(s_1,s_2)+1}\\
&>1,
\end{aligned}
$$

where the last inequality follows from (27) and (28). The theorem is now proved.

Appendix F
Proof of Theorem 3

Let t be the elapsed time from the start of an infection spreading from a single s to the time we observe G_n. We wish to show that Algorithm TSE estimates as sources s and one of its neighbors with probability (conditioned on s being the infection source) converging to 1 as $t\to\infty$. This is equivalent to showing that for t sufficiently large, and for each pair of nodes $u_m,v_l\in G_n$ where either $d(u_m,s)>1$ or $d(v_l,s)>1$, there exists a neighbor r of s such that $\tilde{C}(s,r\mid G_n)>\tilde{C}(u_m,v_l\mid G_n)$.

A typical network is shown in Figure 9, where k,m and l are non-negative integers. If m,l and k are positive, we let r be the neighbor of s that lies on the path connecting s to u_m (i.e., the node w_1 in Figure 9). If m and l are positive and $k=0$, then r is chosen to be either u_1 or v_1. If $m=0$, we must have $k>0$ so that $w_k=u_m$ and $r=w_1$. A similar remark applies for the case $l=0$. Note that $m+l>0$. For t sufficiently large, we have

$$
\begin{aligned}
\frac{\tilde{C}(s,r\mid G_n)}{\tilde{C}(u_m,v_l\mid G_n)}&=\frac{Q(s,r)}{Q(u_m,v_l)}\cdot\frac{\prod\limits_{w\in G_n\backslash\rho(u_m,v_l)}|T_w(u_m,v_l)|}{\prod\limits_{w\in G_n\backslash\{s,r\}}|T_w(s,r)|}\\
&=[2(1+\delta)]^{1-(m+l)}\cdot\frac{\prod_{i=1}^{m+l+1}I_i^*(u_m,v_l)}{\prod_{i=1}^{2}I_i^*(s,r)}\cdot\frac{\prod_{w\in\rho(s,w_{k-1})}|T_w(u_m,v_l)|}{\prod_{i=2}^{k-1}|T_{w_i}(s,r)|\cdot\prod\limits_{w\in\rho(u_m,v_l)}|T_w(s,r)|}\\
&=[2(1+\delta)]^{1-(m+l)}\cdot\prod_{i=1}^{m+l}I_i^*(u_m,v_l)\cdot\frac{\prod_{i=1}^{k-1}|T_{w_i}(u_m,v_l)|}{\prod_{i=2}^{k-1}|T_{w_i}(s,r)|\cdot\prod\limits_{w\in\rho(u_m,v_l)}|T_w(s,r)|}\\
&\geq[2(1+\delta)]^{1-(m+l)}\cdot|T_{w_k}(u_m,v_l)|^{m+l}\cdot\frac{|T_s(u_m,v_l)|^{k-1}}{N_{\max}(t)^{k-2}\cdot N_{\max}(t)^{m+l+1}}\\
&\geq[2(1+\delta)]^{k}\cdot\left[\frac{(d_{\min}-1)N_{\min}(t)}{2(1+\delta)\cdot N_{\max}(t)}\right]^{m+l+k-1}\\
&>1,
\end{aligned}
$$

where the last inequality follows from (28) and Lemma 4 for graphs with a single infection source [17]. The proof of the theorem is now complete.

References

[1] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in Twitter: the million follower fallacy," in *Proc. 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[2] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proc. 3rd ACM International Conference on Web Search and Data Mining*, 2010, pp. 261–270.

[3] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on Twitter," in *Proc. 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 65–74.

[4] D. Gruhl, R. Guha, D. L. Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proc. 13th International Conference on World Wide Web*, 2004, pp. 491–501.

[5] A. Java, P. Kolari, T. Finin, and T. Oates, "Modeling the spread of influence on the blogosphere," in *WWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006.

[6] L. Lu and F. Zhu, "Discovering the important bloggers in blogspace," in *Proc. Int Artificial Intelligence and Education (ICAIE) Conf*, 2010, pp. 151–154.

[7] S.-H. Lim, S.-W. Kim, S. Park, and J. H. Lee, "Determining content power users in a blog network: an approach and its applications," *IEEE Trans. Syst., Man, Cybern. A*, vol. 41, no. 5, pp. 853–862, 2011.

