Estimating Infection Sources in Networks Using Partial Timestamps

Wenchang Tang, Student Member, IEEE, Feng Ji, and Wee Peng Tay, Senior Member, IEEE

Abstract—We study the problem of identifying infection sources in a network based on the network topology, and a subset of infection timestamps. In the case of a single infection source in a tree network, we derive the maximum likelihood estimator of the source and the unknown diffusion parameters. We then introduce a new heuristic involving an optimization over a parametrized family of Gromov matrices to develop a single source estimation algorithm for general graphs. Compared with the breadth-first search tree heuristic commonly adopted in the literature, simulations demonstrate that our approach achieves better estimation accuracy than several other benchmark algorithms, even though these require more information like the diffusion parameters. We next develop a multiple sources estimation algorithm for general graphs, which first partitions the graph into source candidate clusters, and then applies our single source estimation algorithm to each cluster. We show that if the graph is a tree, then each source candidate cluster contains at least one source. Simulations using synthetic and real networks, and experiments using real-world data suggest that our proposed algorithms are able to estimate the true infection source(s) to within a small number of hops with a small portion of the infection timestamps being observed.

Index Terms—Infection source, rumor source, single source estimation, multiple sources estimation, diffusion process, infection timestamps, Gromov product

I. INTRODUCTION

Online social networks such as Facebook, Twitter and Sina Weibo have grown immensely in recent decades. A rumor or piece of news can be shared and forwarded with high speed through friendship circles [1]. Similar propagation phenomena in complex networks are ubiquitous in nature and modern society. For example, viruses propagate throughout the Internet and infect millions of computers or mobile phones [2], [3]. Disease epidemics can quickly spread in human society [4], [5]. We regard such rumors, viruses and epidemics as infections, and the propagation initiated by an infection source can be modeled using a diffusion or infection process in networks [5], [6]. Such infections may incur massive losses to society [7], [8]. In view of this, it is often important to be able to accurately and promptly identify the infection sources, so that proper control measures can be adopted. In this paper, we consider the infection source estimation problem based on knowledge of the network topology, and a subset of infection timestamps.

A. Related Works

Several works have tackled the problem of infection source estimation under different assumptions. Many of these works are based on the knowledge of network topology and the infection status of a portion of nodes in the network. In the literature, various spreading models have been proposed including the Susceptible-Infected (SI) [9], Susceptible-Infected-Recovered (SIR) [10], and Susceptible-Infected-Susceptible (SIS) [11] models. Specifically, in the SI model, an infected node remains infected forever; in the SIR model, it can recover and cannot be further infected; and in the SIS model, a recovered node can become infected again. A rumor centrality estimator under the SI model was developed in [9], [12], while [13]–[16] developed estimators for identifying multiple infection sources under the SI model. The paper [17] considers infection source estimation when only a subset of infected nodes are observed. In [18], multiple observations of a SI spreading process are used for source estimation. In [10] and [19], the problem of infection source estimation is investigated under the SIR model, and [11] considers the SIS model. All the aforementioned works perform source estimation based on the observed status of the nodes, and the network topology. Other related works assume additionally a priori knowledge like the infection spreading rate, including [20] which performs source inference via belief propagation, and [21] which developed a dynamic message passing algorithm. Recently, [22] considered the alternative Independent Cascade (IC) model for infection spreading, and proposed the Short-Fat Tree (SFT) algorithm for source identification under the IC model. The reference [23] developed a framework for optimal source estimation in arbitrary weighted networks with arbitrary distribution of sources by combining controllability theory and compressive sensing. The reference [24] considered the case where additional relative information about the infection times of a fraction of node pairs is also available. Finally, the works [25], [26] have studied the complementary problem of source obfuscation and proposed messaging protocols which can spread the infection as widely as possible while protecting the anonymity of the source.

Infection source estimation has also been investigated using infection timestamps. In this framework, we make use of observations of the first infection times of a subset of nodes in the network. In [27], an algorithm based on the maximum likelihood estimator (MLE) is proposed. Two ranking algorithms using a modified breadth-first search (BFS) tree heuristic for general graphs have been proposed in [28], which demonstrate improved estimation accuracy compared to the
algorithm in [27]. In [29], a two-stage algorithm was proposed to locate a single source in large networks. The reference [30] proposed a sequential source estimation algorithm that allows online update of the source estimate as timestamps are observed sequentially. In [31], a time-reversal backward spreading (TRBS) algorithm was proposed to infer a single source in a weighted network. The papers [32], [33] discuss the selection of observer nodes under the deterministic slotted SI model, so as to achieve low probability of error detection. The problem of multiple sources estimation with infection timestamps is less studied. In [34], a backward diffusion based method was proposed. However, the paper assumes that all infection sources initiate diffusion at the same start time, and the mean of the infection propagation delays along all edges and the number of sources are known. Such assumptions can be restrictive for practical applications.

B. Our Contributions

We first study the single source estimation problem with less assumptions compared with the works [27], [29], [30]. Similar to [27], [29], [30], we assume that the propagation delay along each edge can be modeled using a Gaussian distribution. However, unlike these references, the diffusion parameters, including the mean and variance of the infection propagation delay, are assumed to be unknown. When the graph is a tree, we derive the optimal MLE that simultaneously estimates the source and diffusion parameters. It is common in the literature [9], [27], [29], [35] to use a breadth-first search (BFS) tree heuristic to generalize an algorithm for identifying a single source in a tree to one for a general graph: for each node in the general graph, one constructs a BFS tree and assumes that any infection starting at this node diffuses along this BFS tree. In this paper, we introduce a novel heuristic, which involves finding an optimal convex combination of two BFS trees and a target covariance matrix (cf. Section III-B) for each node to approximate the diffusion process better, before generalizing our algorithm for trees to general graphs. We call this the Gromov Single Source Identification (GSSI) algorithm. Simulation results demonstrate that GSSI performs better than algorithms proposed in [27], [31] although GSSI requires less information.

We also study the case of multiple sources and propose a Source Candidate Clustering and Estimation (SCCE) algorithm, which makes less assumptions compared with the work [34] as we do not assume that infection sources have the same diffusion parameters, and the number of sources is also unknown a priori. SCCE can be divided into two steps. In the first step, we reduce the multiple sources estimation problem to the single source estimation problem by partitioning the graph into source candidate clusters. We show that if the graph is a tree, then each cluster contains at least one source. We then apply a procedure to estimate the number of sources in each cluster and estimate the infection region of each source. In the second step, we apply the GSSI algorithm in each region. Simulation results demonstrate that our approach performs better than the algorithm proposed in [34], and is able to accurately infer multiple sources with a small portion of timestamps.

The rest of this paper is organized as follows. In Section II, we formulate the infection source estimation problem and provide an interpretation based on Gromov matrices. In Section III, we first derive the MLE for estimating a single source in a tree, and then present our GSSI algorithm using Gromov matrices for identifying a source in a general graph. Simulations on both synthetic and real-world networks are also provided to compare the performance of GSSI with various other algorithms in the literature. In Section IV, we investigate the problem of multiple sources estimation, present our SCCE algorithm, and provide simulation results to verify its performance. Section V concludes the paper.

Notations: We use $\mathcal{N}(\mu, \sigma^2)$ to denote the Gaussian distribution with mean $\mu$ and variance $\sigma^2$, $\mathcal{N}(\mu, \Lambda)$ to denote the multivariate Gaussian distribution with mean vector $\mu$ and covariance matrix $\Lambda$. For a set $X$, its cardinality is given by $|X|$. Given a matrix $A$, we use $[A]_{i,j}$ to denote its $(i,j)$-th entry. We write $A^\top$ for the transpose of $A$, $A^{-1}$ for its inverse, $\text{tr}(A)$ for its trace, and $\det A$ for its determinant. For two nodes $u$ and $v$ in a tree, we let $[u,v]$ denote the path from $u$ to $v$, $(u,v)$ the path with the end node $v$ excluded, and so on. $E[\cdot]$ is the expectation operator. The symbol $\sim$ means equality in distribution.

II. Problem Formulation

We model the network as an undirected graph $G=(V,E)$, where $V$ is the set of nodes and $E$ is the set of edges. Suppose a collection of infection sources $S = \{s_1, \ldots, s_m\} \subseteq V$ initiate a diffusion process at unknown start times on $G$. The infection propagates from an infected node to its neighbors stochastically along the edges connecting them. Let $\tau_{uv}$ be the random propagation delay associated with the edge connecting nodes $u$ and $v$. For example, if $u$ is first infected at time $t_u$, then the infection propagating from $u$ reaches $v$ at time $t_u + \tau_{uv}$. Similar to [27], [29], [30], we assume that the propagation delays $\{\tau_{uv}\}$ for all pairs of adjacent nodes $u,v$ are independent and identically distributed (i.i.d.) continuous random variables that follow a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$.\footnote{The Gaussian distribution is adopted for technical convenience although its realizations are not necessarily positive. In our simulations, we will generate the propagation delays using a truncated Gaussian distribution instead.} However, we also assume that the parameters $(\mu, \sigma^2)$ are unknown, and need to be estimated.

