Statistical Estimation of Diffusion Network Topologies

Keqi Han[†] Yuan Tian[‡] Yunjia Zhang[§] Ling Han[†] Hao Huang[†] Yunjun Gao[#]
[†]*School of Computer Science, Wuhan University, China*
[‡]*School of Mathematics and Statistics, Wuhan University, China*
[§]*Department of Computer Sciences, University of Wisconsin-Madison, USA*
[#]*College of Computer Science and Technology, Zhejiang University, China*
{hankeqi, yuan.tian, hanl, haohuang}@whu.edu.cn, yunjia@cs.wisc.edu, gaoyj@zju.edu.cn

Abstract—**Reconstructing the topology of a diffusion network based on observed diffusion results is an open challenge in data mining. Existing approaches mostly assume that the observed diffusion results are available and consist of not only the final infection statuses of nodes, but also the exact timestamps that pinpoint when infections occur. Nonetheless, the exact infection timestamps are often unavailable in practice, due to a high cost and uncertainties in the monitoring of node infections. In this work, we investigate the problem of how to infer the topology of a diffusion network from only the final infection statuses of nodes. To this end, we propose a new scoring criterion for diffusion network reconstruction, which is able to estimate the likelihood of potential topologies of the objective diffusion network based on infection status results with a relatively low statistical error. As the proposed scoring criterion is decomposable, our problem is transformed into finding for each node in the network a set of most probable parent nodes that maximizes the value of a local score. Furthermore, to eliminate redundant computations during the search of most probable parent nodes, we identify insignificant candidate parent nodes by checking whether their infections have negative or extremely low positive correlations with the infections of a corresponding child node, and exclude them from the search space. Extensive experiments on both synthetic and real-world networks are conducted, and the results verify the effectiveness and efficiency of our approach.**

Index Terms—**diffusion network, topology, influence relationship, infection timestamp**

I. Introduction

The topology of a diffusion network describes how the nodes in the network influence each other. Knowledge of these influence relationships is crucial for understanding the properties of diffusion dynamics and for designing effective strategies to promote or prevent future diffusions on the network. Nonetheless, in many real-world settings, such as idea and disease propagation, the influence relationships between human users are not naturally accessible, and need to be recovered based on diffusion results observed from historical diffusion processes. This problem is often referred to as diffusion network reconstruction or diffusion network inference, and has received considerable attention in recent years in areas such as social networks [1], information propagation [2], epidemic prevention [3], and viral marketing [4].

To infer the influence relationships in diffusion networks, most existing approaches to diffusion network reconstruction assume that the infection of a node is caused by previously infected nodes with a high probability [5]. According to this assumption, nodes that are infected sequentially within a time interval are considered to have influence relationships, and the previously infected ones are regarded as potential parent nodes of the subsequently infected ones. Therefore, in order to use these approaches, users need to monitor each diffusion process and record exact occurrence timestamps of node infections. However, monitoring real-world diffusion processes is not always feasible or affordable, especially when the diffusion processes are long, the spatial distribution of nodes is wide, or the monitoring is labor/resource demanding. Furthermore, due to some unavoidable uncertainties in monitoring, such as different incubation periods (i.e., the time from infection to the appearance of observable signs or symptoms), the observed timestamps do not usually reflect the exact occurrence time of each infection.

To reconstruct the topologies of diffusion networks without infection timestamps, two existing approaches try to learn influence relationships between nodes, either from all path traces of fixed length [6], or from initial and resulting sets of infected nodes [7]. Nevertheless, an exact diffusion path is often hard to trace when multiple paths coexist in a diffusion process, let alone obtaining all path traces of fixed length for the former approach. On the other hand, the diffusion sources (i.e., the initially infected nodes) are usually unavailable, not to mention that the later approach requires an extra prior knowledge on the amount of influence relationships in objective network, which is also difficult to obtain in practice.

