Network Topology Inference Using Information Cascades with Limited Statistical Knowledge

Feng Ji, Wenchang Tang, Wee Peng Tay, Senior Member, IEEE, and
Edwin K. P. Chong Fellow, IEEE

Abstract

We study the problem of inferring network topology from information cascades, in which the amount of time taken for information to diffuse across an edge in the network follows an unknown distribution. Unlike previous studies, which assume knowledge of these distributions, we only require that diffusion along different edges in the network be independent together with limited moment information (e.g., the means). We introduce the concept of a separating vertex set for a graph, which is a set of vertices in which for any two given distinct vertices of the graph, there exists a vertex whose distance to them are different. We show that a necessary condition for reconstructing a tree perfectly using distance information between pairs of vertices is given by the size of an observed separating vertex set. We then propose an algorithm to recover the tree structure using infection times, whose differences have means corresponding to the distance between two vertices. To improve the accuracy of our algorithm, we propose the concept of redundant vertices, which allows us to perform averaging to better estimate the distance between two vertices. Though the theory is developed mainly for tree networks, we demonstrate how the algorithm can be extended heuristically to general graphs. Simulation results suggest that our proposed algorithm performs better than some current state-of-the-art network reconstruction methods.

Index Terms

Network topology inference, information cascades, information diffusion, graph theory

The first three authors are supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2014-T2-1-028, and the last author is supported in part by NSF grant CCF-1422658.

F. Ji, W. Tang and W. P. Tay are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: jifeng@ntu.edu.sg, E150012@e.ntu.edu.sg, wptay@ntu.edu.sg). Edwin K. P. Chong is with Dept. of Electrical and Computer Engineering, Colorado State University, USA (e-mail: Edwin.Chong@ColoState.Edu).
I. Introduction

Information is being propagated across networks, such as online social networks, with increasing speed owing to increased network connectivity. There has been recent increased interest to study the spread or diffusion of information, influence, or infections across a network. For example, information dynamics and social learning have been investigated in [1]–[7], most of which assume knowledge of the network structure. Finding the sources of an infection diffusing in a network has been investigated in various works, including [8]–[16]. Most of these efforts again assume that the underlying network structure is known. In many practical applications, knowledge of the network structure may not be readily available, and only the infection times of the nodes may be known. A goal of interest is to infer the network structure based only on this set of infection times, and possibly also limited a priori statistical information or assumptions.

In this paper, we consider the inference of a network topology based on knowledge of the time each vertex in the network receives a piece of information, when an information diffusion is initiated from a known source vertex. More precisely, the network is modeled by an undirected graph $G = (V, E)$, with $V$ the set of vertices and $E$ the set of edges. A source vertex initiates an information diffusion, where the information is spread from an “infected” vertex to its neighbors stochastically along the edges connecting them. The time at which each vertex of $G$ receives the information is observed. We call this the infection time of the vertex, and a collection of infection times with their corresponding source vertex a cascade. We wish to use these information cascades to estimate the connections among vertices of the graph.

The network inference problem has been investigated under various assumptions. For example, [17], [18] considered the inference of directed graphs with a continuous-time diffusion model. The diffusion along each edge is assumed to be exponential or satisfies a power law. In [19], the authors assumed that the network can be undirected, and, along each edge, information spreads in two steps: a Bernoulli selection step and an exponential transmission step. Other related works based on different diffusion models include [18], [20], [21]. In most of these efforts, knowledge of the probability distribution of information spreading along edges is assumed, and many also assumed the spreading along different edges to be identically distributed. In practice, the spreading distribution is not easy to quantify. For example, information diffusion in an online social network depends on a host of factors [1]. However, moments—such as the average amount of time information takes to diffuse from a vertex to a neighbor—are typically
readily estimated from historical data [22], [23]. In this paper, we consider the case where we have limited knowledge of the spreading distribution and require only that the diffusion along different edges be independent. The distribution of information diffusion along edges is unknown to an observer, and only certain moments of the distribution (instead of the full distribution) are given. Therefore, the statistical inference approaches developed in [17]–[20] cannot be applied directly to our problem. Owing to recent advancements in the field of graph signal processing (see [24] for an overview), several network topology inference methods using graph signals have been proposed (for example, [25]–[27]). These approaches are based on the idea that certain graph signals are closely related to eigenvectors of graph-shift operators of a graph (e.g., graph adjacency matrix, graph Laplacian). However, in our setting, the timestamps are related to distance from vertices of a graph to a fixed source node. Such timestamps are usually not directly related to the above-mentioned shift-operators, and depend on the choice of the source node. Therefore, the graph signal processing approaches cannot be applied to our problem.

We approach the problem by first determining a necessary condition for perfect tree reconstruction based only on distance information. We introduce the concept of a separating vertex set, denoted by $V_S$, and show that if $|V_S|$ is sufficiently large, we can reconstruct the tree using the distances of vertices from those in $V_S$. Because the mean of the infection times are proportional to the distances from a source vertex to other vertices in the graph, we can estimate the distances from the information cascades. We propose an algorithm that makes use of the concept of redundant vertices to reduce the estimation variance. We further extend the method heuristically to general graphs by building on the tree reconstruction algorithm. Such an extension is supported by theoretical results obtained in the paper.

The rest of this paper is organized as follows. In Section II, we introduce our graph model and assumptions. In Section III, we introduce the notion of a separating vertex set and derive a necessary condition for perfect tree reconstruction. In Section IV, we introduce the notion of redundant vertices and use that to develop an iterative tree inference algorithm in Section V. We further extend the algorithm to general graphs in Section VI. Theoretical results supporting the extension are also discussed in Section VI. We present simulation results in Section VII and conclude in Section VIII.
II. Graph model

Let $G = (V, E)$ be a connected simple graph (i.e., undirected graph containing no graph loops or multiple edges), with $V$ the set of vertices and $E$ the set of edges. Let $d(\cdot, \cdot)$ be the length of a shortest path between nodes $u$ and $v$. We use $[u, v]$ to denote any shortest path between $u$ and $v$, and $(u, v)$ the path with end vertices $u, v$ excluded. A path $P$ between $u$ and $v$ is called simple if $P$ does not cross itself; thus, $[u, v]$ is always a simple path. For later use, we introduce the following definitions (see Figure 1 for an example).

**Definition 1.** The set of leaf or boundary vertices, denoted by $\partial G$, are vertices of degree 1. The set of branched or ramified vertices, denoted by $B_G$, are vertices of degree at least 3. The union $S_G = \partial G \cup B_G$ is called the special vertices of $G$. The remaining vertices are of degree 2; we call them ordinary vertices.

If $v$ is a leaf and the unique neighbor of $v$ is ordinary, then we say that $v$ is a long leaf. The set of long leaves is denoted by $L_{\partial G}$.

**Definition 2.** We take $\partial B_G \subset B_G$ to be the subset of vertices having a simple path to a leaf without passing through other vertices of $B_G$:

$$\partial B_G = \{ v \in B_G \mid [u, v] \cap B_G = \{v\} \text{ for some } u \in \partial G \}.$$ 

Let $L_{\partial B_G}$ be the set of vertices $v \in \partial B_G$ such that none of the neighbors of $v$ are leaves.

Although $\partial B_G$ is defined for any graph, it is particularly useful when considering tree networks.

**Definition 3.** Given a subset of vertices $V' \subset V$, the convex hull $\text{conv}(V')$ of $V'$ in $V$ is the union of all simple paths connecting any pair of distinct vertices of $V'$.

A cascade $c$ consists of a single source vertex that initiates an information diffusion together with the times each vertex first receives the information. We assume that the information diffusion along different edges is independent with known means. Without loss of generality, we assume that the mean for every edge is $\mu$. We do not require the distributions along different edges to be identical or known.

We assume that each cascade persists long enough to infect all the vertices, and the following information is recorded: $T_c = \{ T_c(v) \in \mathbb{R} : v \in V \} \in \mathbb{R}^{|V|}$, where $T_c(v)$ is the first infection
Fig. 1. In $G$, the boundary $\partial G$ consists of the green vertices. The branched vertices $B_G$ are the red and yellow nodes, and $\partial B_G$ are the yellow nodes. The blue nodes are ordinary. Moreover, $v_1, v_2,$ and $v_3$ are long leaves, and $L_{\partial B_G} = \{u_1\}$ (note that $u_2 \notin L_{\partial B_G}$).

time of vertex $v$. We would like to infer the graph structure, by inferring the adjacency matrix $A_G$ of $G$.

For each vertex $v$, let $d_v(u) = d(v, u)$ denote the distance function (from any other node $u \in G$) to $v$. We have the following simple observation on $T_c$ and $d_v$, with $v$ being the source of $c$.

**Lemma 1.** If $G$ is a tree, then for each cascade $c$ starting at $v$, we have $\mathbb{E}[T_c(u)] = \mu d_v(u)$ for all $u \in V$.

**Proof.** For a vertex $u \in V$, because $G$ is a tree, there is a unique simple path $P$ connecting $v$ and $u$. The infection reaches $u$ from the source $v$ via the path $P$. If the length of $P$ is $d$, it consists of $d$ edges $e_1, \ldots, e_d$. Let $t_{e_i}$ denote the time it takes to pass the infection along $e_i$. Because the information spreading is independent along each edge, by the linearity of expectation, we have the following:

$$\mathbb{E}[T_c(u)] = \sum_{1 \leq i \leq d} \mathbb{E}[t_{e_i}] = \mu d,$$

and the proof is complete. □
The upshot of the lemma is that we can focus on distance functions associated with nodes of the graph.