We assume the diffusion follows the SI model: any infected node never recovers. Suppose that we observe a collection of infected nodes $V = \{v_1, \ldots, v_n\}$, which is a subset of all the infected nodes, and the vector of their associated infection timestamps $T = [t_1, \ldots, t_n]^T$. Our goal is to estimate $S$ using $(V, T)$. To simplify our exposition and avoid non-invertible matrices in our cost functions in (2), (3) and (9), we assume that $S \cap V = \emptyset$. There is no loss of generality making this assumption since one can generalize our methods by adding an additional step in which each vertex in $V$ is considered a potential source candidate, computing an appropriate variant of the respective cost functions in (2), (3) and (9) for these potential source nodes, and then choosing the node from amongst $V$ and that estimated by our method with the lowest cost to be the final estimate.
In this paper, we consider both the single source (|S| = 1) and multiple sources (|S| > 1) estimation problems. In the case where |S| > 1, we assume that |S| is unknown.

A. A Gromov Matrix Interpretation

Let d(u, v) be the length of a shortest path between u and v in G (assuming each edge has length 1). For two vertices u, v in a tree T ⊂ G, let d_T(u, v) be the length of the unique path between u and v. For a node s of the tree T, the Gromov product (Definition 2.6 of [36]) of u and v in T with respect to (w.r.t.) s is given by

\[ (u, v)_s = \frac{1}{2} (d_T(u, s) + d_T(v, s) - d_T(u, v)). \]  

(1)

The following lemma summarizes some simple properties of the Gromov product.

**Lemma 1.** Let s, u, and v be nodes of a tree T.

(i) If u = v, then (u, v)_s = d_T(u, s).

(ii) (·, ·)_s is symmetric and non-negative.

(iii) (u, v)_s = 0 if and only if s is on the unique path [u, v].

(iv) (u, v)_s ≤ d_T(u, s).

**Proof:** Both claim (i) and claim (ii) follow immediately from definition (and the triangle inequality). For claim (iii), (u, v)_s = 0 if and only if d(u, s) + d(v, s) = d(u, v). On a tree, this happens only when s ∈ [u, v]. Claim (iv) clearly holds if (u, v)_s = 0. If (u, v)_s > 0, from claim (iii), there exists a vertex w ∈ [u, v] such that [u, s] ∩ [u, v] = [u, w] and [w, s] ∩ [u, v] = [w, v]. Then,

\[ (u, v)_s = \frac{1}{2} (d_T(u, w) + d_T(w, s) + d_T(v, w) + d_T(w, s)) - d_T(u, w) - d_T(w, v) \]

\[ = d_T(u, s) \leq d_T(u, s), \]

and the proof is complete.

Let U = \{∪_1, ..., u_n\}⊂ V be a subset of vertices of G, s ∉ U be a vertex of G, and T be a tree that spans U ∪ \{s\} (i.e., U ∪ \{s\} is a subset of the vertex set of T). We define an n × n matrix Λ whose (i, j)-th entry is the Gromov product (u_i, u_j)_s in T. We call Λ the Gromov matrix with base (T, s, U), where T is known as its base tree and s its base vertex.

Recall that two trees T_1 and T_2 are said to be isometric to each other if there is a bijection φ : T_1 → T_2 such that d_{T_1}(u, v) = d_{T_2}(φ(u), φ(v)) for any u, v ∈ T_1. The map φ is called an isometry. We say that the two bases (T_1, s_1, U_1) and (T_2, s_2, U_2) are isometrically equivalent to each other if there is an isometry φ : T_1 → T_2 such that φ(s_1) = s_2 and φ(U_1) = U_2.

**Proposition 1.** Suppose that Λ is a Gromov matrix, then its base (T, s, U) is uniquely determined, up to isometric equivalence.

**Proof:** We assume that (T_1, s_1, U_1) and (T_2, s_2, U_2) are two bases of Λ. We prove that they are isometrically equivalent to each other by induction on |U_1| = |U_2|. The statement is clearly true if |U_1| = 1. Indeed, the condition that T_1 spans U_1 \{s_i\}, for i = 1, 2 imply that both T_1 and T_2 are simple paths, with end points s_i, and the single vertex in U_i, respectively. Therefore, (T_1, s_1, U_1) and (T_2, s_2, U_2) are equivalent as T_1 and T_2 have the same length given by Λ (which is a 1 × 1 matrix).

Suppose the statement holds for |U_1| = |U_2| = n + 1. Let Λ' be the n × n upper-left block of Λ. Since Λ is a Gromov matrix, so is Λ'. For each i = 1, 2, there is a subset U_i ⊂ U_i with |U_i'| = n, and T_i' ⊂ T_i such that (T_i', s_i, U_i') is a base of Λ'. By the induction hypothesis, (T_i', s_i, U_i'), i = 1, 2 are equivalent to each other via an isometry φ_i' : T_i' → T_i'. Without loss of generality, for i = 1, 2, let U_i = \{u_i,k : k = 1, ..., n + 1\} such that φ_i(u_{i,k}) = u_{i,k}, for k = 1, ..., n.

Let 1 ≤ j ≤ n be an index such that the (j, n + 1)-th entry \[|A_i|_{j,n+1}\] of Λ satisfy \[|A_i|_{j,n+1} ≥ |A_i|_{k,n+1} \] for all 1 ≤ k ≤ n. From Lemma 1(iv), for each i = 1, 2, we can define the unique vertex on the path from u_{i,j} to s_i such that d_{T_i'}(u_{i,j}, s_i) = |A_i|_{j,n+1}. As φ_i is an isometry and φ_i(s_i) = s_2, φ_i(u_{i,j}) = u_{i,j}, we have φ(u_{i,j}) = u_{2,j}.

We claim that the path \[u_{i,n+1}, s_i\] in \[T_i\] satisfies \[u_{i,n+1}, s_i \cap T_i' = [w_i, s_i].\] To see this, by the choice of \[w_i\], we have

\[w_i, s_i = [u_{i,n+1}, s_i] \cap [u_{i,j}, s_i] \subset [u_{i,n+1}, s_i] \cap T_i'.\]

On the other hand, the maximality on \[|A_i|_{j,n+1}\] implies that \[u_{i,n+1}, s_i \cap T_i'\] cannot be a path longer than \[w_i, s_i\]. Hence, we have \[u_{i,n+1}, s_i \cap T_i' = [w_i, s_i].\]

Therefore, \([T_i \setminus T_i'] \cup \{w_i\} = [w_i, u_{i,n+1}]\) for both \(i = 1, 2\). Moreover, both \([w_1, u_{1,n+1}]\) and \([w_2, u_{2,n+1}]\) are of the same length \(|A_i|_{n+1,n+1} = |A_i|_{j,n+1}\). Therefore, \(\phi_i\) can be extended to an isometry \(φ : T_1 → T_2\) such that \(φ(u_{1,n+1}) = u_{2,n+1}\). The induction is now complete and the proposition is proved.

For a base \((T, s, U)\), as the ordering of the vertices in the set \(U\) is not fixed, its Gromov matrix Λ is determined uniquely up to conjugation by permutation matrices of size \(|U|\). With this observation and Proposition 1, we conclude that Λ gives the same amount of information as \((T, s, U)\). Now we can describe a reformulation of the problem and the basic idea of our approach.

Since the propagation delays along each edge are continuous random variables, with probability one, the infection diffusion on a graph from a source \(s\) forms a propagation path that is a spanning tree of all the nodes infected by \(s\). Therefore, in the source inference task, in addition to the source \(s\), we are implicitly required to find a \(T\) in the set of spanning subtrees in \(G\) of the observed infected set \(V\) rooted at \(s\). If \(G\) is a tree, the source \(s\) uniquely determines \(T\). However, in a general graph, there may be more than one subtree rooted at \(s\) that spans the observed infected nodes. Therefore, finding \((s, T)\) together is no longer equivalent to finding \(s\) alone.

In a dense graph, it is usually intractable to identify all possible \(s\) and spanning subtrees rooted at \(s\). On the other hand, the results in this section imply that to identify \(s\) and the infection propagation tree \(T\) is equivalent to estimating the Gromov matrix associated with the infection diffusion process. In this paper, we formulate the source estimation problem as
an optimization of a cost function of a parametrized Gromov matrix and the diffusion parameters \((\mu, \sigma^2)\). By optimizing over a parametrized family of Gromov matrices, we overcome the intractability of identifying all possible Gromov matrices or infection propagation paths. However, the base tree of the optimal Gromov matrix found may not correspond to an actual subtree in \(G\). Our proposed approach is therefore a heuristic in the case of general graphs. A widely used heuristic in the source inference literature \([9, 27, 29, 35]\) for general graphs is the BFS heuristic, which assumes that the infection propagation path is a BFS tree, which means that the infection is spread from a source to each observed infected node along a minimum-length path. However, this approximation has many limitations. For example, in general there exists more than one BFS tree rooted at a node, and the BFS heuristic typically chooses only one BFS tree \([9, 27, 29, 35]\). Furthermore, the actual infection tree is not likely to be a BFS tree, especially when the graph is dense or \(\sigma^2/\mu\) is large. In our Gromov matrix approach, some BFS trees are included in the family of Gromov matrices that we optimize over. Therefore, our method is expected to achieve better performance than the BFS heuristic.