Aiming at a more widely applicable solution to diffusion network reconstruction, we investigate the problem of how to infer the topology of a diffusion network based on only the final infection statuses of nodes, which are more easily accessible in most cases. We present an effective and efficient approach called TENDS (which is an anagram of the bold letters in **S**tatistical **E**stimation of **D**iffusion **N**etwork **T**opologies) to solving this problem. Instead of studying sequential relationships of node infections, TENDS finds influence relationships with high statistical significance. To this end, we propose a new scoring criterion to estimate the fits between potential topologies of the objective diffusion network and observed

2375-026X/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDE48307.2020.00060

infection status results. It is designed to balance the likelihood and statistical error of an inferred network topology, and thus can help our TENDS algorithm to accurately find the most probable diffusion network topology. Meanwhile, as the scoring criterion is decomposable, the task of finding the most probable diffusion network topology can be transformed into finding for each node in the network a set of most probable parent nodes that maximizes the value of a local score. In addition, based on the scoring criterion, we can also derive a theoretical upper bound on the number of most probable parent nodes, which will avoid an overly complex inferred network topology that is conceptually and computationally intractable. Furthermore, to eliminate redundant computations during the search of most probable parent nodes, for each node in the network, TENDS identifies its insignificant candidate parent nodes by checking whether their infections have negative or extremely low positive correlations with the infections of this node, and then excludes these insignificant candidates from the search space of its most probable parent nodes.

In summary, our key contributions include the following: (1) We propose a statistical approach to infer diffusion network topologies from only the final infection statuses of nodes. Compared with existing approaches, it does not rely on monitoring the exact infection timestamps of nodes, and does not require any extra information, such as all path traces of fixed length, diffusion sources and a prior knowledge on diffusion network topologies, which are often difficult to access in practice. (2) We design a decomposable scoring criterion for diffusion network reconstruction. It decomposes the reconstruction process into finding each node a limited number of most probable parent nodes. (3) We present a heuristic method to prune the search space of the most probable parent nodes and help reduce redundant computations.

The remainder of the paper is organized as follows. We review the related work in Section II and present a problem statement in Section III. We then elaborate the proposed TENDS algorithm in Section IV, and report experimental findings in Section V before concluding the paper in Section VI.

II. Related Work

Existing approaches to diffusion network reconstruction can be classified into two groups: (1) approaches based on infection timestamps, and (2) approaches without infection timestamps.

A. Approaches Based on Infection Timestamps

Most existing approaches to diffusion network reconstruction utilize the sequences of node infections (known as cascades) to infer potential parent-child influence relationships. Therefore, they need to know the exact infection timestamp of each infected node in every diffusion process. According to solution strategies adopted, these approaches can be categorized into three main types: (1) the convex programming-based approaches, (2) the submodularity-based approaches, and (3) the embedding-based approaches.

Convex programming-based approaches try to find a diffusion network topology that maximizes the likelihood of given cascades based on convex optimization. To approximate the optimal solution, these approaches utilize different techniques, such as sequential quadratic programming [8], [9], the EM algorithm [10], [11], block coordinate descent [12], stochastic and proximal gradient methods [13], [14], survival theory [15], sparse recovery [16], and decoupling into multiple parallelizable problems [17]–[19], to solve their optimization problems. These approaches generally exhibit nice inference performance on tree-like or sparse networks.

Submodularity-based approaches transform the problem of diffusion network reconstruction into a problem of submodular optimization, as they use likelihood functions of cascades for given propagation trees that have the property of submodularity. NetInf [20] and MulTree [21] are state-of-the-art approaches of this type. Due to the submodularity of their objective functions, both approaches adopt a greedy algorithm to achieve a near-optimal solution. The main difference between them is that during the submodular optimization, NetInf considers only the most probable propagation tree, to achieve high efficiency, while MulTree considers all propagation trees supported by each cascade, to achieve high accuracy.