III. RECONSTRUCTION ACCURACY AND SEPARATING VERTEX SET

We start this section with the following notion of “distance” associated with any subset of vertices of the graph.

**Definition 4.** Let $V' = \{v_1, \ldots, v_l\} \subset V$. For any two nodes $u$ and $v$, define their relative distance with respect to (w.r.t.) $V'$ as

$$d_{V'}(u, v) = \sup_{v_i \in V'} |d_{v_i}(u) - d_{v_i}(v)|.$$

By the triangle inequality associated with the usual absolute value, it is easy to verify that $d_{V'}(\cdot, \cdot)$ defines a pseudometric (which means that $d_{V'}(u, v)$ can be 0 for $u \neq v$) on the set of vertices $V$ of $G$.

In this section, using $d_{V'}(\cdot, \cdot)$, we aim to develop a necessary condition under which a tree $G$ can be reconstructed uniquely. To this end, we introduce the concept of a separating vertex set as follows (see Figure 2).

**Definition 5.** A set $V_S = \{v_i, 1 \leq i \leq l\}$ separates $G$ if for any distinct vertices $u, v \in V$, there exists $v_i \in V_S$ such that $d_{v_i}(u) \neq d_{v_i}(v)$. We say that $V_S$ is a separating vertex set.

Note that in the above definition, $G$ can be any graph. However, in the case of trees, we have the following result.
Lemma 2. Suppose that $G$ is a tree. Let $V' = \{v_1, \ldots, v_l\} \subset V$ be a set of distinct vertices in $V$. Then $V'$ separates $\text{conv}(V')$.

Proof. Let $u, v \in \text{conv}(V')$ be two distinct vertices. By the definition of convex hull, we can find four vertices $v_1, v_2, v_3, v_4 \in V'$ (not necessarily distinct) such that $u \in [v_1, v_2]$ and $v \in [v_3, v_4]$. Because $G$ is a tree, there is a unique simple path $P$ connecting $[v_1, v_2]$ and $[v_3, v_4]$ if they are disjoint. If $[v_1, v_2]$ and $[v_3, v_4]$ have a non-empty intersection, then take $P$ to be any vertex in the intersection. Let $u' = P \cap [v_1, v_2]$ and $v' = P \cap [v_3, v_4]$. Without loss of generality, we assume that $u \in [v_1, u']$ and $v \in [v', v_4]$. Therefore, $[v_1, v_4] = [v_1, u'] \cup [u', v'] \cup [v', v_4]$, and hence $u, v \in [v_1, v_4]$. Consequently, $d_{v_1}(u) \neq d_{v_1}(v)$ and $d_{v_4}(u) \neq d_{v_4}(v)$. By definition, $V'$ separates $\text{conv}(V')$. \hfill \qed

Next, we demonstrate how to approximate $G$ using a subset $V'$ and the pseudometric $d_{V'}(\cdot, \cdot)$.

Definition 6. Let $V' = \{v_1, \ldots, v_l\}$. We say that a graph $G' = (V, E')$ is reconstructed from $V'$ if any two vertices $u, v \in V$ are connected by an edge in $E'$ if and only if $d_{V'}(u, v) \leq 1$.

Theorem 1. Let $V' = \{v_1, \ldots, v_l\}$, and $G' = (V, E')$ be reconstructed from $V'$. Then the following holds true:

(a) $E \subseteq E'$.
Fig. 4. This figure illustrates the condition of Theorem 1(b). Suppose $V' = \{v_1, \ldots, v_5\}$. The red edges form $\text{conv}(V')$. Clearly, $\partial B'_G = \{v_5\}$ and $\partial \text{conv}(V') = \{v_1, \ldots, v_4\}$. Therefore, because $V'$ separates $G$, the conditions of Theorem 1(b) are satisfied.

(b) Suppose that $G$ is a tree. Let $\partial B'_G$ contain vertices of $\partial B_G$ such that for each $v \in \partial B'_G$, some neighbors of $v$ are not in $\text{conv}(V')$ (see Figure 4). Moreover, assume that $d(u, v) > 1$ for $u, v \in \partial B'_G \cup \partial \text{conv}(V')$. If $V'$ separates $G$, then $E' = E$, and hence $G' = G$.

(c) In the converse direction, suppose that $G$ is a general graph that does not contain any triangle (three pairwise connected vertices). If $E' = E$, then $V'$ separates $G$.

Proof. (a) Suppose that $u, v$ are connected by an edge $e$ in $E$. For each $v_i$, let $P_u$ be a geodesic connecting $v_i$ and $u$. Concatenating $P_u$ with $e$ gives a path (not necessarily simple) connecting $v_i$ and $v$, and therefore $d_{v_i}(v) \leq d_{v_i}(u) + 1$. The same argument switching the roles of $u$ and $v$ gives $d_{v_i}(u) \leq d_{v_i}(v) + 1$. Part (a) thus follows.

(b) By part (a), it suffices to show that $E' \subset E$. Let $u$ and $v$ be two vertices connected by an edge in $E'$. This means that for all $i = 1, \ldots, l$, we have $|d_{v_i}(u) - d_{v_i}(v)| \leq 1$. Suppose on the contrary that $u$ and $v$ are not connected by an edge in $E$. Because $V'$ separates $G$, we have that each connected component of $G \setminus \text{conv}(V')$ is a simple path; for otherwise, there will be two distinct vertices of $G \setminus \text{conv}(V')$ having the same distance to all $v_i \in V'$.

We first claim that both $u$ and $v$ are not in $\text{conv}(V')$. If on the contrary, $u \in \text{conv}(V')$, let $v'$ be the vertex in $\text{conv}(V')$ closest to $v$ (called the projection of $v$ onto $\text{conv}(V')$). As in the proof of Lemma 2, we see that there are $v_1, v_2 \in V'$ such that $[v', u] \subset [v_1, v_2]$. Without loss
of generality, assume that $u \in [v', v_1]$. Therefore $u \in [v, v_1]$ and $|d(v, v_1) - d(u, v_1)| \leq 1$ if and only if $d(u, v) = 1$.

If $u$ and $v$ are in the same component of $G \setminus \text{conv}(V')$, then the condition $|d_{v_1}(u) - d_{v_1}(v)| \leq 1$ easily implies that $u$ and $v$ are connected by an edge (notice that each component of $G \setminus \text{conv}(V')$ is a simple path if $V'$ separates $G$), which gives a contradiction.

Next, assume that $u$ and $v$ are in different components of $G \setminus \text{conv}(V')$. Let $u_1$ and $u_2$ be their respective closest vertex (projections) on $\text{conv}(V')$. Notice that $u_1, u_2 \in \partial B'_G \cup \partial \text{conv}(V')$.

According to the given condition, $d(u_1, u_2) > 1$. Without loss of generality, we assume that $d(u, u_1) \geq d(u_2, v)$. Moreover, we have seen in the proof of Lemma 2 that we can choose $v_1, v_2 \in V'$ such that $[u_1, u_2] \subset [v_1, v_2]$ and $u_1 \in [v_1, u_2]$. Therefore,

$$d_{v_2}(u) - d_{v_1}(v) = d(u_2, u) - d(u_2, v)$$

$$= d(u_2, u_1) + d(u, u_1) - d(u_2, v) > 1.$$ 

This contradicts the fact that $u$ and $v$ are connected by an edge in $E'$.

(c) Suppose on the contrary that $V'$ does not separate $G$. We can find two vertices $u_1$ and $u_2$ such that $d_{v_1}(u_1) = d_{v_1}(u_2), v_i \in V'$. Let $u_3$ be a vertex connected by an edge to either $u_1$ or $u_2$, say $u_1$. Then for each $v_i \in V'$, $|d_{v_1}(u_3) - d_{v_1}(u_2)| \leq |d_{v_1}(u_3) - d_{v_1}(u_1)| + |d_{v_1}(u_1) - d_{v_1}(u_2)| \leq 1$. Therefore, because $E = E'$, $u_3$ and $u_2$ are also connected by an edge. This contradicts the assumption that $G$ does not contain any triangle. The proof is now complete. 

---

Fig. 5. This illustrates the proof of Theorem 1 (b), when $u, v$ are in different components of $G \setminus \text{conv}(V')$. The red edges form the convex hull of $V' = \{v_1, \ldots, v_4\}$. Because $u$ and $v$ are not connected by an edge, in the proof, we find $v_1$ (or $v_2$) such that $|d(v_1, u) - d(v_1, v)| > 1$. 


Motivated by Theorem 1, we introduce the following quantifier to evaluate the effectiveness (using the same notation as in Theorem 1) of a given $V'$.

**Definition 7.** Let $V' \subset V$ and $G' = (V, E')$ be reconstructed from $V'$. We say that a subset of vertices $V_0 \subset V$ is *perfectly reconstructed* from $V'$ if the subgraphs spanned by $V_0$ in $G' = (V, E')$ and $G = (V, E)$ are the same. The *reconstruction accuracy* $r.a.(V')$ of $V'$ is defined as

$$r.a.(V') = \sup \left\{ \frac{|V_0|}{|V|} : V_0 \text{ perfectly reconstructed from } V' \right\}.$$ 

We have the following observations regarding $r.a.$ based on the definition and results obtained so far.