### III. Single Source Estimation

In this section, we consider the single source case, i.e., \(S = \{s_1\}\). We first derive an MLE that simultaneously estimates the source and diffusion parameters when the network is a tree. We then describe our source estimation approach using Gromov matrices for general graphs. Finally, simulations on both synthetic and real-world networks are provided to compare the performance of our proposed algorithm with several other timestamp-based source estimation algorithms in the literature.

#### A. Preliminaries

We first consider the special case where \(G\) is a tree. Recall that \(V = \{v_1, \ldots, v_n\}\) is a set of observed infected nodes with infection timestamps \(T\). Suppose that \(s_1\) is the source node. Let \(t_0\) be the time the source node starts its infection spreading. Then, it is easy to see that the infection time of each node \(v_k \in V\) is \(t_k \sim \mathcal{N}(t_0 + d(s_1, v_k)\mu, d(s_1, v_k)\sigma^2)\), since all propagation delays \(\{\tau_{uv}\}\) are i.i.d. \(\mathcal{N}(\mu, \sigma^2)\) variables. The covariance of \(t_i\) and \(t_j\) is \(|A_{s_1}|_{i,j} \cdot \sigma^2\), with \(A_{s_1}\) being the Gromov matrix with base \((T, s_1, V)\), where \(T\) is the subtree rooted at \(s_1\) that spans \(V\). The likelihood of observing the infection times \(T\) is then given by

\[
p(T | s_1, t_0, \mu, \sigma^2) = \frac{\exp\left(-\frac{1}{2\sigma^2}(T - D_{s_1}t_0)'A_{s_1}^{-1}(T - D_{s_1}t_0)\right)}{\sqrt{(2\pi\sigma^2)^n \det A_{s_1}}},
\]

where \(t_0 = [t_0, \mu]'\), and \(D_{s_1}\) is a matrix with its \(k\)-th row being \([1, d(s_1, v_k)]\). We have the following result.

**Proposition 2.** Suppose that \(G\) is a tree. Then, the MLE \((\hat{s}, \hat{t}_0, \hat{\mu}, \hat{\sigma}^2) = \arg\max \ p(T | s_1, t_0, \mu, \sigma^2)\) for \((s_1, t_0, \mu, \sigma^2)\) is given by

\[
\hat{s} = \arg\min_{s \in V \setminus V} \det(R_sA_s/n), \quad [\hat{t}_0, \hat{\mu}]' = \beta_s, \quad \hat{\sigma}^2 = R_s/n, \tag{3}
\]

where

\[
R_s = (T - D_s\beta_s)'A_s^{-1}(T - D_s\beta_s), \quad \beta_s = (D_s'A_s^{-1}D_s)'^{-1}D_s'A_s^{-1}T.
\]

**Proof:** For any estimator \(\hat{s}\) of \(s_1\), by maximizing (2) w.r.t. \(\beta_0 = [t_0, \mu]'\), the MLE of \(\beta_0\) is given by

\[
\hat{\beta}_s = \arg\min_{\beta \in \mathbb{R}^{2 \times 1}} (T - D_s\beta)'A_s^{-1}(T - D_s\beta) = (D_s'A_s^{-1}D_s)'^{-1}D_s'A_s^{-1}T.
\]

Similarly, the MLE of \(\sigma^2\) is given by

\[
\hat{\sigma}^2 = \arg\max_{\sigma^2 \in \mathbb{R}} \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{R_s}{2\sigma^2}\right) = R_s/n.
\]

Finally, the MLE of \(s_1\) is

\[
\hat{s} = \arg\max_{s \in V \setminus V} \ p(T | s, \beta_s, \sigma^2 = R_s/n) = \arg\min_{s \in V \setminus V} \det(R_sA_s/n),
\]

and the proof is complete.

The reference [27] derives the MLE of the source \(s_1\) by assuming that \((\mu, \sigma^2)\) are known, while Proposition 2 provides the MLE for both the source and diffusion parameters.

#### B. Single Source Estimation for General Graphs

In this subsection, we consider source estimation for general graphs. To find the MLE for the source, we need to optimize the likelihood in (2) over all possible infection trees rooted at each candidate source node. As discussed in Section II-A, this is intractable due to multiple paths between each pair of vertices. Therefore, we propose to perform the optimization over a parameterized family of Gromov matrices as follows. We call the following procedure GSSI, which is summarized in Algorithm 1.

(i) For each candidate source node \(s\), we first find two BFS trees in opposite search directions. For example, we index the nodes from 1 to \(|V|\). Then we construct one BFS tree by prioritizing nodes with smaller indices, and another BFS tree by prioritizing nodes with bigger indices. Let the two corresponding Gromov matrices be \(A_1^s\) and \(A_2^s\). For each \(\theta \in [0, 1]\), let

\[
M_s(\theta) = \theta A_1^s + (1 - \theta) A_2^s, \quad \beta_s(\theta) = (D_s'M_s^{-1}(\theta)D_s)'^{-1}D_s'M_s^{-1}(\theta)T,
\]

and

\[
R_s(\theta) = (T - D_s\beta_s(\theta))'M_s^{-1}(\theta)(T - D_s\beta_s(\theta)).
\]
We find
\[ \theta_s = \arg \min_{\theta \in [0,1]} \det (R_s(\theta)M_s(\theta)/n), \]
and let \( M_s \triangleq M_s(\theta_s) \).

(ii) We introduce a target covariance matrix
\[ H_s \triangleq m_s I_n \text{ or } \text{diag}(M_s), \]
where \( m_s = \text{tr}(M_s)/n \), \( I_n \) is the identity matrix with dimension \( n \), and \text{diag}(M_s) is the diagonal matrix with \[ \text{diag}(M_s)_{i,i} = [M_s]_{i,i}. \]
This yields \( \text{tr}(H_s) = \text{tr}(M_s) \).

For each \( \alpha \in [0,1] \), let
\[ A_s(\alpha) = \alpha H_s + (1 - \alpha) M_s, \]
\[ \beta_s(\alpha) = (D_s' A_s^{-1}(\alpha) D_s)^{-1} D_s' A_s^{-1}(\alpha) T, \]
and
\[ \hat{R}_s(\alpha) = \left( T - D_s \beta_s(\alpha) \right)' A_s^{-1}(\alpha) (T - D_s \beta_s(\alpha)). \]

We find
\[ \alpha_s = \arg \min_{\alpha \in [0,1]} \det (R_s(\alpha)A_s(\alpha)/n), \]
and let \( A_s \triangleq A_s(\alpha_s) \).

(iii) We define our estimator \( (\hat{s}, \hat{t}_0, \hat{\mu}, \hat{\sigma}^2) \) for \( (s_1, t_0, \mu, \sigma^2) \) as
\[ \hat{s} = \arg \min_{s \in V \setminus V'} \det (R_s(\alpha_s) A_s/n), \]
\[ [\hat{t}_0, \hat{\mu}]' = \beta_s(\alpha_s), \]
\[ \hat{\sigma}^2 = \hat{R}_s(\alpha_s)/n. \]

Steps (i) and (ii) above essentially constructs a Gromov matrix to approximate the infection diffusion path starting at a node \( s \), based on the observed infection timestamps \( T \). In step (ii), we let the new Gromov matrix \( A_s \) be a convex combination of \( M_s \) and a predefined invertible target matrix \( H_s \) as shown in (6). This is similar to the Stein-type shrinkage covariance matrix estimator proposed in [37], [38]. The difference is that we obtain the parameter \( \alpha_s \) by solving the optimization problem in (8). In this heuristic, the Gromov matrix \( A_s \) obtained in step (ii) may not correspond to an actual subtree of \( G \). To demonstrate that step (ii) improves our estimation performance, we present simulations in Section III-D where we compare the performance of GSSI with another procedure that we call naïve-GSSI, and which omits step (ii) and replaces \( A_s \) in step (iii) with \( M_s \).

We now give an interpretation of step (ii). When the graph is dense and the variance of the infection propagation delay is large, the spreading paths from the source to the observed infected nodes can be approximately regarded as uncorrelated (i.e., the spanning tree is a star tree). Therefore \( \alpha \) in (6) indicates how far the spanning tree is from a star tree. Intuitively, if the graph is denser, then \( \alpha_s \) is greater, which is validated by the simulation results in Fig. 6 of Section III-D.

In step (iii), the proposed estimator is simply the MLE assuming that \( A_s \) is the Gromov matrix of the underlying infection tree.