Embedding-based approaches map the nodes in observed diffusion process into a latent embedding space, in which the distance between each two mapped nodes represents the propagation probability (or transmission rate). These approaches model the propagation probabilities using Weibull distributions [22], uniform distributions [1], or via kernels [23], and they learn the propagation probabilities between nodes based on observed cascades. Although embedding-based approaches do not explicitly reveal the diffusion network topologies, they enable users to observe influence relationships between nodes via low-dimensional spaced visualizations.

The above three types of paradigms for diffusion network reconstruction all require complete and correct cascades. Abrahao et al. have proven that with an adequate amount of complete and correct cascades, the objective diffusion network can be inferred accurately using simple reconstruction approaches [24]. Nevertheless, in reality, observed cascades may have partially incorrect infection timestamps, and they may miss partial snapshots of the network. Several methods have been proposed to mitigate the effects of partially incorrect [25] or missing infection timestamps [26], [27]. These methods are complementary to the above three types of approaches.

Departing from the existing approaches based on infection timestamps, the proposed TENDS algorithm requires nothing but only the final infection statuses of nodes, which are more easily accessible in most real-world diffusion cases. Therefore, TENDS has a wider range of applicability, and is also unaffected by incorrect and missing infection timestamps.

B. Approaches without Infection Timestamps

To infer diffusion network topologies without infection timestamps, two existing works have partially address this problem by learning the influence relationships between nodes, either from diffusion path traces (referred to as the PATH approach), or based on lifting effects (referred to as the LIFT approach) from the initially infected nodes to the finally infected nodes.

PATH takes as inputs path-connected node sets, each of which contains a fixed number of nodes that are activated along a diffusion path through a network. It inserts edges between the nodes that co-occur most frequently in the path-connected node sets [6]. This approach has nice properties such as a solid mathematical foundation and low computational cost. Nonetheless, it requires complete path-connected node sets, which are often unaccessible in natural diffusion processes. Even if complete and correct cascades are available, inferring exact path-connected node sets is still difficult.

LIFT studies the problem of diffusion network reconstruction in the case that only diffusion sources and final infection statuses of nodes are available [7]. It calculates the lifting effect of a source node u to an infected node v, which measures the increase in the probability of v's infection on the condition that u is initially infected. A larger lifting effect indicates a higher probability that the corresponding two nodes have an influence relationship. Furthermore, LIFT requires a prior knowledge on the amount of influence relationships in the objective diffusion network, otherwise it will iteratively add influence relationships until each two nodes have an influence relationships.

Compared with PATH and LIFT, our TENDS algorithm only requires infection status results, and does not rely on any other extra information on node infections or prior knowledge on objective diffusion networks. Therefore, TENDS is more widely applicable in practice.

III. Problem Statement

A diffusion network can be represented as a directed graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ denotes the set of n nodes in the network, and E refers to the set of m directed edges between nodes. A directed edge from a node $v_i \in V$ to a node $v_j \in V$ indicates that v_i has an influence relationship to v_j. Specifically, when v_i is infected and v_j is uninfected, v_i will infect v_j with a certain probability, which is known as propagation probability or transmission rate.

In the problem of diffusion network reconstruction, the node set is given, while edge set and propagation probabilities are unknown and needed to be inferred. To reconstruct a diffusion network, a set of diffusion results observed from historical diffusion processes on the network is required. In this paper, we assume that the diffusion results contain only infection status results, i.e., the final infection statuses of nodes observed at the end of each diffusion process. Furthermore, as a few existing approaches have presented how to quantify the propagation probability for a specific edge based on observed infection status results [28], we focus on inferring the unknown topology (i.e., edge set) of the objective diffusion network.

Formally, our problem statement can be formulated as follows (Table I lists notation that will be used henceforth).