**Corollary 1.** Let $V' \subset V$.

(a) The entire graph $G = (V, E)$ is perfectly reconstructed from $V'$ if and only if $r.a.(V') = 1$.

(b) If $G$ is a tree and $r.a.(V') = 1$, then $V'$ is a separating vertex set.

(c) If $G$ is a tree and $G = \text{conv}(V')$, then $r.a.(V') = 1$.

**Proof.**

(a) Follows immediately from the definition.

(b) Follows from Theorem 1 (c).

(c) From Lemma 2 and noting that $\partial B'_G = \emptyset$ in Theorem 1(b), the result follows.

**Remark 1.** The condition $d_{V'}(u, v) \leq 1$ in Definition 6 is equivalent to $|d_{v_i}(u) - d_{v_i}(v)| \leq 1$ for all $v_i \in V'$. On the other hand, if $G$ is a tree, it is clear that for two vertices $u, v$ connected by an edge, $d(u, u') \neq d(v, u')$ for any $u' \in G$. Therefore, we can change the condition $d_{V'}(u, v) \leq 1$ in Definition 6 to $d_{V'}(u, v) = 1$ if $G$ is a tree.

Moreover, we have the following lower bound regarding the size of $V'$.

**Corollary 2.** Suppose that $G$ is a tree, $V' = \{v_1, \ldots, v_l\} \subset V$ and $\{d_{v_i}(\cdot) : v_i \in V'\}$ are given. Suppose that the following uniqueness property holds true: for any tree $G'$ on $V$ with associated distance function $d'(\cdot, \cdot)$ such that for any $v_i \in V'$ and $u \in V$, $d'(v_i, u) = d_{v_i}(u)$, we have $G' = G$. Then $l \geq |L_{\partial G}| - |L_{\partial B_G}|$ (see Definitions 1 and 2). In particular, if $l < |L_{\partial G}| - |L_{\partial B_G}|$, then $r.a.(V') < 1$.

**Proof.** Suppose on the contrary that $l < |L_{\partial G}| - |L_{\partial B_G}|$. By the pigeon-hole principle and Lemma 2, we can always find two simple paths $P_1$ and $P_2$ in $G \setminus \text{conv}(V')$ that are disjoint.
Fig. 6. This figure illustrates the proof of Corollary 2. The red edges form \( \text{conv}(V') \). The two paths \( P_1 \) and \( P_2 \) in the proof are the blue and green paths, respectively.

except where they intersect at a vertex \( v \in B_G \), and the length of at least one of them, say \( P_1 \), is larger than 1 (see Figure 6).

Therefore, we can find \( u_1 \in P_1 \) such that \( d(u_1, v) = 2 \), and \( u_2 \in P_2 \) such that \( d(u_2, v) = 1 \). Moreover, let \( u'_1 \) be the vertex between \( u_1 \) and \( v \). Because both \( P_1 \) and \( P_2 \) are not in \( \text{conv}(V') \), it is impossible to determine if \( u_1 \) is connected to \( u'_1 \) or \( u_2 \) based only on the information \( \{d(v_i(\cdot)) : v_i \in V'\} \), which contradicts the uniqueness property. The claim therefore holds. From Corollary 1(a), \( r.a.(V') = 1 \) implies the uniqueness property, and the last statement of the corollary follows from contrapositiveness.

For further insight, we randomly construct trees with 500 vertices. On average, the lower bound \( |L_{\partial G} - L_{\partial B_G}| \) is approximately 16% of the size of the tree. Heuristically, this demonstrates that on average if the size of \( V' \) is less than 0.16|\( V \)|, then the tree cannot be determined uniquely. On the other hand, even if the size of \( V' \) is larger than 0.16|\( V \)|, the nodes of \( V' \) must be carefully chosen to accomplish the reconstruction task. This suggests that perfect reconstruction of the entire network is in general difficult if insufficient information is available.

For any given \( V' \), we can estimate a lower bound of \( r.a.(V') \) using Corollary 1(c). We perform simulations on randomly generated trees with 500 nodes to investigate the relation between the size of \( V' \) and \( r.a.(V') \) (see Figure 7). In practice, we cannot determine a separating vertex set a priori, and achieving \( r.a.(V') = 1 \) is usually not possible. However, from Figure 7, we see that on average a set \( V' \) with reasonable size allows perfect reconstruction of a large part of the
IV. REDUNDANT VERTICES

Our tree inference algorithm is based on the infection times of cascades, whose means are proportional to the distances between vertices. In practice, we face the following problems: (1) owing to the stochastic nature of the diffusion process, the recorded infection times at various vertices are not exactly proportional to the distance to the source; (2) infection times provided by cascades starting at certain vertices may be redundant when considered together with timestamps given by cascades starting at other vertices. In this section, we introduce and discuss the concept of redundant vertices before presenting a method to mitigate the aforementioned shortcomings in our tree inference algorithm in Section V.

Definition 8. Let $V' = \{v_1, \ldots, v_l\}$ be a set of vertices. A vertex $v_i \in V'$ is called redundant w.r.t. $V'$ if $E'$ reconstructed (see Definition 6) using $V'$ and $V' \setminus \{v_i\}$ are the same. Two vertices $v_1, v_2 \in V'$ are called mutually replaceable if $E'$ reconstructed from $V' \setminus \{v_1\}$ and $V' \setminus \{v_2\}$ are the same.
Proposition 1. Suppose that $T$ is a tree. For $u, v \in V$, remove the first edge in the path $[u, v]$, and let $T^u_{v}$ be the subtree rooted at $u$.

Lemma 3. Let $V' = \{v_1, \ldots, v_l\} \subset V$.

(a) A vertex $v_i \in V'$ is redundant if and only if for each pair $u, v$ such that $|d_{u_i}(u) - d_{u_i}(v)| > 1$, there is $v_j \in V' \setminus \{v_i\}$ such that $|d_{u_j}(u) - d_{u_j}(v)| > 1$.

(b) If $v_i$ is redundant in $V'$, then $v_i$ is redundant in any set of vertices containing $V'$.

(c) Two vertices $v_1, v_2 \in V'$ are mutually replaceable if and only if both $v_1$ and $v_2$ are redundant w.r.t. $V'$.

Proof. (a) If $|d_{u_i}(u) - d_{u_i}(v)| > 1$, then $u, v$ are not connected to each other when we use $V'$. Therefore, $v_i$ is redundant if and only if they are not connected to each when we use $V' \setminus \{v_i\}$; or equivalently there is some $v_j \in V' \setminus \{v_i\}$ such that $|d_{u_j}(u) - d_{u_j}(v)| > 1$.

(b) This follows immediately from the criterion given in (1).

(c) According to the definition, $E'$ constructed from $V' \setminus \{v_1\}$ and $V' \setminus \{v_2\}$ are both the same as that constructed from $V'$.

Definition 9. Suppose $G$ is a tree. For $u, v \in V$, remove the first edge in the path $[u, v]$, and let $T^u_{v}$ be the subtree rooted at $u$.

Proposition 1. Suppose that $G$ is a tree, and $V' = \{v_1, \ldots, v_l\} \subset V$. Let $v'_i$ be the vertex in $\text{conv}(V' \setminus \{v_i\})$ closest to $v_i$ (i.e., $v'_i = \arg\min_{v \in \text{conv}(V' \setminus \{v_i\})} d(v, v_i)$).

(a) If $v_i = v'_i$ (i.e., $v_i \in \text{conv}(V' \setminus \{v_i\})$), then $v_i$ is redundant w.r.t. $V'$.

(b) Let $D_{v'_i}(r) = \{v \in V \mid d(v, v'_i) \leq r\}$, and $T^v_{v'_i}$ be the subtree rooted at $v'_i$ pointing away from $v_i$. If $v_i \neq v'_i$, and the open path $P = (v_i, v'_i)$ contains only ordinary vertices and $D_{v'_i}(2) \cap T^v_{v'_i} \subset \text{conv}(V' \setminus \{v_i\})$, then $v_i$ is redundant w.r.t. $V'$.

Proof. Suppose that we are given $u, v \in V$ such that $|d_{u_i}(u) - d_{u_i}(v)| > 1$. Let $u'$ and $v'$ be their respective closest points in $\text{conv}(V' \setminus \{v_i\})$. We have seen that we can always find $v_j, v_k \in V' \setminus \{v_i\}$ such that $[u', v'] \subset [v_j, v_k]$. Without loss of generality, we assume that $u' \in
We notice that \([u, u'] \cap \text{conv}(V' \setminus \{v_i\}) = u'\) because \(u'\) is the closest point; similarly, \([v, v'] \cap \text{conv}(V' \setminus \{v_i\}) = v'\).

Suppose that \(d(u', v') > 1\) and without loss of generality that \(d(u, u') \leq d(v, v')\). Therefore, we find \(|d_{v_j}(v) - d_{v_j}(u)| \geq d(u', v') > 1\). We are done in view of Lemma 3 (a).