For each node in \( V \setminus V' \) we need to find two of their BFS trees, which can be computed with time complexity \( O(|V|^2). \)

The optimization problems in (4) and (8) do not have analytical solutions. We utilize standard iterative techniques like gradient descent to solve them, which require a time complexity of \( O(|V|^3) \), assuming that the maximum number of iterations used is fixed. The overall time complexity of GSSI is then \( O(|V|(|V|^2+|V|^3)) = O(|V|^3+|V|^2) \). To compare with other algorithms, the time complexity of TRBS proposed in [31] is \( O(|V||V|^2) \); and the time complexity of the method proposed in [27] is \( O(|V|^3+|V||V|^3) \) (note that \( |V| \) matrix inverses are required in [27], each incurring a time complexity of \( O(|V|^3) \)), assuming that the diffusion parameters are known a priori.

**Algorithm 1 Gromov Single Source Identification (GSSI)**

**Input:** Adjacency matrix of the graph \( G, V = \{v_1, ..., v_n\} \) and \( T = [t_1, ..., t_n] \).

**Output:** \( (\hat{s}, \hat{t}_0, \hat{\mu}, \hat{\sigma}^2) \).

1: Let \( f_m = \infty \).
2: for every \( s \in V \setminus V' \) do
3: Find two BFS trees rooted at \( s \) in opposite search directions to obtain \( A_s^1 \) and \( A_s^2 \).
4: Find \( \theta_s \) according to (4) to obtain \( M_s \).
5: Find \( \alpha_s \) according to (8) to obtain \( A_s \).
6: Compute \( f_s = \det (R_s(\alpha_s) A_s/n) \).
7: if \( f_s \leq f_m \) then
8: Set \( f_m = f_s \), and \( \hat{s} = s \).
9: end if
10: end for
11: Set \( (\hat{t}_0, \hat{\mu}, \hat{\sigma}^2) \) according to (9).

**C. Discussions**

In the GSSI algorithm, if \( G \) is a tree then we do not need to optimize \( \theta \) because \( A_s^1 = A_s^2 \), but in this case, GSSI is not equivalent to MLE due to the introduction of the parameter \( \alpha_s \) and target matrix \( H_s \). However, simulations in Section III-D indicate that the introduction of \( \alpha_s \) and \( H_s \) does not unreasonably impair our estimation results in trees, while the introduction of the additional step in (8) leads to better performance for general graphs.

Simulations in Section III-D suggest that when \( G \) is a tree, then with high probability the MLE in Proposition 2 and GSSI give the same source estimates, and \( \alpha_s \) obtained by GSSI is close to 0, which implies that the performance of GSSI is close to the MLE when the graph is a tree. We provide some theoretical results in Appendix A that indicate why this is true.

**D. Simulations for Single Source Estimation**

In this subsection, we present simulation results on both synthetic and real networks to compare the performance of GSSI with the MLE in Proposition 2, the TRBS algorithm proposed in [31], and the method proposed in [27], which we call the GAU algorithm. We note that both the TRBS and

\[ \text{We say that } f(N) = O(g(N)) \text{ if } f(N)/g(N) \leq k \text{ for some fixed } k > 0 \text{ as } N \rightarrow \infty. \]
GAU algorithms require a priori knowledge of the mean and/or variance of the propagation delay along each graph edge, while our approach estimates these parameters from the observed timestamps.

We first perform simulations on the Erdős-Rényi (E-R) trees ER(N) and the Barabási-Albert (B-A) trees BA(N), where N = 500 is the number of nodes. B-A graphs are scale-free while E-R graphs are not. Some properties like the diameter and average pairwise distance are listed in Table I. Our simulations show that choosing \( H_s = m_s I_s \) or \( \text{diag}(M_s) \) in (5) does not lead to much difference in the performance of GSSI. Therefore, here we only present the results for \( H_s = m_s I_s \).

### TABLE I

| Graph   | |E|/|V| | Diameter | Average pairwise distance |
|---------|-------|-----------------|--------|---------------------------|
| ER(500) | 2.00  | 20               | 9.23   |
| BA(500) | 1.00  | 15               | 6.40   |
| ER(500, 4) | 1.98  | 11               | 4.69   |
| BA(500, 4) | 1.99  | 7                | 3.82   |
| ER(500, 16) | 8.12  | 4                | 2.54   |
| BA(500, 16) | 7.88  | 4                | 2.46   |
| Enron   | 9.86  | 9                | 3.32   |
| Facebook| 25.9  | 9                | 2.95   |
| Twitter | 5.21  | 11               | 3.41   |

For each simulation, we randomly pick a node from the network to be the source. We simulate the propagation delays along each edge using a truncated Gaussian distribution \( N(2, 1) \), and randomly choose a subset of nodes as observed nodes. We perform 300 simulation runs. The error distance is defined as the distance between the estimated source and the real source. All four algorithms rank all the nodes in the graph according to their likelihood of being the source. Following [28], we define \( \gamma \)-accuracy as the proportion of simulations in which the real sources are ranked in the top \( \gamma \) percent of all the nodes. Simulation results are shown in Fig. 1. We only consider the case where the fraction of timestamps is less than 50%. Simulations indicate that with high probability (greater than 0.95) that GSSI and the MLE give the same source estimate.

The plots in Fig. 1 show that the performance of GSSI is very close to the MLE when \( G \) is a tree, which means that the introduction of parameter \( \alpha_s \) and target matrix \( H_s \) in (6) does not unreasonably impair our estimation results when \( G \) is a tree. We observe that GSSI performs no worse than GAU in almost all cases, and better than TRBS for both B-A and E-R trees although GSSI requires less information. The reason why GAU performs worse than GSSI in most cases is because the simulated propagation delays are generated using a truncated Gaussian distribution instead of the Gaussian distribution assumed by GAU. Although GSSI also assumes a Gaussian distribution, it estimates the distribution parameters from the observed data, which help to mitigate this mismatch. TRBS has the worst performance because it does not utilize any variance information. For GSSI, we compute the mean squared error (MSE) of the estimates \( \hat{\mu} \) and \( \hat{\sigma}^2 \) in Fig. 2, which show that the MSE of GSSI is very close to that of the MLE.

For each node \( s \in V \backslash V \), we find a BFS tree rooted at \( s \) and then perform MLE to compute \( g_s \). Let \( \hat{\sigma}^2 = \text{det}(R_s A_s /n) \) from Proposition 2. Finally we minimize \( g_s \) over \( s \in V \backslash V \) to obtain the source estimate. For the naive-GSSI, we follow the same procedure as GSSI, but only optimize \( \theta_s \) in (4), and let \( \alpha_s = 0 \). Simulation results are shown in Figs. 3 to 5. The

---

1https://snap.stanford.edu/data/email-Enron.html
2https://snap.stanford.edu/data/egonets-Facebook.html
plots show that in almost all the cases, GSSI performs best and achieves significant improvement over BFS-MLE, while naive-GSSI also performs better than BFS-MLE. Therefore, we see that the introduction of $\theta_s, \alpha_s$ and $H_s$ in GSSI can improve estimation performance.

![Fig. 3. Single source estimation on ER(500, 4) and BA(500, 4). For the diffusion process, set $(\mu, \sigma^2) = (2, 1)$.](image)

![Fig. 4. Single source estimation on ER(500, 16) and BA(500, 16), with $(\mu, \sigma^2) = (3, 1)$.](image)

Fig. 6. Simulations to study how $\alpha_s$ varies with the fraction of observed timestamps and for different graphs.

IV. MULTIPLE SOURCES ESTIMATION

In this section, we discuss the case of multiple infection sources, i.e., $|S| > 1$ but unknown. We first reduce the multiple sources estimation problem to the single source estimation problem, which is then solved by applying GSSI. We develop a reduction step for trees, and then extend it heuristically to general graphs.

A. The Reduction Step for Trees

In this subsection, we consider the case where $G$ is a tree, and introduce the concepts of observation cluster and source candidate cluster, which allow us to reduce the multiple sources estimation problem to that of estimating a single source within each cluster. Since all $\{\tau_{uv}\}$ are positive continuous random variables, the observed timestamps are distinct with probability one. The following definitions are illustrated in Fig. 7.

**Definition 1.** A node $v_k \in V$ is called observable w.r.t. $u \in V$ if the following condition holds: for each pair of distinct nodes $v_i, v_j \in V \cap [u, v_k]$ such that $v_i \in [u, v_j)$, then $t_i < t_j$. The collection of observable nodes w.r.t. $u \in V$ is called its observation cluster, and denoted by $\mathcal{V}_u$. We also study how $\alpha_s$ in (8) varies with increasing fraction of observed timestamps. From Fig. 6, we see that for B-A and E-R trees, the average $\alpha_s$ is very close to 0, which agrees with the intuition provided by Lemma A.1 in Appendix A. For B-A and E-R graphs, the average $\alpha_s$ increases as the average degree of the graph gets larger, which is intuitively satisfying because the actual infection propagation path can deviate significantly from BFS trees as the graphs become denser.
Definition 2. Let \( u \in V \). The collection of all nodes \( v \in V \) such that \( \mathcal{V}_v = \mathcal{V}_u \) is denoted by \( \mathcal{A}_u \), which is called the source candidate cluster w.r.t. \( \mathcal{V}_u \).