TABLE I
Notation

Symbol	**Description**
G	A directed graph.
V	The set of nodes in G.
n	The number of nodes in G.
v_i	The i-th node in V $(1 \leqslant i \leqslant n)$.
E	The set of directed edges in G.
m	The number of directed edges in G.
S	The infection statuses of nodes in G observed across β diffusion processes.
T	A potential topology of G.
α	The initial infection ratio of nodes.
β	The number of diffusion processes on G.
s_i^ℓ	The infection status of node v_i in the ℓ-th diffusion process $(1 \leqslant i \leqslant n, 1 \leqslant \ell \leqslant \beta)$.
X_i	The infection status variable of node $v_i \in V$.
F_i	The parent node set of node v_i.
$\|F_i\|$	The number of nodes in F_i.
X_{F_i}	The set of infection status variables of nodes in F_i.
π_i^ℓ	The infection statuses of v_i's parent nodes in the ℓ-th diffusion process $(1 \leqslant i \leqslant n, 1 \leqslant \ell \leqslant \beta)$.
π_{ij}	The j-th combination of the infection statuses of nodes in F_i.
N_{ijk}	The number of times situation $X_i = s_k \wedge X_{F_i} = \pi_{ij}$ appears in S $(k \in \{1, 2\}, s_1 = 0, s_2 = 1)$.
N_{ij}	The number of instances of π_{ij} in S $(N_{ij} = N_{ij1} + N_{ij2})$.
$L(v_i, F_i)$	The likelihood of a parent node set F_i for node v_i.
$g(T)$	The scoring criterion for topology T.
$g(v_i, F_i)$	The local score for parent node set F_i of node v_i.
$MI(X_i, X_j)$	The mutual information (MI) between the variables X_i and X_j.
$IMI(X_i, X_j)$	The infection MI between the infections of nodes v_i and v_j.
τ	A threshold for infection MI.
P_i	The set of candidate parent nodes of v_i $(\forall v_j \in P_i, IMI(X_i, X_j) > \tau)$.
C_i	The set of possible combinations of v_i's candidate parent nodes.

Given: a set $S = \{S^1, ..., S^\beta\}$ of infection status results observed on a diffusion network G in β historical diffusion processes, where $S^\ell = (s_1^\ell, ..., s_n^\ell)$ is a n-dimensional vector that records the final infection status, $s_i^\ell \in \{0, 1\}$ (0 denotes uninfected, and 1 denotes infected) of each node $v_i \in V$ observed at the end of the ℓ-th diffusion process ($\ell \in \{1, \ldots, \beta\}$).

Infer: the unknown edge set E of diffusion network G.

IV. The TENDS Algorithm

In this section, we first explain how to measure the likelihood of potential diffusion network topologies by proposing a new scoring criterion, followed by introducing how to eliminate redundant computations when using the scoring criterion to reconstruct diffusion network topology. Then, we present the detailed steps of the TENDS algorithm, and conclude this section with a complexity analysis on the algorithm.

A. Scoring Criterion

Let matrix $T \in \mathbb{R}^{n\times n}$ denote a network topology variable of the objective diffusion network $G = (V, E)$. Each element $T_{ij} \in \{0, 1\}$ $(i, j \in \{1, \dots, n\})$ in this matrix indicates whether there is a directed edge from node $v_i \in V$ to node $v_j \in V$ (1 for yes, 0 for no). Then, diffusion network reconstruction using infection status results S is equivalent to the problem of finding a optimal T that maximizes the following probability:

$$\max_{T} \hat{P}(S \mid T) \tag{1}$$

As each historical diffusion process is independent to each other, each S^ℓ is generated independently. Therefore, the probability $\hat{P}(S \mid T)$ can be reformulated as follows.

$$\begin{aligned}\hat{P}(S \mid T) &= \prod_{\ell=1}^{\beta} \hat{P}(S^\ell \mid T)\\ &= \prod_{\ell=1}^{\beta} \hat{P}(X_1 = s_1^\ell, \dots, X_n = s_n^\ell \mid T)\end{aligned} \tag{2}$$

where $X_i \in \{0, 1\}$ $(i \in \{1, \dots, n\})$ refers to the infection status variable of node v_i.