For the remaining case where \(d(u', v') \leq 1\), we treat (a) and (b) separately.

(a) Suppose that \(d(u', v') \leq 1\) and \(v_i = v'_i\). Without loss of generality, we can further assume in this case that \(u' \in [v_i, v']\) (notice that this requires that \(d(u', v') \leq 1\)). In this case, we see that

\[
|d_{v_j}(v) - d_{v_j}(u)| = |d(u', u) - d(u', v)| = |d_{v_i}(v) - d_{v_i}(u)| > 1.
\]

(b) If both \(u'\) and \(v'\) are different from \(v'_i\), then the same argument as above does the job because, in this case, \(|d_{v'_i}(v) - d_{v'_i}(u)| = |d_{v_i}(v) - d_{v_i}(u)|\). Suppose that \(u' = v'_i\) (see Figure 8). By the definition of convex hull and the choice of \(u', v'\), we have \([u', v'] \subset [u, v]\). Let \(v''\) be the neighbor of \(v'\) on \([v, v']\); it is immediate that \(d(v'', u) = d(v'', v'_i) \leq 2\). Notice that \(P\) contains ordinary vertices. Therefore, the assumption asserts that there is a \(v_l \in V' \setminus \{v_i\}\) such that \(d(v_l, u') \geq 2\) and \(v'' \in [v', v_l]\).

If \(|d(u, u') - d(v, v')| > 2\), then either use \(v_k\) or \(v_j\) to satisfy Lemma 3 (1). Otherwise, \(|d(v''_l, u) - d(v'', v)| \geq 1\). It is easy to verify that

\[
|d_{v_l}(u) - d_{v_l}(v)| \geq |d(v'', u) - d(v'', v)| > 1.
\]

\[\square\]

![Fig. 8. This figure illustrates the proof of Proposition 1 (b) when \(d(u', v') \leq 1\). The red edges form \(\text{conv}(V')\). The other nodes follow the same notation as given in the proof; \(v_l\) is the node of \(V'\) such that \(|d(v_l, u) - d(v_l, v)| > 1\).](image)

Using the previous proposition, we can prove the following corollary.
**Corollary 3.** Suppose that $G$ is a tree. Let $\partial B'_G$ be any subset of $\partial B_G$ such that the distance between any two vertices of $\partial B'_G$ is at least 2. Moreover, if each element of $\partial B'_G$ has at most two non-leaf neighbors, then any $V'$ contains a subset $V''$ of size at most $|\partial G| - |\partial B'_G|$ (which is a constant that depends only on $G$) such that all the vertices not in $V''$ are redundant w.r.t. $V'$.

**Proof.** Let $T$ be the convex hull of $V'$ and $L$ the leaves of $T$. For each $u \in \partial B'_G \cap T$, we define the set $L_u \subset L$ as follows:

$$L_u = \{ w \in [v, u] \cap L \mid v \in \partial G \text{ such that } [v, u] \cap B_G = \{u\} \}.$$ 

In other words, $L_u$ consists of the leaves of $T$ to whom $u$ is the closest among all the members of $B_G$.

Suppose that $|L_u| = \deg(u) - 1$. If a neighbor $v$ of $u$ is not a leaf, then let $l_u$ be the (unique) node in $L_u$ such that $v \in [l_u, u]$; and otherwise, let $l_u$ be any node in $L_u$. Form $V''$ by removing these $l_u$ from $V'$ (see Figure 9). It is clear that $|V''| \leq |\partial G| - |\partial B'_G|$. Moreover, each other vertex is redundant by Proposition 1(a) and (b).  

![Diagram](image)

**Fig. 9.** This figure illustrates the proof of Corollary 3. The red edges form $\text{conv}(V')$. According to the construction, $L_u = \{v_1, v_2\}$ and $L_{u'} = \{u'\}$. We can remove $v_1$ from $V'$.

Simulation results on randomly generated trees with 500 nodes show that on average, the bound $|\partial G| - |\partial B'_G|$ given in Corollary 3 is at most 32% of $|V|$.

Proposition 1 should be modified if $G$ is a general graph. To be more precise, in addition to the given condition, we have to assume further that $v_i$ is not contained in a cycle (see Figure 10 for a counterexample).
Fig. 10. In this simple example, \( V' = \{v_1, v_2, v_3\} \). By definition, \( v_3 \in \text{conv}\{\{v_1, v_2\}\} \). However, \( v_3 \) is connected to its opposite vertex (the dotted line) if we do not use \( d_{v_3} \). If we include \( d_{v_3} \), the method described in Theorem 1 recovers the square. Hence, we conclude that \( v_3 \) is not redundant in \( V' \).

**Example 1.** A simple yet important example is when \( V' \) contains two vertices \( v_1, v_2 \) connected by a direct edge. We assume that \( |V'| \geq 3 \). Let \( v_3 \) be different from \( v_1 \) and \( v_2 \).

Because \( v_1 \) and \( v_2 \) are connected by a direct edge, without loss of generality, we can assume that \( v_2 \in \text{conv}\{v_1, v_3\} = [v_1, v_3] \). By Proposition 1(a), \( v_2 \) is redundant w.r.t. \( V' \).

In the following cases, we can also conclude that \( v_1 \) is redundant:

1. Either (a) or (b) of Proposition 1 holds for \( v_2 \). Notice that we always have that \( v'_1 \) is either \( v_1 \) or \( v_2 \).
2. If we construct \( E' \) using \( |d_{v_i}(u) - d_{v_i}(v)| = 1 \), then \( D_{v_1}(2) \) in Proposition 1(b) can be replaced by \( D_{v_2}(1) \).

To see this, suppose that \( u \) and \( v \) are not connected by an edge using \( v_1 \), but are connected by an edge using \( v_2 \). This can only happen when \( |d(u, v_2) - d(v, v_2)| = 1 \) and \( |d(u, v_1) - d(v, v_1)| = 3 \). The same argument as the the proof of Proposition 1 (use a neighbor of \( v_2 \) in place of \( v''_1 \) in the last two paragraphs; and see Figure 11 for an illustration) proves the claim.

If any of the above cases happens, and information diffusion happens deterministically, \( v_1 \) and \( v_2 \) are mutually replaceable w.r.t. \( V' \) in inferring the structure of the graph. Simulation results show that if \( |V'| \geq 0.3|V| \), then on average, more than 84% of pairs of \( V' \) connected by a direct edge are mutually replaceable. In practice, the diffusion process is stochastic. Time information provided by \( v_1 \) (respectively, \( v_2 \)) can be used to average out the time information at \( v_2 \) (respectively, \( v_1 \)) to obtain a better estimate of the distance with a lower variance.
Fig. 11. In the figure, $v_1$ tells that the nodes $u$ and $v$ cannot be connected by a direct edge; while $v_2$ does not. Therefore, we can conclude that $v_1$ is redundant if we have the additional node $v_l$ available.

V. Iterative Tree Inference Algorithm

From our discussion in the previous sections, in order to use infection-time information associated with cascades from various sources effectively, we propose the following general scheme if $G$ is a tree. More details are provided in the following discussion.

1. **Selection step**: Given $V' = \{v_1, \ldots, v_l\}$, find pairs of vertices $(v_i, v_j)$ having a “high chance” of being connected by a single edge.

2. **Transfer step**: For a pair of vertices $(v_i, v_j)$ connected by an edge, we use the cascade at $v_i$ to construct a new cascade at $v_j$, and average the infection times with those from the existing cascade at $v_j$. This is the where we use the theory developed in Section IV (in particular, Example 1).

3. **Reconstruction step**: Use the new timestamp information obtained in the previous steps to estimate $d_{v_i}(\cdot)$ for each $v_i \in V'$, and reconstruct the graph $G$ based on Definition 6.

Recall that for a cascade $c$, we use $T_c(u)$ to denote the time that node $u$ receives the information of the cascade. Let $V' = \{v_1, \ldots, v_l\}$ be a set of vertices containing the sources of all the cascades. If there are several cascades with the same source, we may average their infection times. Doing so, we denote the cascades by their respective sources for convenience. The timestamp information is therefore $\{T_{v_i}(\cdot) : v_i \in V'\}$. In view of Lemma 1, for each $i$, we use $T_{v_i}(\cdot)/\mu$ as an approximation of $d_{v_i}(\cdot)$.

We first discuss the reconstruction step. For $u, v \in V$, from Theorem 1 and Remark 1, to determine if $u, v$ are connected by an edge, we want to compare $|d_{v_i}(u) - d_{v_i}(v)|$ with 1 for
each \( v_i \in V' \). This motivates us to introduce the following:

\[
W(u, v) \triangleq \frac{\sum_{v_i \in V'} ||T_{v_i}(u) - T_{v_i}(v)||}{|V'|} / \mu - 1.
\] (1)

Here, we take the average difference with 1 instead of using \( \sup \) as in Definition 4. This is because when the timestamps are stochastically generated, taking \( \sup \) is sensitive to change of variance of distributions. Essentially, this is nothing but replacing the \( L^\infty \) pseudometric by the \( L^1 \) pseudometric.

If \( W(u, v) \) is small, it suggests a higher chance that \( u, v \) are connected by a direct edge. We call \( W \) the weight matrix, and its entries the weights. If we want to infer the structure of a tree of size \( n \), we can select \( n - 1 \) (total number of edges) pairs of distinct vertices \((u, v)\) such that \( W(u, v) = W(v, u) \) are the smallest \( n - 1 \) weights.