![Diagram](image_url)

Fig. 7. An example to illustrate the notions of observation cluster and source candidate cluster. Suppose that we observe 5 infected nodes \( \{1, 2, 3, 5, 9\} \) and their respective timestamps, labeled as \( \{(v_i, t_i)\}_{i=1}^{5} \) in the figure. For the node 4, since \( t_5 < t_4 \) then \( \mathcal{V}_4 = \emptyset \). For the node 3, we have \( t_3 < t_5 \) and \( t_4 < t_5 \) therefore \( \mathcal{V}_3 = \{1, 3, 5\} \). It is easy to check that \( \mathcal{V}_s = \emptyset \) for some node \( s \in \{1, 2, 3, 4, 6, 8, 9\} \) and for any node \( u \notin \{8, 9\} \), we have \( \mathcal{V}_u \neq \emptyset \). Therefore, \( \mathcal{A}_9 = \{8, 9\} \).

The observation cluster \( \mathcal{V}_u \) of \( u \) is the set of observed infected nodes in \( V \) that can possibly be infected by \( u \). However, \( \mathcal{V}_u \) may also be the observation cluster of other nodes \( v \neq u \). We collect all these potential source nodes into the source candidate cluster \( \mathcal{A}_u \). From Definitions 1 and 2, we have the following immediate observations. For a given \( v \in V \), since \( v_i \in \mathcal{V}_v \), the set \( \mathcal{V}_v \) is non-empty. Similarly, \( v_i \in \mathcal{A}_v \). Let \( \mathcal{I}_sm \) be the collection of nodes in \( V \) infected by \( s_m \in S \). Specifically, if there is only one source \( s \), then \( \mathcal{V}_s = \mathcal{I}_s = V \).

For any \( u \in V \), let
\[
\nu_u = \arg \min_{v \in \mathcal{V}_u} t_k. 
\] (10)

We have the following results regarding \( \mathcal{V}_u \), \( \mathcal{A}_u \), and \( \mathcal{I}_sm \).

Lemma 2. Consider any node \( u \in V \).

(i) If \( v_k \in \mathcal{V}_u \), then for any \( v_i \in [u, v_k) \cap \mathcal{V} \), we have \( v_i \in \mathcal{V}_u \) and \( t_i < t_k \).

(ii) \( \mathcal{A}_u \cap \mathcal{V} = \{v_u\} \). Specifically, if \( u = v_s \) for some \( s \in S \), then \( \mathcal{V}_u = \mathcal{V}_s \) (or \( s_m \in \mathcal{A}_u \)).

(iii) The source candidate cluster is connected.

(iv) For each \( s_m \in S \), \( \mathcal{I}_s \subseteq \mathcal{V}_s \).

(v) If \( v_k \in \mathcal{V}_u \), then \( \mathcal{V}_v \subset \mathcal{V}_u \). Specifically, if \( v_k \in \mathcal{I}_sm \), then \( \mathcal{V}_v \subset \mathcal{V}_s \).

(vi) If \( v_u \in \mathcal{V}_u \), then \( \mathcal{V}_u \subset \mathcal{V}_v \).

Proof:

(i) This follows directly from Definition 1.

(ii) We first prove that for any \( v_k \in \mathcal{V}\setminus\mathcal{V}_u \), then \( \mathcal{V}_u \neq \mathcal{V}_v \). If \( v_k \notin \mathcal{V}_u \), then it follows directly that \( \mathcal{V}_u \neq \mathcal{V}_v \). If \( v_k \in \mathcal{V}_u \), then \( v_k \notin \mathcal{V}_v \) because \( t_k > t' \), where \( t' \) is the timestamp of node \( v' \). Since \( v_k \in \mathcal{V}_v \), we have \( \mathcal{V}_u \neq \mathcal{V}_v \). We then prove that \( \mathcal{V}_u = \mathcal{V}_v \). If \( u \in \mathcal{V}_u \), then \( u = v_u \) and it follows directly that \( \mathcal{V}_u = \mathcal{V}_v \). We next consider the case where \( u \notin \mathcal{V}_u \). Suppose \( v_k \in \mathcal{V}_u \setminus \mathcal{V}_v \). According to claim (i), we find \( v_i \in \mathcal{V}_u \) such that \( v_i \in [u, v_k) \) and \( [u, v_i) \cap \mathcal{V} = v_i \). By the minimality condition (10), \( [u, v_i) \cap \mathcal{V} = \{v_u\} \). Then we have \( [u, v_k) \cap \mathcal{V} = \{v_u, v_i\} \). Since \( u \notin \mathcal{V}_u \), \( v_i \notin \mathcal{V}_u \). Therefore, \( v_k \notin \mathcal{V}_u \) from Definition 1. Suppose on the contrary \( v_k \notin \mathcal{V}_u \). Then from Definition 1, there exists a pair of distinct nodes \( v_i, v_j \in \mathcal{V} \setminus [u, v_k) \) such that \( v_i \in [u, v_j) \). As a consequence, \( t_i > t_j \). Since \( u \notin \mathcal{V}_u \cap \mathcal{V} = \{v_u\} \), we have \( v_i \notin \mathcal{V}_u \) and \( v_j \notin \mathcal{V}_u \). Therefore \( \mathcal{V}_u = \mathcal{V}_v \) from Definition 1. According to claim (ii), it is equivalent to prove \( \mathcal{A}_u \) is connected. We now prove that for any \( v_k \in \mathcal{V} \), \( \mathcal{A}_u \) is connected. From claim (iii) we have \( \mathcal{A}_u \cap \mathcal{V} = \{v_k\} \). Therefore given \( u \in \mathcal{A}_u \setminus \mathcal{V}_u \), we have \( u \notin \mathcal{V} \). We show that \( (v_k, v) \cap \mathcal{V} = \emptyset \). Suppose on the contrary that there exists \( v_i \in \mathcal{V} \) and \( v_i \in (v_k, v) \). According to claim (i), since \( v_k \in \mathcal{V}_u \) then \( v_i \in \mathcal{V}_s \) and \( t_k < t_i \). Then we must have \( v_i \notin \mathcal{V}_u \) which contradicts \( \mathcal{V}_u = \mathcal{V}_v \). Therefore for any \( v \in (v_k, u) \), we have \( u \notin \mathcal{V}_u \cap \mathcal{V} = \emptyset \), then it is easy to see that \( \mathcal{V}_u = \mathcal{V}_v \). Thus \( u \in \mathcal{A}_v \). Therefore the claim of connectedness holds.

(iv) If \( v_k \in \mathcal{I}_sm \), then we have \( \mathcal{V}_u = \mathcal{V}_s \). As the propagation delay over each edge is positive, it is easy to verify that \( \mathcal{V}_u \subset \mathcal{V}_s \) from Definition 1.

(v) We prove that if \( v_i \in \mathcal{V}_u \) and \( v_j \in \mathcal{V}_v \), then \( v_j \in \mathcal{V}_u \). The proof follows directly from Definition 1.

We perform BFS, with the search path stopping immediately before hitting any node in \( \mathcal{V} \setminus \{v_i\} \). Let \( B(v_i, V) \) be the collection of discovered nodes in this tree.

Proposition 3. For any \( v_i \in \mathcal{V} \), \( A_{v_i} \subset B(v_i, V) \). If \( v_i \notin \mathcal{V}_v \) for any \( v_j \in \mathcal{V} \setminus \{v_i\} \), then \( A_{v_i} = B(v_i, V) \).

Proof: According to the proof of Lemma 2(iii), it is easy to see that if \( u \in \mathcal{A}_v \), then \( u \in B(v_i, V) \). Since \( u \notin \mathcal{V}_v \), then we have \( u \notin B(v_i, V) \). Therefore \( \mathcal{V}_u = \mathcal{V}_v \). We now introduce the notion of an admissible covering of \( V \).

Definition 3. A covering that decomposes \( V = \bigcup_{i=1}^{l} \mathcal{U}_i \) into a union (not necessarily disjoint) of non-empty subsets \( \mathcal{U}_i \), for \( 1 \leq i \leq l \), is called admissible if for each \( i \), and there exists \( u_i \) such that \( \mathcal{V}_{u_i} = \mathcal{U}_i \).
It is easy to see that \( \{ \{ V_{m} \}_{m=1}^{S} \} \) is an admissible covering of \( V \). However, since the sources are unknown, we propose the Multiple Sources Reduction (MSR) algorithm in Algorithm 2 to find an admissible covering. We have the following result.

**Algorithm 2 Multiple Sources Reduction (MSR)**

**Input:** Adjacency matrix of the tree \( G \), observations \( V = \{ v_1, \ldots, v_s \} \) and \( T = [t_1, \ldots, t_n] \).