In a diffusion network, the infection of each node can be only caused by its parent nodes. Therefore, the relationship $P(X_1, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_{F_i})$ holds, where F_i refers to the parent node set of node v_i in current topology T, and X_{F_i} represents the infection status variables of the parent nodes of v_i. Then, the probability $\hat{P}(S \mid T)$ can be further reformulated as follows.

$$\begin{aligned}\hat{P}(S \mid T) &= \prod_{\ell=1}^{\beta}\prod_{i=1}^{n} \hat{P}(X_i = s_i^\ell \mid X_{F_i} = \pi_i^\ell)\\ &= \prod_{i=1}^{n}\prod_{j=1}^{2^{|F_i|}}\prod_{k=1}^{2} \hat{P}(X_i = s_k \mid X_{F_i} = \pi_{ij})^{N_{ijk}}\\ &= \prod_{i=1}^{n}\prod_{j=1}^{2^{|F_i|}}\prod_{k=1}^{2} \Big(\frac{N_{ijk}}{N_{ij}}\Big)^{N_{ijk}}\end{aligned} \tag{3}$$

where π_i^ℓ refers to the infection statuses of v_i's parent nodes in the ℓ-th diffusion process, $s_k \in \{0, 1\}$ refers to the k-th possible infection status of a node (without loss of generality, let $s_1 = 0, s_2 = 1$), $2^{|F_i|}$ is the number of all possible combinations of the infection statuses of v_i's parent nodes, π_{ij} represents the corresponding j-th possible combination, N_{ijk} is the number of times situation $X_i = s_k \wedge X_{F_i} = \pi_{ij}$ appears in the observed infection status results S, $N_{ij} = N_{ij1} + N_{ij2}$, and $\forall v_i, \sum_{j=1}^{2^{|F_i|}} N_{ij} = \beta$. In addition, as some combinations of the infection statuses of nodes in F_i may not have instances in S, we denote the number of these non-existent combinations as ϕ_{F_i}. It can be obtained by traversing S and checking how many of the $2^{|F_i|}$ possible combinations have instances in S.

Let $L(v_i, F_i) = \prod_{j=1}^{2^{|F_i|}} \prod_{k=1}^{2} \big(\frac{N_{ijk}}{N_{ij}}\big)^{N_{ijk}}$. The value of $L(v_i, F_i)$ reflects the likelihood of parent node set F_i for node v_i. Then, according to Eq. (3), to maximize the value of probability $\hat{P}(S \mid T)$, we should find for each node $v_i \in V$ a set F_i of parent nodes that maximizes the value of $L(v_i, F_i)$. However, we find that the value of $L(v_i, F_i)$ will be maximized when all the other nodes in node set V are added into F_i. This observation can be explained by the following lemma and theorem.

Lemma 1: For any non-negative integers a_1, a_2, b_1, b_2, the relationship

$$\Big(\frac{b}{a}\Big)^{b} \leqslant \Big(\frac{b_1}{a_1}\Big)^{b_1} \cdot \Big(\frac{b_2}{a_2}\Big)^{b_2} \tag{4}$$

always holds, where $a = a_1 + a_2$ and $b = b_1 + b_2$.

Proof. Since $\ln(\cdot)$ is a concave function and $\frac{b_1}{b} + \frac{b_2}{b} = 1$, then according to Jensen's inequality, the following relationship

$$\begin{aligned}\ln\frac{a}{b} &= \ln(\frac{b_1}{b}\cdot\frac{a_1}{b_1} + \frac{b_2}{b}\cdot\frac{a_2}{b_2})\\ &\geqslant \frac{b_1}{b}\ln\frac{a_1}{b_1} + \frac{b_2}{b}\ln\frac{a_2}{b_2}\end{aligned} \tag{5}$$

holds, and can be transformed as

$$b\ln\frac{b}{a} \leqslant b_1\ln\frac{b_1}{a_1} + b_2\ln\frac{b_2}{a_2}, \tag{6}$$

which is equivalent to

$$\Big(\frac{b}{a}\Big)^{b} \leqslant \Big(\frac{b_1}{a_1}\Big)^{b_1} \cdot \Big(\frac{b_2}{a_2}\Big)^{b_2}.$$