In doing so, we obtain an estimated graph \( G' = (V, E') \). We can directly use \( E' \) for the selection step. An alternative is to use the weight matrix \( W \) for the job. More precisely, we can set a numerical condition \( C \) based on \( W \), and select a pair \((u, v)\) as long as \( W(u, v) = W(v, u) \) satisfies the pre-set condition \( C \). For example, for a fixed parameter \( k \), we can choose \( kn \) pairs of \((u, v)\) with the smallest \( W(u, v) \) value. This reflects our belief that each selected \((u, v)\) has a “high chance” of being connected by an edge.

For a selected pair of vertices \((v_i, v_j)\) with \( v_i, v_j \in V' \) and \( u \in V \), we construct the times \( T_{v_i}^u(\cdot) \) at \( v_j \) using \( T_{v_j} \) in the transfer step as follows: for each \( u \in V \),

\[
T_{v_j}^{v_i}(u) = \arg \min_{x \in \{T_{v_i}(u) + T_{v_i}(v_j), T_{v_i}(u) - T_{v_i}(v_j)\}} |x - T_{v_j}(u)|. \tag{2}
\]

We can now update \( T_{v_j}(\cdot) \) by its average with \( T_{v_j}^{v_i}(\cdot) \). Once \( T_{v_j}(\cdot) \) are updated for every \( v_j \in V' \), we can repeat the reconstruction step.

We summarize the above discussion in as our Iterative Tree Inference (ITI) algorithm in Algorithm 1.

VI. THE CASE OF GENERAL GRAPHS

In this section, we consider general simple graphs. We first discuss theoretical results on information propagation on general graphs, which lend support to our heuristic extension of the ITI algorithm. We then propose a general graph inference algorithm.

We assume that \( G \) is undirected, and the information propagation along every edge of \( G \) is independently distributed according to an unknown continuous distribution with probability
Algorithm 1 Iterative tree inference (ITI) algorithm

1: Input $I$: number of transfer steps, $m_s$: number of edges selected for the selection step, $n$: number of vertices, $V'$: set of cascade sources, $\{T_{v_i}(\cdot) : v_i \in V'\}$: average infection time information of cascades.

2: Initialize $\eta_v = 0$ for each $v \in V'$.

3: for $s = 1, \ldots, I$ do

4: Compute weight matrix $W$ in (1) and select $m_s$ pairs $\{(v_i, v_j)\}$ having the $m_s$ smallest $W(v_i, v_j)$ weights.

5: for Each pair of $(v_i, v_j)$ selected do

6: Generate new cascades $T_{v_i}^{v_j}(\cdot)$ and $T_{v_j}^{v_i}(\cdot)$ as in (2).

7: Update $\eta_v = \eta_v + 1$ for $v = v_i, v_j$.

8: Update $T_{v_i}(\cdot) = (\eta_v T_{v_i}(\cdot) + T_{v_i}^{v_j}(\cdot)) / (\eta_v + 1)$ and $T_{v_j}(\cdot) = (\eta_v T_{v_j}(\cdot) + T_{v_j}^{v_i}(\cdot)) / (\eta_v + 1)$.

9: end for

10: end for

11: Select $n - 1$ edges $\{(u, v)\}$ corresponding to the $n - 1$ smallest $W(u, v)$ weights.

12: Output $\{(u, v)\}$.

density function (pdf) $f$ with mean $\mu$. We also assume that $f$ has infinite support. The continuous distributions commonly used in the literature to model diffusions (for example, [17], [18]) satisfy this assumption.

As a notational convention, we use a lower-case letter (e.g., $f$) to denote the pdf of a continuous probability distribution, and the corresponding capital letter (e.g., $F$) for its cumulative distribution function (cdf). Moreover, we let $\bar{F} = 1 - F$.

A. Theoretical observations

Let $u, v$ be two distinct vertices of a graph $G$. We use $X_{u,v}$ to denote the random variable associated with the time it takes for a piece of information to propagate from $u$ to $v$. Let $f_{u,v}$ be the density function of the associated distribution. Because $G$ is undirected, $f_{u,v} = f_{v,u}$. Consider the case where there are multiple paths from $u$ to $v$, but no edge $(u, v)$ in $G$. The propagation time $X_{u,v}$ is then the minimum of the propagation times along each of these paths. We first show that it is always possible to differentiate this case with the case where $u$ and $v$ are connected only by a single edge. To do this, we need the following elementary result.
Lemma 4. Let $m, n, k$ be positive integers. Then for some $x \in (0, 1)$, we have

$$x^m + x^n - x^{m+n} > x^{1/k}.$$

Proof: Set $h(x) = x^m + x^n - x^{m+n} - x^{1/k}$. The first-order derivative is

$$h'(x) = mx^{m-1} + nx^{n-1} - (m+n)x^{m+n-1} - 1/kx^{1/k-1}.$$  

We find immediately that $h'(1) = -1/k < 0$. As $h'(x)$ is continuous in a small neighborhood containing 1, we have $h'(x) < 0$ in a small interval $(1 - \epsilon, 1)$ for some $0 < \epsilon < 1/2$. Because $h(1) = 0$, the mean value theorem allows us to conclude that $h(1 - \epsilon) > 0$.

Theorem 2. Suppose that the propagation time along each edge is identically distributed with the common density function $f$. For two distinct vertices $u, v$, if there is a path connecting $u, v$ containing a vertex different from both $u$ and $v$, then $f_{u,v} \neq f$.

Proof: It suffices to prove that $\bar{F} \neq \bar{F}_{u,v}$. Suppose that the contrary is true. If $u$ and $v$ is connected by an edge, the mean propagation time will be strictly smaller than that of $f$ as we assume $f$ has infinite support (a more general result is given in Lemma 5 below). For the rest of the proof, we assume $(u, v) \notin E$.

![Fig. 12](image.png)

Fig. 12. This illustrates the proof of Theorem 2. In $G$, we shrink the two red edges to points, giving $G_1$. Two additional blue edges are added in $G_1$ to give $G_2$.

We first reduce the graph $G$ to a simpler graph (see Figure 12 for illustration). Construct $G_1$ as follows: if both end points of an edge $e$ in $G$ are different from $u$ and $v$, then shrink $e$ to a single point (take note that this is different from removing $e$) in $G_1$. As $u, v$ are not connected
by an edge in $G$, each path between $u$ and $v$ is of length 2 in $G_1$. We should take note that multiple edges are allowed between two nodes in $G_1$ (and $G_2$ constructed below). The shrinking process reduces the time to travel from $u$ to $v$ and hence $\bar{F}_{u,v} \geq \bar{F}_1$.

Let $C$ be a connected component of $G_1 \backslash \{u,v\}$. It is easy to see that $C$ is made up of a single node $v_C$, with $m_C$ paths connecting $v_C$ to $u$, and $n_C$ paths connecting $v_C$ to $v$. Denote the total number of connected components of $G_1 \backslash \{u,v\}$ by $k$. Define $m = \max_C \{m_C\}$ and $n = \max_C \{n_C\}$. Construct $G_2$ as follows: for each $C$, we add $m - m_C$ edges between $v_C$ and $u$, and add $n - n_C$ edges between $v_C$ and $v$. As we add additional edges between nodes without changing the rest of the graph, we have $\bar{F}_1 \geq \bar{F}_2$. The number of connected components of $G_2 \backslash \{u,v\}$ is still $k$.

Let $f^{[n]}$ be the first-order derivative of $1 - \bar{F}^n$. As $\bar{F}(x) = 1$ if $x \leq 0$, for $t > 0$, we have the following

\[
\bar{F}_{u,v}(t) \geq \bar{F}_1(t) \geq \bar{F}_2(t) \\
= \left( \int_0^t \bar{F}^m(t-x)f^{[n]}(x)dx \right)^k \\
= \left( \int_0^t \bar{F}^m(t-x)f^{[n]}(x)dx + \int_t^\infty \bar{F}^m(t-x)f^{[n]}(x)dx \right)^k \\
= \left( \int_0^t \bar{F}^m(t-x)f^{[n]}(x)dx + \int_t^\infty f^{[n]}(x)dx \right)^k \\
\geq \left( \bar{F}^m(t) \int_0^t f^{[n]}(x)dx + \bar{F}^n(t) \right)^k \\
= (\bar{F}^m(t)(1 - \bar{F}^n(t)) + \bar{F}^n(t))^k \\
= (\bar{F}^m(t) + \bar{F}^n(t) - \bar{F}^{m+n}(t))^k.
\]

Because $\bar{F}(t)$ is continuous, $\bar{F}(0) = 1$ and $\lim_{t \to \infty} \bar{F}(t) = 0$. By the intermediate value theorem, $(0,1]$ is contained in the image of $\bar{F}$. Therefore, from Lemma 4, we obtain a contradiction and the theorem is proved.