**Output:** \( \{ V_{\xi_i} \}_{i=1}^{l} \) and \( \{ A_{\xi_i} \}_{i=1}^{l} \).

1. Let \( U = V \), and \( i = 1 \).
2. while \( U \neq \emptyset \) do
   3. \( \xi_i = \arg \min_{v_j \in U} t_k \).
   4. Find \( V_{\xi_i} \) according to Definition 1, then find \( A_{\xi_i} = B(\xi_i, V) \) (cf. Proposition 4(i)).
   5. Set \( l = i, i = i + 1, U = U \setminus V_{\xi_i} \).
6. end while

**Proposition 4.** Consider the the MSR algorithm in Algorithm 2.

(i) For each \( i = 1, \ldots, l \), \( A_{\xi_i} = B(\xi_i, V) \).

(ii) \( \{ V_{\xi_i} \}_{i=1}^{l} \) is an admissible covering of \( V \).

(iii) Each \( A_{\xi_i} \) contains at least one source, so \( l \leq |S| \). If \( l = |S| \), then each \( A_{\xi_i} \) contains exactly one source.

(iv) \( l = |S| \) if and only if for each distinct pair \( s_i, s_j \in S \), \( v_{s_i} \notin V_{s_j} \).

**Proof:**

(i) Since \( \xi_i = \arg \min_{v_j \in U} t_k \), for any \( v_j \in U \setminus \{ \xi_i \} \), we have \( \xi_i \notin V_{s_j} \). Suppose there exists \( v_j \in V \setminus U \) such that \( \xi_i \in V_{s_j} \). Since \( v_j \in V_{s_k} \) for some \( k < i \), from Lemma 2(ii), \( V_{s_k} \subset V_{\xi_i} \) and we obtain a contradiction with \( \xi_i \notin U \). The claim then follows from Proposition 3.

(ii) It follows that each \( V_{\xi_i} \) is non-empty and \( V = \bigcup_{i=1}^{l} V_{\xi_i} \) with \( l \leq |V| \).

(iii) If \( \xi_i \in V_{s_1} \), then \( \xi_i = v_{s_1} \). According to Lemma 2(ii), \( s_1 \in A_{\xi_i} \). If \( U \neq \emptyset \) and suppose \( \xi_i \notin V_{s_1} \), then from Lemma 2(vi), we must have \( v_{s_1} \notin \bigcup_{j=1}^{l} V_{\xi_j} \). Therefore \( v_{s_1} \notin V_{s_j} \) and \( s_1 \in A_{\xi_i} \). This procedure is then repeated and each \( A_{\xi_i} \) contains at least one source. Moreover, it is likely that some sources are not contained in any \( A_{\xi_i} \). Refer to the example in Fig. 8. So \( l \leq |S| \). If \( l = |S| \), then each \( A_{\xi_i} \) contains exactly one source.

(iv) By the proof of claim (iii), if for each distinct pair \( s_i, s_j \in S \) such that \( v_{s_i} \notin V_{s_j} \), then we can find \( \{ C_m \}_{m=1}^{S} \) with \( C_m = v_{s_m} \). On the contrary, if there exists \( v_{s_i} \notin V_{s_j} \) then \( v_{s_i} < s_i \), so we must choose \( v_{s_j} \) before \( v_{s_i} \). From Lemma 2(vi), \( V_{s_j} \subset V_{s_i} \) thus \( V_{s_j} \cap U \setminus V_{s_i} = \emptyset \). Hence \( s_i \in A_{\xi_i} \) (cf. Fig. 8(a)) or \( s_i \) will not be contained at any \( \{ A_{\xi_j} \}_{j=1}^{l} \) (cf. Fig. 8(b)). This implies that \( l < |S| \).

Finding \( \xi_i \) in Algorithm 2 can be accomplished by maintaining a min-heap for \( T \), which incurs an overall time complexity of \( O(|V|) \) for the whole algorithm. Finding \( A_{\xi_i} \) involves BFS, which incurs a time complexity of \( O(|V|) \). Therefore, the overall time complexity of MSR is \( O(|V|) \).

Proposition 4(iii) indicates that if we assume that there is only a single source in each source candidate cluster \( A_{\xi_i} \), then we are underestimating the number of sources (see Fig. 8 for an example). Therefore we adopt an additional procedure developed in [39] to further partition each \( A_{\xi_i} \) and \( V_{\xi_i} \). The paper [39] has proposed an algorithm, which we call the Source-Set Size Estimation (SSSE) algorithm, to estimate the number of infection sources in a tree by partitioning the tree into several covering components. We briefly summarize the SSSE algorithm in Algorithm 3.

**Algorithm 3 Source-Set Size Estimation (SSSE)**

**Input:** Adjacency matrix of the tree \( G \).

**Output:** A set of disjoint and connected subtrees of \( G \), denoted by \( \{ T_i \}_{i=1}^{K} \).

1. Given \( G \), compute the average pairwise distance \( \bar{d} \), and then find a path \( P = [u, v] \) with the greatest length \( L_G \).
2. if \( L_G > \bar{d} \) then
   3. Define a function \( s_p(\cdot) \) on \( P \). For each \( w \in P \), find the leaves (with degree 1) in \( G \) that are closest to \( w \) compared with nodes in \( P \). Then \( s_p(w) \) is assigned to the sum of distances between \( w \) and each of these leaves. If \( s_p(w) \) is a local minimum and \( w' \) is a neighbor of \( w \) in \( P \), then delete the edge connecting \( w \) and \( w' \). \( G \) is divided into two subtrees \( T_1 \) and \( T_2 \).
4. end if
5. For \( T_1 \) and \( T_2 \), repeat the above partition procedure until the diameters of the subtrees are less than \( \bar{d} \) or \( s_p(\cdot) \) does not have a local minimum.

The time complexity of SSSE is \( O(|V|) \). From Lemma 2(ii) we see that each \( A_{\xi_i} \) forms a connected subtree. So we apply the SSSE algorithm on each \( A_{\xi_i} \) to obtain a new covering for \( V \), which are denoted by \( \{ C_i \}_{i=1}^{L} \), where \( L \geq l \). At the same time \( \{ V_{\xi_i} \}_{i=1}^{l} \) is also partitioned as \( \{ O_i \}_{i=1}^{L} \). Our simulations indicate that it is now possible for \( L > |S| \) as SSSE may over-estimate the number of sources in each \( A_{\xi_i} \).

Finally, for each \( 1 \leq i \leq L \), we apply GSSI to find a source estimate \( \hat{s}_i \in C_i \) using \( O_i \) as an approximation for \( I_{\hat{s}_i} \). Note that different sources may have different diffusion parameters \((\hat{t}_0, \mu, \sigma^2) \) and GSSI is able to estimate these parameters for each possible source.

**B. Multiple Sources Estimation for General Graphs**

We now heuristically extend the sources estimation procedure for trees described in the previous subsection to general graphs, and summarized it in Algorithm 4, where \( \{ \hat{s}_i \}_{i=1}^{L} \) is the set of estimated sources. First, we make use of the BFS heuristic in generalizing the MSR algorithm to a general
Algorithm 4 Source Candidate Clustering and Estimation (SCCE)

Input: Adjacency matrix of the graph $G$, observation $V = \{v_1, \ldots, v_n\}$ and $T = \{t_1, \ldots, t_n\}$.

Output: $\hat{S} = \{\hat{s}_i\}_{i=1}^L$.

1: Let $U = V$, and $i = 1$ 
2: while $U \neq \emptyset$ do 
3: \hspace{0.5cm} $\xi_i = \arg\min_{v_k \in U} \Delta L_k$.
4: \hspace{0.5cm} Find one BFS tree rooted at $\xi_i$ and then obtain $B(\xi_i, V)$ and $V_{\xi_i}$ on this tree. Set $A_{\xi_i} = B(\xi_i, V)$.
5: \hspace{0.5cm} Set $l = i$, $i = i + 1$, $U = U \setminus V_{\xi_i}$.
6: \hspace{0.5cm} end while 
7: For each $1 \leq i \leq l$, run SSSE algorithm to partition $V_{\xi_i}$ and $A_{\xi_i}$ on the BFS tree rooted at $\xi_i$. Finally we obtain $\{C_i\}_{i=1}^L$ and $\{O_i\}_{i=1}^L$.
8: for every $C_i$ and $O_i$ do 
9: \hspace{0.5cm} Run GSSI algorithm. Compute $f_s$ in line 6 of Algorithm 1 for each $s \in C_i \setminus \{\xi_i\}$.
10: \hspace{0.5cm} Pick $s_i$ to be a node $s$ that minimizes $f_s$.
11: end for

C. Experimental Evaluations for Multiple Sources Estimation

We evaluate the performance of our SCCE algorithm with the same graphs used in Section III-D. We compare SCCE with the method proposed in [34], which we call the BDSL algorithm. To quantify the performance of SCCE and BDSL, we adopt the performance metric (with a slight modification for fairer comparison) proposed in [13]: We first match the estimated sources $\{\hat{s}_i\}_{i=1}^L$ with the actual sources $\{s_i\}_{i=1}^{|S|}$ so that the sum of the error distances between each estimated source and its match is minimized. Denote this matching as $\pi$. If we incorrectly estimate the number of infection sources, i.e., $L \neq |S|$, we add a penalty $\eta|S| - L$. Denote $\bar{d}$ as the average pairwise distance of the graph. We define the average error to be

$$\Delta = \frac{1}{L} \left( \sum_{i=1}^{\min(L, |S|)} d(s_i, \hat{s}_{\pi(i)}) + \eta|S| - L \right),$$

(11)

where

$$L = \begin{cases} \min(L, |S|) & \text{if } \eta = 0, \\ \max(L, |S|) & \text{if } \eta = \bar{d}. \end{cases}$$

For different applications, we may assign different values to $\eta$ depending on how important it is to estimate correctly the number of infection sources. In (11) we divide the sum of distances by $L$ to obtain the average. The BDSL algorithm cannot estimate the number of sources, so we assume it has prior knowledge of $|S|$. Therefore for BDSL, we have $L = |S|$.