Therefore, the lemma is correct. ■

Theorem 1: Assume a diffusion network G with node set V and node infection status results S. Further assume that a node $v_i \in V$ has a parent node set F_i. Then, for any node $v_{i'} \in \{v \in V \mid v \notin F_i \cup \{v_i\}\}$, the relationship

$$L(v_i, F_i) \leqslant L(v_i, F_i \cup \{v_{i'}\}) \tag{7}$$

always holds.

Proof. Let $F_i' = F_i \cup \{v_{i'}\}$, then

$$L(v_i, F_i') = \prod_{j'=1}^{2^{|F_i|+1}} \prod_{k=1}^{2} (\frac{N_{ij'k}}{N_{ij'}})^{N_{ij'k}}. \tag{8}$$

For nodes in parent node set F_i, the j-th possible combinations of infection statuses is denoted as π_{ij}. Assume that the situations $(X_{F_i} = \pi_{ij}, X_{i'} = 0)$ and $(X_{F_i} = \pi_{ij}, X_{i'} = 1)$ correspond to the j_1'-th and j_2'-th possible combinations of infection statuses for nodes in F_i', respectively. Then, we have

$$\begin{aligned}N_{ijk} &= N_{ij_1'k} + N_{ij_2'k},\\ N_{ij} &= N_{ij_1'} + N_{ij_2'}.\end{aligned} \tag{9}$$

According to Lemma 1, the following relationship

$$\Big(\frac{N_{ijk}}{N_{ij}}\Big)^{N_{ijk}} \leqslant \Big(\frac{N_{ij_1'k}}{N_{ij_1'}}\Big)^{N_{ij_1'k}} \cdot \Big(\frac{N_{ij_2'k}}{N_{ij_2'}}\Big)^{N_{ij_2'k}} \tag{10}$$

always holds. Thus, we have

$$
\begin{aligned}
L(v_i, F_i) &= \prod_{j=1}^{2^{|F_i|}} \prod_{k=1}^{2} \Big(\frac{N_{ijk}}{N_{ij}}\Big)^{N_{ijk}} \\
&\leqslant \prod_{j=1}^{2^{|F_i|}} \prod_{k=1}^{2} \Big(\frac{N_{ij'_1k}}{N_{ij'_1}}\Big)^{N_{ij'_1k}} \cdot \Big(\frac{N_{ij'_2k}}{N_{ij'_2}}\Big)^{N_{ij'_2k}} \\
&= \prod_{j'=1}^{2^{|F_i|+1}} \prod_{k=1}^{2} \big(\frac{N_{ij'k}}{N_{ij'}}\big)^{N_{ij'k}} \\
&= L(v_i, F'_i)
\end{aligned}
\tag{11}
$$

and the theorem is correct. ■

According to Theorem 1, for node v_i, the maximum value of likelihood $L(v_i, F_i)$ can be achieved after adding all other nodes to the parent node set F_i. Therefore, if we simply pursue a higher probability $\hat{P}(S \mid T)$, the inferred topology T will become very complex and contain many parent-child influence relationships that may not exist in reality.

From the view of statistics, if more nodes are included in the parent node set F_i of node $v_i \in V$, there will be a larger statistical error for the computation of likelihood $L(v_i, F_i)$. The reason is that given an F_i then for every possible combination π_{ij} of the infection statuses of the nodes in F_i (where $1 \leqslant j \leqslant 2^{|F_i|}$), we need to count the number of instances from all the β historical diffusion processes to estimate the corresponding probability $\hat{P}(X_i = s_k \mid X_{F_i} = \pi_{ij})$ (where $s_k \in \{0,1\}$) for the computation of $L(v_i, F_i)$. Therefore, the number of probability estimations increases exponentially with the cardinality of F_i. In each probability estimation, a statistical error may be introduced when the number N_{ij} of corresponding instantiations is insufficient. For a given infection status results S, the more probability estimations to make, the fewer instances for each probability estimation, resulting in a larger statistical error. In brief, the introduced statistical errors are affected by two factors: (1) the number $2^{|F_i|}$ of possible infection status combinations of the nodes in set F_i and (2) the number N_{ij} of instances to be used for probability estimation.