[8] L. Akritidis, D. Katsaros, and P. Bozanis, "Identifying the productive and influential bloggers in a community," *IEEE Trans. Syst., Man, Cybern. C*, vol. 41, no. 5, pp. 759–764, 2011.

[9] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," in *Proc. IEEE Computer Society Symp Research in Security and Privacy*, 1991, pp. 343–359.

[10] L. Han, S. Han, Q. Deng, J. Yu, and Y. He, "Source tracing and pursuing of network virus," in *Proc. 8th IEEE International Conference on Computer and Information Technology Workshops*, 2008, pp. 230–235.

[11] C. Scoglio, W. Schumm, P. Schumm, T. Easton, S. Roy Chowdhury, A. Sydney, and M. Youssef, "Efficient mitigation strategies for epidemics in rural regions," *PLoS ONE*, vol. 5, no. 7, 2010.

[12] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004.

[13] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 137–146.

[14] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proce. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 199–208.

[15] H. Liu and N. Agarwal, *Modeling and Data Mining in Blogosphere*. Morgan and Claypool Publishers, 2009.

[16] Y. Zhang, Z. Wang, and C. Xia, "Identifying key users for targeted marketing by mining online social network," in *Proc. IEEE 24th Int Advanced Information Networking and Applications Workshops (WAINA) Conf*, 2010, pp. 644–649.

[17] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[18] N. Bailey, *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, 1975.

[19] T. Zhao and A. Nehorai, "Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, 2007.

[20] E. B. Fox, J. W. Fisher, and A. S. Willsky, "Detection and localization of material releases with sparse sensor configurations," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1886–1898, 2007.

[21] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Bayesian detection in bounded height tree networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 4042–4051, 2009.

[22] ——, “On the impact of node failures and unreliable communications in dense sensor networks,” *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2535–2546, 2008.

[23] P. Bianchi, M. Debbah, M. Maida, and J. Najim, “Performance of statistical tests for single-source detection using random matrix theory,” *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2400–2419, 2011.

[24] S. Aldalahmeh and M. Ghogho, “Robust distributed detection, localization and estimation of a diffusive target in clustered wireless sensor networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011.

[25] M. Kimura, K. Saito, R. Nakano, and H. Motoda, “Extracting influential nodes on a social network for information diffusion,” *Data Min. Knowl. Discov.*, vol. 20, pp. 70–97, 2010.

[26] C. Moore and M. E. J. Newman, “Epidemics and percolation in small-world networks,” *Phys. Rev. E*, vol. 61, pp. 5678–5682, 2000.

[27] P. D. ONeill, “A tutorial introduction to Bayesian inference for stochastic epidemic models using Markov chain Monte Carlo methods,” *Mathematical Biosciences*, vol. 180, no. 1-2, pp. 103 – 114, 2002.

[28] G. Brightwell and P. Winkler, “Counting linear extensions is #P-complete,” in *Proc. 23rd Annual ACM Symposium on Theory of Computing*, 1991, pp. 175–181.

[29] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks.” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[30] K.-T. Goh, J. Cutter, B.-H. Heng, S. Ma, B. K. W. Koh, C. Kwok, C.-M. Toh, and S.-K. Chew, “Epidemiology and control of SARS in Singapore.” *Annals Of The Academy Of Medicine Singapore*, vol. 35, no. 5, pp. 301–316, 2006.

[31] G. Sabidussi, “The centrality index of a graph,” *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[32] G. Hu, “Robust consensus tracking for an integrator-type multi-agent system with disturbances and unmodelled dynamics,” *International Journal of Control*, vol. 84, no. 1, pp. 1–8, 2011.

[33] ——, “Robust consensus tracking of a class of second-order multi-agent dynamic systems,” *Systems and Control Letters*, vol. 61, no. 1, pp. 134–142, 2012.

[34] Y. Wen, G. Shi, and G. Wang, “Designing an inter-cloud messaging protocol for content distribution as a service (CoDaas) over future internet,” in *International Conference on Future Internet Technologies*, 2011.

[35] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. Vasilakos, “CoDaaS: An experimental cloud-centric content delivery platform for user-generated contents,” in *IEEE International Conference on Computing, Networking and Communications*, 2012.