Theorem 2 suggests that even if the mean propagation time (as was used in the ITI algorithm) for $X_{u,v}$ is similar to that of a single edge, we can use higher moments to determine if $u$ and $v$ are connected by a single edge (except for certain pathological distributions that share the same moments). In the following, we provide bounds on the moments of $X_{u,v}$ to guide us in our extension of the ITI algorithm to general graphs.
Lemma 5. Suppose that \((u, v) \in E\). Consider the graph \(G'\) in which \((u, v)\) is removed, and suppose that \(G'\) is connected. Let \(Y_{u,v}\) be the propagation time from \(u\) to \(v\) in \(G'\) and its pdf be \(h\). Then for all \(k \geq 1\), and any \(\epsilon_0 > \epsilon_1 > 0\),
\[
\mathbb{E}[Y_{u,v}^k] - \mathbb{E}[X_{u,v}^k] \geq \epsilon_1 F((\epsilon_0 - \epsilon_1)^{1/k}) \bar{H}(\epsilon_0^{1/k}).
\]

Proof: We have
\[
\mathbb{E}[Y_{u,v}^k] - \mathbb{E}[X_{u,v}^k] = \int_0^\infty \int_0^\infty (y^k - \min(x, y)^k) h(y)f(x)dydx
\]
\[
= \int_0^\infty \int_x^\infty (y^k - x^k) h(y)f(x)dydx
\]
\[
\geq \int_0^\infty \int_{(x^k + \epsilon_1)^{1/k}}^\infty (y^k - x^k) h(y)f(x)dydx
\]
\[
\geq \epsilon_1 \int_0^\infty \int_{(x^k + \epsilon_1)^{1/k}}^\infty h(y)f(x)dydx
\]
\[
= \epsilon_1 \int_{\epsilon_1^{1/k}}^\infty \int_0^\infty f(x)h(y)dydx
\]
\[
= \epsilon_1 \int_{\epsilon_1^{1/k}}^\infty \int_0^\infty F((y^k - \epsilon_1)^{1/k}) h(y)dy
\]
\[
\geq \epsilon_1 F((\epsilon_0 - \epsilon_1)^{1/k}) \int_{\epsilon_1^{1/k}}^\infty h(y)dy
\]
\[
= \epsilon_1 F((\epsilon_0 - \epsilon_1)^{1/k}) \bar{H}(\epsilon_0^{1/k}),
\]
where the interchange of integration in the third equality follows from Tonelli’s theorem. The lemma is proved.

In particular, by choosing \(\epsilon_0 - \epsilon_1\) to be sufficiently large and noting that \(\bar{H}(\cdot) > 0\) (as \(h\) has infinite support), Lemma 5 implies that \(\mathbb{E}[Y_{u,v}^k] > \mathbb{E}[X_{u,v}^k]\) for all \(k \geq 1\). Moreover, if the graph \(G\) is not too dense in the sense that \(F(1) \bar{H}(1)\) is suitably large, then the difference between the means of \(Y_{u,v}^k\) (without an edge between \(u\) and \(v\)) and \(X_{u,v}^k\) (with an edge between \(u\) and \(v\)) can be made suitably large by choosing a sufficiently large \(k\). This allows us to determine if \((u, v) \in E\) based on empirical observations of the propagation times raised to the \(k\)-th power. However, we may not observe information cascades from every vertex \(u\) in the network, and typically have to rely on information cascades starting at sources other than \(u\) and \(v\). We have the following bound relating the propagation times to \(u\) and \(v\) from a distinct source node.
Lemma 6. Let $G_{u,v}$ be the union of all the simple paths connecting $u,v$ (i.e., the convex hull, according to Definition 3) in $G$; and $w \in G$. Then for any $k \geq 1$,

$$|E[X_{w,u}] - E[X_{w,v}]|^k \leq E[X_{u,v}^k].$$

Proof: For any vertices $a,b$, let $P_{a,b}$ be the path associated with $X_{a,b}$. The concatenation of $P_{u,w}$ and $P_{u,v}$ is a path from $w$ to $v$ with possibly some edges repeated. We therefore have almost surely,

$$X_{w,u} + X_{u,v} \geq X_{w,v} \Rightarrow X_{w,v} - X_{w,u} \leq X_{u,v}.$$ 

Similarly, $X_{w,u} - X_{w,v} \leq X_{u,v}$ almost surely. We then obtain

$$|E[X_{w,u}] - E[X_{w,v}]|^k \leq E[|X_{w,u} - X_{w,v}|^k] \leq E[X_{u,v}^k],$$

where the first inequality follows from Jensen’s inequality, and the lemma is proved. 

Lemma 6 shows that if $E[X_{u,v}^k]$ is small, then so is the more computable $|E[X_{u,w}] - E[X_{v,u}]|^k$ (recall that we estimate $E[X_{u,w}]$ using $T_{v}(u)$ and the transfer step in the ITI algorithm in Section V). The reverse implication is not necessarily true, but for algorithmic convenience, the lemma suggests that we use an empirical estimate of the latter term as a proxy for $E[X_{u,v}^k]$.

B. Discussions and implications

We now discuss some implications of the results obtained in Section VI-A. Consider any pair of distinct nodes $u,v$, and the following cases:

(i) Suppose that the number of paths between $u$ and $v$ is small (i.e., the graph $G$ is sparse).

   a) If $u$ and $v$ are connected by an edge, then the sample mean and moments of the propagation time between $u$ and $v$ approximate well the mean and corresponding moments of a diffusion across a single edge.

   b) On the other hand, if $u$ and $v$ are not connected by a direct edge, then the sample mean and variance of the propagation time between $u$ and $v$ are close to integer multiples of the mean and variance of a diffusion across a single edge.

(ii) Suppose that the number of paths between $u$ and $v$ is large (i.e., the graph $G$ is dense).

   According to Lemma 5 and the discussions thereafter, the existence of an edge between
and $v$ can make both the sample mean and higher moments small relative to the corresponding moment values if such an edge is missing. Therefore, in this case, we may choose to infer that the edge $(u, v)$ exists based on the size of the sample mean and higher moments. However, we should mention that if there are too many paths between $u$ and $v$, then in practice, any distribution-based estimation is prone to errors as demonstrated in the discussion and example below.

Our next example demonstrates that it is almost impossible to determine if there exists an edge $(u, v)$ in the graph $G$ if it is very dense. To show this, we need the following result.

**Lemma 7.** Suppose that $X$ and $Y$ are two continuous random variables on $(0, \infty)$, with cdf $P$ and $H$ respectively. Let $Z = \min\{X, Y\}$. Then the total variation distance between the distributions of $Y$ and $Z$ is bounded from above by $\inf_{\epsilon>0} \{\bar{H}(\epsilon) + P(\epsilon)\}$.

**Proof:** Let $p$ and $h$ be the pdf of $P$ and $H$ respectively. It is easy to show that the pdf of the distribution associated with $Z$ is $p(x)(1 - H(x)) + h(x)(1 - P(x)) - h(x)$. Therefore, we have to show that for each $\epsilon > 0$,

$$\int_0^\infty |p(x)(1 - H(x)) + h(x)(1 - P(x)) - h(x)| dx \leq 2(\bar{H}(\epsilon) + P(\epsilon)).$$

To see this, we first apply the triangle inequality:

$$\int_0^\infty |p(x)(1 - H(x)) + h(x)(1 - P(x)) - h(x)| dx \leq \int_0^\infty h(x)P(x)dx + \int_0^\infty p(x)(1 - H(x))dx. \quad (3)$$

The two integrals on the right hand side of (3) can be bounded separately as:

$$\int_0^\infty h(x)P(x)dx = \int_0^\epsilon h(x)P(x)dx + \int_\epsilon^\infty h(x)P(x)dx \leq P(\epsilon) \int_0^\epsilon h(x)dx + \int_\epsilon^\infty h(x)dx \leq P(\epsilon) + \bar{H}(\epsilon),$$

$$\int_0^\infty p(x)(1 - H(x))dx = \int_0^\epsilon p(x)(1 - H(x))dx + \int_\epsilon^\infty p(x)(1 - H(x))dx \leq \int_0^\epsilon p(x) + \bar{H}(\epsilon) \int_\epsilon^\infty p(x)dx \leq P(\epsilon) + \bar{H}(\epsilon).$$

The result follows by adding up the two inequalities.
Example 2. Suppose that the propagation along each edge are i.i.d. with exponential distribution having mean 1. Then, we have \( F(x) = 1 - e^{-x} \). Assume that there are \( k \) independent paths of length \( l \) between two distinct nodes \( u \) and \( v \), which are not connected by an edge. Along each path, the propagation follows a Gamma distribution \( \Gamma(l, 1) \), whose cdf is \( 1 - e^{-x} \sum_{i=0}^{l-1} x^i/i! \). Let \( Y \) be the propagation time between \( u \) and \( v \). Then, it can be shown that its complementary cdf \( \bar{H}(\epsilon) \leq (e^{-\epsilon} \sum_{i=0}^{l-1} \epsilon^i/i!)^k \).