We first perform simulations on ER(500) and BA(500). For each simulation, we let the number of sources be uniformly chosen from $\{2, 3, 4\}$. From Proposition 4(iv), we see that if two sources are close to each other, then it is likely that one of them cannot be identified. So we choose sources such that the distance between each pair of sources is no less than the average pairwise distance of the tree, which is a reasonable assumption in real applications. For the diffusion process, let the start times of all sources be identical, which is an assumption required by BDSL, and propagation delays along all edges follow a truncated Gaussian distribution $\mathcal{N}(2, 1)$. Notice that BDSL requires to know the mean of the Gaussian distribution used while our algorithm does not. For the BDSL algorithm, since we uniformly choose $|S|$ from $\{2, 3, 4\}$, the average $L$ is 3, which is used as a benchmark to compare how well SCCE performs in estimating the number of sources. We randomly choose a portion of nodes as observed nodes for the multiple sources estimation. We average $\Delta$ in (11), and $L$ over 300 simulations and plot against the fraction of timestamps as shown in Fig. 9. The plots in Figs. 9(a) and 9(b) show that SCCE performs better than BDSL for both B-A and E-R trees although SCCE requires less information. For SCCE algorithm, from Fig. 9(c) we see that $L > |S|$ in some cases due to the introduction of the additional partition procedure developed in the SSSE algorithm; and the average $L$ is close to the average $|S|$ if we observe enough timestamps.

We next perform simulations on some synthetic graphs and
real networks (cf. Table I for some properties of these graphs). Let the number of sources be uniformly chosen from \(\{2, 3\} \), so that its average value is 2.5. For the truncated Gaussian distribution used to simulate the infection spreading, we set \((\mu, \sigma^2) = (3, 1)\). Since in real applications, \(|S|\) is typically not very large, we make the additional assumption that \(|S| \leq 5\) when running the SCCE algorithm by controlling the number of iterations in the MSR step. Simulation results in Figs. 10 to 13 indicate that in terms of average \(\Delta\), SCCE performs best in the case where \(\eta = 0\). In most cases SCCE also performs better than BDSL even when \(\eta = 7\). For the performance of average \(L\), we see that generally SCCE finds more source estimates as the fraction of timestamps increases. On E-R and B-A graphs, SCCE is able to find estimates that are on average within 2 hops of the real sources when the fraction of timestamps is at least 15%. When the graph is denser (i.e., \(Np\) is greater), SCCE tends to obtain smaller values of \(L\). In particular, we over-estimate the number of sources on BA(500, 4) and ER(500, 4). For the Enron and Facebook networks, BDSL has poor estimation accuracy, while SCCE produce estimates that are on average within 2.5 hops of the real sources with no less than 30\% of nodes being observed.

![Fig. 10](https://example.com/fig10.png)

**Fig. 10.** Multiple sources estimation on (a) BA(500, 4) and (b) ER(500, 4).

![Fig. 11](https://example.com/fig11.png)

**Fig. 11.** Multiple sources estimation on (a) BA(500, 16) and (b) ER(500, 16).

Finally, we evaluate the performance of SCCE with a sample of real data provided by SNAP. First developed in the work [40], the Higgs dataset has been built after monitoring the spreading processes on Twitter before, during and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson on 4 July 2012. Before the announcement on 1 July 2012, there were some rumors about the discovery of a Higgs-like boson at Tevatron. Such rumors were posted on the Twitter social network and quickly spread among Twitter users around the world. The dataset provides a social network with 456,626 users and 14,855,842 connections. In addition, a retweet network including the users and timestamps of retweet events is also given. We extract a connected sub-network with 5,078 users and 26,449 connections. All the users in this sub-network posted or retweeted the rumors related to the Higgs boson discovery. We also collect 4,254 users’ first retweet timestamps and the total spanned time is about 39 hours starting from 1 July 2012. We then model the sub-network as an undirected graph (cf. Table I), regard the timestamps as our observations, and the first several users who posted the rumors independently as infection sources.

The diffusion of the retweets is heterogeneous and unknown to us, which means that our assumptions for the diffusion model are not satisfied. However, we still apply our SCCE algorithm to estimate the sources given the graph topology and timestamp observations. Algorithms such as GAU and BDSL require more information such as the mean of the propagation delays, which need to be learned from historical data. From the dataset we find three infection sources, let \(S = \{s_1, s_2, s_3\}\). In the extracted graph we have \(d(s_1, s_2) = 2\), \(d(s_2, s_3) = 2\), and \(d(s_1, s_3) = 3\). These three infection sources have different start times. We observe that \(s_1\) first initiated the diffusion process, then after about 7 and 15 hours, \(s_2\) and \(s_3\) started their diffusion, respectively. We then randomly choose a fraction of timestamps to perform multiple sources estimation using SCCE. After obtaining the estimated sources, we compute the average \(\Delta\) and average \(L\) to show its performance. To test the BDSL algorithm, we learn the mean time of a tweet spreading across an edge from 50,000 historical tweets, which yields an estimate of about 4 hours. We average over 50 experiments. The comparison between SCCE and BDSL is shown in Table II. We see that SCCE performs well on this real data. Comparatively, BDSL has poorer accuracy since the start times of the sources differ significantly from each other.

\(^{6}\)https://snap.stanford.edu/data/higgs-twitter.html
and the mean propagation delays are not accurately known. The average number of estimated sources $L$ obtained by SCCE is 2.2 with 10% timestamps and 3.1 with 20% timestamps.

| Fraction of timestamps | Average $\Delta$ |
|------------------------|------------------|
| SCCE ($\eta = 0$)     | 0.73             |
| SCCE ($\eta = d$)     | 1.88             |
| BDSL                  | 5.6              |

**TABLE II**

**COMPARISON BETWEEN SCCE AND BDSL ON REAL DATA.**

V. CONCLUSION

We have developed algorithms to estimate infection sources using a subset of timestamps for the cases where there is a single source, and where there are multiple sources and the number of sources is unknown. We adopt a Gaussian spreading model with unknown diffusion parameters, which are estimated as part of our inference algorithms. In the single source case, we introduce a new heuristic that optimizes over a parametrized family of Gromov matrices to more accurately identify the source in a general graph. In the multiple sources estimation problem, we develop theory that allows us to partition the observed infected nodes into observation clusters. We then apply our single source algorithm in each cluster. Experimental evaluations with synthetic and real-world data suggest that our approaches can find the infection sources to within a small number of hops from the true sources with a small number of timestamp observations.

APPENDIX A

HEURISTIC ANALYSIS OF GSSI

Suppose that the estimates by GSSI $(\hat{s}, [\hat{t}_0, \hat{\mu}], \hat{\sigma}^2) = (s_1, \beta_0, \sigma^2)$, the true parameter values. Then, $\alpha_s = \alpha_{s_1}$ in (8) is the solution of the following optimization problem:

$$
\alpha_{s_1} = \arg \max_{\alpha \in [0,1]} \alpha \exp \left(-\frac{1}{2\sigma^2} U' A_{s_1}^{-1}(\alpha) U \right) = \arg \max_{\alpha \in [0,1]} \exp \left(-\frac{1}{2\sigma^2} U' A_{s_1}^{-1}(\alpha) U \right)
$$

where $U = T - D_{s_1} \beta_0$ and $A_{s_1}(\alpha) = \alpha H_{s_1} + (1 - \alpha) A_{s_1}$ according to (6). Let $f_1(\alpha) = \log \det A_{s_1}(\alpha)$ and $f_2(\alpha) = U' A_{s_1}^{-1}(\alpha) U$ so that $f'(\alpha) = f_1(\alpha) + f_2(\alpha)/\sigma^2$. Our goal is to show that on average, $f(0)$ is close to a local optimum when $G$ is a tree, since $\alpha = 0$ makes GSSI equivalent to MLE.

**Lemma A.1.** Suppose that $G$ is a tree, and the GSSI estimates $(\hat{s}, [\hat{t}_0, \hat{\mu}], \hat{\sigma}^2) = (s_1, \beta_0, \sigma^2)$. Then, we have

(i) $f_1(\alpha)$ is concave and non-decreasing for $\alpha \in [0,1]$;
(ii) $f_2(\alpha)$ is convex w.r.t $\alpha$ and $\frac{d}{d\alpha} f_2(\alpha)|_{\alpha=0} \geq U' A_{s_1}^{-1}(\alpha) U (1 - \lambda_{\max}(A_{s_1}(\alpha) H_{s_1}))$, where $\lambda_{\max}(\cdot)$ is the largest eigenvalue of its matrix argument.
(iii) $\mathbb{E} \left[ \frac{d}{d\alpha} f(\alpha) \right]_{\alpha=0} = 0$.

**Proof:**

(i) The first and second derivatives of $f_1(\alpha)$ w.r.t. $\alpha$ are given by

$$
\frac{d}{d\alpha} f_1(\alpha) = \frac{1}{\det A_{s_1}(\alpha)} \frac{d}{d\alpha} \det A_{s_1}(\alpha)
$$

and

$$
\frac{d^2}{d\alpha^2} f_1(\alpha) = \text{tr}(A_{s_1}^{-1}(\alpha) \frac{d}{d\alpha} A_{s_1}(\alpha)) = \text{tr}(A_{s_1}^{-1}(\alpha)(H_{s_1} - A_{s_1})),
$$

(ii) We have

$$
\frac{d}{d\alpha} f_2(\alpha) = -U' A_{s_1}^{-1}(\alpha) \frac{d}{d\alpha} A_{s_1}(\alpha) A_{s_1}^{-1}(\alpha) U = -U' A_{s_1}^{-1}(\alpha)(H_{s_1} - A_{s_1}) A_{s_1}^{-1}(\alpha) U,
$$

and

$$
\frac{d^2}{d\alpha^2} f_2(\alpha) = 2U' A_{s_1}^{-1}(\alpha) \frac{d}{d\alpha} A_{s_1}(\alpha) A_{s_1}^{-1}(\alpha) (H_{s_1} - A_{s_1}) A_{s_1}^{-1}(\alpha) U = 2[U' A_{s_1}^{-1}(\alpha)(H_{s_1} - A_{s_1}) A_{s_1}^{-1}(\alpha) (H_{s_1} - A_{s_1}) A_{s_1}^{-1}(\alpha) U] \geq 0,
$$

where the last inequality holds because $A_{s_1}(\alpha)$ is positive semidefinite. Therefore, $f_2(\alpha)$ is convex.

To show the lower bound, we have

$$
\frac{d}{d\alpha} f_2(\alpha)|_{\alpha=0} = U' A_{s_1}^{-1} U - U' A_{s_1}^{-1} H_{s_1} A_{s_1}^{-1} U.
$$

Let $W = H_{s_1} A_{s_1}^{-2}$ and $u = A_{s_1}^{-2} U$. Let $\sigma^2(W)$ be the spectral norm of $W$, then we obtain from Theorem 5.2.7 of [41],

$$
\sigma^2(W) = \sup_{x \neq 0} \frac{x' W x}{x' x} \leq \frac{u' W u}{u' u} = \frac{U' A_{s_1}^{-1} H_{s_1} A_{s_1}^{-1} U}{U' A_{s_1}^{-1} U},
$$

and $U' A_{s_1}^{-1} H_{s_1} A_{s_1}^{-1} U \leq \sigma^2(W) U' A_{s_1}^{-1} U$. Since $\sigma^2(W) = \sigma_{\text{max}}(WW')$, we have

$$
\frac{d}{d\alpha} f_2(\alpha)|_{\alpha=0} \geq U' A_{s_1}^{-1} U (1 - \lambda_{\max}(A_{s_1} H_{s_1})).
$$
since $H_2^\alpha \Lambda s_1^{-1}H_2^\alpha$, and $\Lambda s_1^{-1}H_2$, have the same eigenvalues.

(iii) From (2), we have $T \sim N(\mathbf{D}_s, \beta_0, \sigma^2 \Lambda s_1)$ so $\frac{1}{\sigma^2} \Lambda s_1^{-\frac{1}{2}} U \sim N(0, \mathbf{I}_s)$, and $\frac{1}{\sigma^2} U^\alpha \Lambda s_1^{-\frac{1}{2}} U$ follows a chi-squared distribution with $n$ degrees of freedom. From (12) and (13), we obtain

$$
E \left[ \frac{d}{d\alpha} f(\alpha) \right]_{\alpha=0} = -\text{tr}(\Lambda s_1^{-1}(Hs_1 - A_s)),
$$

and the proof is complete.

Lemma A.1(iii) says that on average the first derivative of $f(\alpha)$ is zero at $\alpha = 0$, which heuristically implies that $\alpha s_1$ cannot be much larger than zero with high probability.

**References**

[1] R. Kumar, J. Novak, and A. Tomkins, “Structure and evolution of online social networks,” in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 2006, pp. 611–617.
[2] A. L. Lloyd and R. M. May, “How viruses spread among computers and people,” *Science*, vol. 292, no. 5520, pp. 1316–1317, 2001.
[3] P. Wang, M. C. González, C. A. Hidalgo, and A. L. Barabási, “Understanding the spreading patterns of mobile phone viruses,” *Science*, vol. 324, no. 5930, pp. 1071–1076, 2009.
[4] F. Neumann, T. Noda, and Y. Kawoaka, “Emergence and pandemic potential of swine-origin H1N1 influenza virus.” *Nature*, vol. 459, no. 7249, pp. 931–939, 2009.
[5] R. Pastor-Satorras and A. Vespignani, “Epidemic spreading in scale-free networks,” *Phys. Rev. Lett.*, vol. 86, no. 14, pp. 3200–3203, Apr 2001.
[6] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, “Information diffusion in online social networks: A survey,” *ACM SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, Jul. 2013.
[7] H. Berghel, “The code red worm,” *Commun. ACM*, vol. 44, no. 12, pp. 15–19, 2001.
[8] R. Richardson and C. Director, “Csi computer crime and security survey,” *Computer Security Institute*, vol. 1, pp. 1–30, 2008.
[9] D. Shah and T. Zaman, “Rumors in a network: Who’s the culprit?” *Proc. ACM SIGMETRICS Performance Evaluation Review*, vol. 42, 2014, pp. 199–210.
[10] W. Luo, W. P. Tay, and M. Leng, “Identifying infection sources in real-world social networks,” *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4228–4243, Aug 2016.
[11] K. Zhu, Z. Chen, and L. Ying, “Locating the contagion source in networks with partial timestamps,” *Data Mining and Knowledge Discovery*, pp. 1–32, 2015.
[12] Z. Shen, C. Cao, W. Wang, Z. Di, and H. E. Stanley, “Locating the source of diffusion in complex networks by time-reversal backward spreading,” *Phys. Rev. E.*, vol. 93, p. 032301, Mar 2016.
[13] S. Zejnilovíc, J. Gomes, and B. Sinopoli, “Network observability and localization of the source of diffusion based on a subset of nodes,” in *Proc. IEEE Annu. Allerton Conf. on Communication, Control, and Computing*, 2013, pp. 847–852.
[14] S. Zejnilovíc, J. Xavier, J. Gomes, and B. Sinopoli, “Selecting observers for source localization via error exponents,” in *Proc. IEEE Int. Symp. Information Theory*, 2015, pp. 2914–2918.
[15] L. Fu, Z. Shen, W. Wang, Y. Fan, and Z. Di, “Multi-source localization on complex networks with limited observers,” *Europhysics Letters*, vol. 113, no. 1, p. 18006, 2016.
[16] L. Zheng and C. W. Tan, “A probabilistic characterization of the rumor graph boundary in rumor source detection,” in *Proc. IEEE Int. Conf. Digital Signal Process.*, 2015, pp. 765–769.
[17] I. Kavouglou and J. Benakdi, “Boundaries of hyperbolic groups,” *arXiv preprint math/0202296*, 2002.
[18] O. Ledoit and M. Wolf, “A well-conditioned estimator for large-dimensional covariance matrices,” *J. Multivar. Anal.*, vol. 88, no. 2, pp. 365–411, Feb. 2004.
[19] A. Touloumis, “Nonparametric stein-type shrinkage covariance matrix estimators in high-dimensional settings,” *Computational Stat. & Data Anal.*, vol. 83, pp. 251–261, 2015.
[20] F. Ji, W. P. Tay, and L. R. Varshney, “Estimating the number of infection sources in a tree,” in *Proc. IEEE Global Conf. Signal and Inf. Process.*, Dec 2016, pp. 380–384.
[21] F. Ji, W. P. Tay, and L. R. Varshney, “Nonparametric stein-type shrinkage covariance matrix estimators in high-dimensional settings,” *Computational Stat. & Data Anal.*, vol. 83, pp. 251–261, 2015.
[22] F. Ji, W. P. Tay, and L. R. Varshney, “An algorithmic framework for estimating rumor sources with different start times,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2517–2530, May 2017.
[23] W. Luo, W. P. Tay, and M. Leng, “How to identify an infection source with limited observations,” *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 4, pp. 586–597, 2014.