Let \( Z \) be the propagation time from \( u \) to \( v \) if the edge \((u,v)\) is added to the graph. By Lemma 7, the total variation distance between \( Y \) and \( Z \) is bounded from above by

\[
\inf_{\epsilon > 0} \left( e^{-\epsilon} \sum_{i=0}^{l-1} \epsilon^i/i! \right)^k + 1 - e^{-\epsilon}. \tag{4}
\]

We have \( \lim_{\epsilon \to 0^+} (1 - e^{\epsilon}) = 0 \). On the other hand, the derivative of \( e^{-x} \sum_{i=0}^{l-1} x^i/i! \) is \( -e^{-x}x^{l-1}/(l-1)! < 0 \) for \( x > 0 \). This means that \( e^{-\epsilon} \sum_{i=0}^{l-1} \epsilon^i/i! < 1 \) for any \( \epsilon > 0 \). Therefore, if \( l \) is fixed, we can always choose \( \epsilon \) small enough and \( k \) large enough such that the upper bound (4) is as close to 0 as we wish. This suggests that in practice, if there are many paths between the two nodes, then it is almost impossible to determine if there is an edge between them or not by any distribution-based method.

As a specific numerical example, if \( l = 2 \) and \( k = 100 \), (4) drops below 0.3.

C. The graph inference algorithm

The discussion in Section VI-B can be summarized in the following dichotomy: when the graph is sparse (as measured by the edge to vertex ratio, for example) or the distributions of the propagation times along each edge have small variance, it is enough to use the mean of the distributions as in the case of trees. On the other hand, if the graph is highly connected, the existence of an edge between two vertices \( u \) and \( v \) can make the mean propagation time between \( u \) and \( v \) small relative to \( \mu \). Therefore, it is instructive to use the length of the propagation time between \( u \) and \( v \) to decide if they are connected by an edge or not. The same consideration applies to other moments (as compared against unbiased sample moments); and they can be used as additional criteria to decide the existence of edges.

In a general graph, it is usually unknown whether the connection between two vertices \( u \) and \( v \) is dense or sparse. One way to overcome such a difficulty is to compute the difference between the sample mean (respectively, sample moments) with both the theoretical mean (respectively, theoretical moments) as well as 0. Once these two values are obtained, it is enough to take the
smaller one. Hence the weight matrix being used in ITI should be modified based on available moment information. Suppose that the mean $\mu$ and variance $\sigma^2$ of propagation time along edges are known. We define the following:

\[
W_1(u, v) \triangleq \min(\text{ave } |T_{v_i}(u) - T_{v_i}(v)|, \\
\text{ave } (|T_{v_i}(u) - T_{v_i}(v)| - \mu),
\]

\[
W_2(u, v) \triangleq \min(\text{ave } |T_{v_i}(u) - T_{v_i}(v)|^2, \\
\text{ave } ||T_{v_i}(u) - T_{v_i}(v)||^2 - (\sigma^2 + \mu^2)),
\]

where “ave” is used to denote average over $v_i \in V'$ as in (1) to shorten the expressions; and

\[
W(u, v) \triangleq W_1(u, v) + W_2(u, v). \tag{5}
\]

In the case of general graphs, we usually do not know the total number of edges based on the number of vertices $n$, in contrast to tree networks. Therefore, we first randomly choose a small percent of vertices and find their average degree $\text{deg}_s$. We take $\text{deg}_s$ to be an estimate of the actual average degree of the network; and estimate the total number of edges as $n_e = n \cdot \text{deg}_s$.

Based on these, we propose below the Graph Inference (GI) algorithm (Algorithm 2) as a heuristic extension of ITI.

A possible generalization of Algorithm 2 if higher order moments are available is to modify $W_1, W_2$ and hence $W$ accordingly as follows. According to Lemma 6 and the discussion thereafter, we may use $|T_{v_i}(u) - T_{v_i}(v)|$ to estimate the sample moments. For each $k \geq 1$, the $k$-th sample moment is denoted by $S_{u,v}^{(k)}(v_i)$ as the average of $|T_{v_i}(u) - T_{v_i}(v)|^k$ over $v_i \in V'$. The $k$-th moment of the distribution (of propagation time) along each edge is denoted by $\mu_k$.

We fix a finite set of $m$ positive numbers: \{ $k_1 = 1, k_2, \ldots, k_m$ \}; and a continuous $m$-variable function $\phi$. Define for each $1 \leq i \leq m$:

\[
W_i(u, v) \triangleq \min(\text{ave } S_{u,v}^{(k_i)}(v_i), \\
\text{ave } |S_{u,v}^{(k_i)}(v_i) - \mu_{k_i}|);
\]

and $W$ is obtained by:

\[
W(u, v) \triangleq \phi(W_1, \ldots, W_m). \tag{6}
\]

Under this generalization, the procedure depicted in Algorithm 2 uses $m = 2$, $k_1 = 1$, and $k_2 = 2$, while $\phi$ is the averaging function. The choice of $\phi$ should reflect one’s belief about which moment should play a more important role in the network inference task.
Algorithm 2 Graph Inference (GI) algorithm

1: **Input** $I$: number of transfer steps, $m_s$: number of edges selected for the selection step, $n_e$: estimated number of edges of $G$, $\{T_{v_i}\}$: time information of cascades.

2: Initialize $\eta_v = 0$ for each $v \in V'$.

3: **for** $s = 1, \ldots, I$ **do**

4: Compute weight matrix $W$ in (5) and select $m_s$ pairs $\{(v_i, v_j)\}$ having the $m_s$ smallest $W(v_i, v_j)$ weights.

5: **for** Each pair of $(v_i, v_j)$ selected **do**

6: Generate new cascades $T_{v_i}^{v_j}(\cdot)$ and $T_{v_j}^{v_i}(\cdot)$ as in (2).

7: Update $\eta_v = \eta_v + 1$ for $v = v_i, v_j$.

8: Update $T_{v_i}(\cdot) = (\eta_v T_{v_i}(\cdot) + T_{v_i}^{v_j}(\cdot))/(\eta_v + 1)$ and $T_{v_j}(\cdot) = (\eta_v T_{v_j}(\cdot) + T_{v_j}^{v_i}(\cdot))/(\eta_v + 1)$.

9: **end for**

10: **end for**

11: Select $n_e$ edges $\{(u, v)\}$ corresponding to the $n_e$ smallest $W(u, v)$ weights.

12: **Output** $\{(u, v)\}$.

VII. SIMULATION RESULTS

In this section, we present simulation results to verify the performance of our proposed topology inference algorithms. We first apply our ITI algorithm on tree networks, and compare its performance with with the tree reconstruction (TR) algorithm proposed in [19], which is most similar to ours in assumptions. We then perform simulations on general graphs, including some real-world networks, and compare the performance of the GI algorithm with the the NetRate algorithm proposed in [18]. As the NetRate algorithm assumes knowledge of the diffusion distribution, we also study the performance impact when there is a mismatch between the assumed and actual distributions.

Suppose that $A_G$ is the true adjacency matrix of $G$ and $A$ is an estimated adjacency matrix. To evaluate the performance of our method, we define the edge recovery rate as

$$ R \triangleq 1 - \frac{\sum_{1 \leq i < j \leq n} |A(i, j) - A_G(i, j)|}{2|E|}. \quad (7) $$

For the same number of edges, each mistake in identifying an edge causes a mistake at another pair of vertices. To account for this, we have a factor of 2 in the denominator; and this makes $R$
a real number in $[0, 1]$. The term $1 - R$ is called the error rate, in which both undetected edges and false positives are taken into account.

A. Tree networks

In this subsection, we show and discuss simulation results to verify the performance of our proposed algorithm for tree networks. As we are considering tree networks, in the reconstruction step, we always have $|E| = n - 1$ edges, where $n$ is the number of vertices.

In each simulation run, we randomly generate trees with $n = 500$ vertices. We then randomly choose $n_c$ distinct sources, and generate $\kappa_v$ cascades per source using independent exponential spreading with mean $\mu = 1$ at each edge. In applications, $\kappa_v$ can be different for distinct nodes $v$; however, in our simulations below, we keep it constant for all $v$ for simplicity. We apply the ITI algorithm, and evaluate the performance by using (7).

We first study the edge recovery rate by varying $n_c$ and $\kappa_v$, and skip the selection and transfer steps by setting $I = 0$ (see Figure 13). We notice that $\kappa_v$ has a significant impact on the performance. If the number of cascades per source $\kappa_v$ is small, the performance can be greatly improved if we perform additional transfer steps (see Figures 14). These results demonstrate the usefulness of the theory developed in Section IV. The choice $I = 2$ is usually enough, and the number of edges $m_s$ in the selection step can be chosen between $n$ and $1.5n$. Moreover, comparing the curves for $n_c/n = 0.3, \kappa_v = 2$ and $n_c/n = 0.5, \kappa_v = 1$, we see that the performance is improved if we have more cascades originating from the same source. This suggests that the averaging process allows a better approximation of the distance on a tree.

Finally, we compare our method with the TR algorithm proposed in [19]. Although the theoretical part of [19] assumes i.i.d. exponential distribution for propagation time along different edges, the TR algorithm itself requires $n$ and $\{T_v\}$ as the only input. The comparison is shown in Figures 15 and 16. In Figure 16, we test the effect of having different diffusion distributions for different edges. The distributions are randomly selected unknown Gamma distributions with the same mean. As shown from the plots, our method performs much better in all the cases because our method does not require identical distributions for propagation times along different edges.
Fig. 13. Performance of the algorithm with $I = 0$ with varying $n_c$ and $\kappa_v$.

Fig. 14. Performance of the algorithm with varying $n_c$, $\kappa_v$, $I$ and $m_s$. Curves with the same color have the same $n_c$ and $\kappa_v$ values.
Fig. 15. Performance comparison between the TR algorithm (yellow) and the ITI algorithm (blue), under the same simulation setup.

Fig. 16. Performance comparison between the TR algorithm (yellow) and the ITI algorithm (blue). The distribution of propagation time along each edge follows a randomly selected unknown Gamma distribution.

B. General graphs

For general graphs, we compare our GI algorithm with the NetRate algorithm proposed in [18].\footnote{The source code for NetRate was retrieved from SNAP (Stanford Network Analysis Project). We thank the authors of [18] for sharing it online.} Although NetRate works under a completely different set of assumptions (in particular,
[18] assumes propagation along edges follows one of the following families of distributions: exponential, power-law, or Rayleigh), it proves to be very powerful in graph topology inference using cascade information.

To accommodate comparison with NetRate, we use the standard exponential distribution with rate $\frac{1}{\mu} = 1$ for propagation time along each edge. We compare the performance of NetRate and GI on Erdős-Rényi graphs, the Forest-fire network [28], and a real-world Email network [28]. The parameters of all networks are described in the plots.

From the plots (Figures 17, 18, 19, and 20), we see that our method performs better in all the tested cases if the number of cascades does not exceed 60% of the number of nodes. However, we should mention that if the ratio is closer to 1 (i.e., on average each node sends a cascade), then NetRate can have a better performance for certain graph types (Figures 17 and 19). On the other hand, for ER-graphs with large average degree and the Email network, GI performs better for the entire spectrum of $\frac{n_c}{n}$ from 10% to 100%. The GI algorithm has a noticeably better performance when $\frac{n_c}{n}$ is very small, as compared with NetRate. For example, in the case of the Email Network and $\frac{n_c}{n} = 10\%$, NetRate has less than 5% edge recovery while the GI has more than 27% edge recovery. We note that since the Email network is dense, all inference methods based on the assumption that diffusion across each edge follows a distribution will have limited performance (cf. Example 2).

We would like to point out another important advantage of our method. Because of the nature of the two algorithms, our method is computationally much more efficient. For example, using the same computational resource under the same simulation settings, the time used to run GI and NetRate is a few seconds versus a few hours. Therefore, our method is more suitable in time-critical applications when computational resource is limited.

C. Distribution mismatch

For the next set of simulations, we test the effect of distribution mismatch on NetRate. To be precise, as our algorithm does not require knowledge of the type of distribution governing the diffusion process while NetRate does, we want to test whether such prior knowledge is crucial to NetRate.

The setup of the simulations is as follows: on each type of network, we generate the cascades using the exponential distribution with rate 1. We run the NetRate algorithm with the incorrect distributions: either the power-law distribution (POW) or the Rayleigh distribution (RAY), as
Fig. 17. Performance comparison between the NetRate-algorithm (blue) and the GI-algorithm (red), on ER-graphs with 300 nodes and average degree 4. The information propagation along the edges follow the exponential distribution with parameter $\lambda = 1$. The dashed curves show the performance of NetRate if there is a distribution mismatch for comparison purpose.

Fig. 18. Performance comparison between the NetRate algorithm (blue) and the GI algorithm (red), on ER-graphs with 300 nodes and average degree 8. The information propagation along the edges follow the exponential distribution with rate 1. The dashed curves show the performance of NetRate if there is a distribution mismatch for comparison purposes.

these are the other two families of distributions discussed in [18]. The results are shown in the same Figures 17, 18, 19, and 20, by dashed curves.
Fig. 19. Performance comparison between the NetRate algorithm (blue) and the GI algorithm (red), on Forest-fire network with 500 nodes and average degree about 5. The information propagation along the edges follow the exponential distribution with rate 1. The dashed curves show the performance of NetRate if there is a distribution mismatch for comparison purposes.

Fig. 20. Performance comparison between the NetRate algorithm (blue) and the GI algorithm (red), on a real-world Email network with 500 nodes and average degree 12. The information propagation along the edges follow the exponential distribution with rate 1. The dashed curves show the performance of NetRate if there is a distribution mismatch for comparison purposes.

From the plots, we see that the performance of NetRate drops significantly if there is a distribution mismatch. For example, if the power-law distribution is used, the edge recovery of
NetRate is close to 0% regardless of $n_c/n$ and the network type.

The experiments suggest that prior knowledge of the diffusion distribution is important to guarantee the performance of NetRate. In contrast, for the GI algorithm, no such prior knowledge is required.

VIII. CONCLUSION

In this paper, we have developed a theory and method for graph topology inference using information cascades and knowledge of some moments of the diffusion distribution across each edge, instead of the distribution itself. In the case of tree networks, we provided a necessary condition for perfect reconstruction, and used the concept of redundant vertices to propose an iterative tree inference algorithm. Our method works well as long as the total number of cascades is not too small compared to the size of the network. Moreover, our method works better if there are more cascades at each chosen source. Simulations demonstrate that our method outperforms the tree reconstruction algorithm in [19]. We have also provided some theoretical insights into how the moments of the propagation time between two vertices in a general graph behave, and extended our tree inference method heuristically to general graphs. Our simulation results suggest that our graph inference algorithm performs reasonably well, if the total number of cascades is not too small compared to the size of the network, and often outperforms the NetRate algorithm in [18]. Moreover, our method is suitable for time-critical applications owing to its low complexity.

REFERENCES

[1] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, “Information diffusion in online social networks: A survey,” SIGMOD Rec., vol. 42, no. 2, pp. 17–28, Jul. 2013.

[2] D. W. Soh, W. P. Tay, and T. Q. S. Quek, “Randomized information dissemination in dynamic environments,” IEEE/ACM Trans. Netw., vol. 21, no. 3, pp. 681 – 691, Jun. 2013.

[3] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2003, pp. 137–146.

[4] A. Java, P. Kolari, T. Finin, and T. Oates, “Modeling the spread of influence on the blogosphere,” in Proc. Int. conf. world Wide Web, 2006, pp. 22–26.

[5] J. Leskovec, L. A. Adamic, and B. A. Huberman, “The dynamics of viral marketing,” ACM Trans. Web, vol. 1, no. 5, pp. 1–39, 2007.

[6] W. P. Tay, “Whose opinion to follow in multihypothesis social learning? A large deviations perspective,” IEEE J. Sel. Topics Signal Process., vol. 9, no. 2, pp. 344 – 359, Mar. 2015.
[7] J. Ho, W. P. Tay, T. Q. S. Quek, and E. K. P. Chong, “Robust decentralized detection and social learning in tandem networks,” *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5019–5032, Oct. 2015.

[8] 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.

[9] W. Dong, W. Zhang, and C. W. Tan, “Rooting out the rumor culprit from suspects,” in *Proc. IEEE Int. Symp. Inf. Theory*, July 2013, pp. 2671–2675.

[10] W. Luo, W. P. Tay, and M. Leng, “Identifying infection sources and regions in large networks,” *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2850–2865, 2013.

[11] W. Luo and W. P. Tay, “Finding an infection source under the SIS model,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, May 2013, pp. 2930–2934.

[12] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, “Inferring the origin of an epidemic with a dynamic message-passing algorithm,” *Phys. Rev. E*, vol. 90, p. 012801, Jul 2014.

[13] W. Luo, W. P. Tay, and M. Leng, “How to identify an infection source with limited observations,” *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 586–597, 2014.

[14] ———, “Rumor spreading and source identification: A hide and seek game,” *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4228–4243, 2016.

[15] 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, 2017.

[16] W. Tang and W. P. Tay, “A particle filter for sequential infection source estimation,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, Mar. 2017.

[17] M. G. Rodriguez, J. Leskovec, and A. Krause, “Inferring networks of diffusion and influence,” in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2010, pp. 1019–1028.

[18] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, “Uncovering the temporal dynamics of diffusion networks,” in *Proc. ACM Int. Conf. Machine Learning*, 2011, pp. 561–568.

[19] B. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi, “Trace complexity of network inference,” in *Proc. ACM SIGKDD Int. Conf. Knowledge discovery and data mining*, 2013, pp. 491–499.

[20] P. Netrapalli and S. Sanghavi, “Learning the graph of epidemic cascades,” *SIGNETRICS Perform. Eval. Rev.*, vol. 40, pp. 211–222, Jun. 2012.

[21] V. Gripon and M. Rabbat, “Reconstructing a graph from path traces,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2013, pp. 2488–2492.

[22] D. Gruhl, R. Guha, D. L. Nowell, and A. Tomkins, “Information diffusion through blogspace,” in *Proc. ACM Int. Conf. World Wide Web*, 2004.

[23] D. Centola, “The spread of behavior in an online social network experiment,” *Science*, vol. 329, no. 5996, pp. 1194–1197, 2010.

[24] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[25] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Laplacian matrix learning for smooth graph signal representation,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, April 2015, pp. 3736–3740.

[26] V. Kalofolias, “How to learn a graph from smooth signals,” in *Proc. Int. Conf. Artificial Intell. and Stat.*, May 2016, pp. 920–929.
[27] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, “Network topology identification from spectral templates,” in *Proc. IEEE Statistical Signal Process. Workshop*, Jun. 2016.

[28] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007.