To balance the likelihood and statistical error, we propose a new scoring criterion $g(T)$ for diffusion network reconstruction, which evaluates the quality of an inferred diffusion network topology T as follows.

$$
\begin{aligned}
g(T) &= \log \hat{P}(S \mid T) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1) \\
&= \sum_{i=1}^{n}\Big(\log L(v_i, F_i) - \frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1)\Big)
\end{aligned}
\tag{12}
$$

where the base of log is 2, using $N_{ij}+1$ is to avoid log of 0.

A higher value of $g(T)$ indicates that the current topology T is a better inference result. As the scoring criterion $g(T)$ is decomposable, maximizing the value of $g(T)$ is equivalent to maximizing the value of each local score $g(v_i, F_i)$, where

$$
g(v_i, F_i) = \log L(v_i, F_i) - \frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1). \tag{13}
$$

Note that according to Theorem 1, inclusion of more parent nodes for each node v_i will increase the value of likelihood $L(v_i, F_i)$. But, if there are too many parent nodes, the value of the penalty term $\frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1)$ tends to increase exponentially, and then decreases the value of local score $g(v_i, F_i)$. Thus, this penalty term can help us avoid adding too much nodes into set F_i, and prevent a high statistical error. Furthermore, for each node $v_i \in V$, the parent node set F_i selected by maximizing the value of local score $g(v_i, F_i)$ is asymptotically consistent with the true parent node set of v_i. This nice property can be theoretically explained by the following corollary.

***Corollary** 1:* Let $\hat{F}_i$ be the parent node set selected by the scoring criterion $g'(v_i, F_i) = \log L(v_i, F_i) - \frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \lambda$ for a given node $v_i \in V$ based on infection status results S of β historical diffusion processes, i.e., $\hat{F}_i$ maximizes the value of $g'(v_i, F_i)$ for a given v_i. If λ satisfies conditions

$$
\begin{aligned}
&\lim_{\beta\to\infty} \frac{\lambda}{\beta} = 0, \\
&\lim_{\beta\to\infty} \lambda = +\infty,
\end{aligned}
\tag{14}
$$

then $\hat{F}_i$ is a weakly consistent estimator of the true parent node set F_i^* of node v_i, i.e.,

$$
\lim_{\beta\to\infty} P(\hat{F}_i = F_i^*) = 1. \tag{15}
$$

Proof. Corollary 1 follows directly from Theorem 5 in reference [29]. ■

As $\log(N_{ij}+1)$ satisfies conditions $\lim_{\beta\to\infty} \frac{\log(N_{ij}+1)}{\beta} = 0$, and $\lim_{\beta\to\infty} \log(N_{ij}+1) = +\infty$, according to Corollary 1, the parent node set F_i selected using local score $g(v_i, F_i)$ tends to be more consistent with the true parent node set of v_i, when there are more historical diffusion processes used for diffusion network reconstruction.

To obtain an optimal F_i that maximizes local score $g(v_i, F_i)$, one should intuitively find a few parent nodes that are most likely to affect the infection of node v_i, and one should prevent the set of parent nodes from growing too large. In fact, a theoretical upper bound for the number of parent nodes can be derived from this local score $g(v_i, F_i)$.

***Theorem** 2:* Given infection status results S of β historical diffusion processes, in order to maximize the value of $g(v_i, F_i)$, the size $|F_i|$ of parent node set F_i of node v_i should satisfy condition

$$
|F_i| \leqslant \log(\phi_{F_i} + \delta_i), \tag{16}
$$

where

$$
\delta_i = 2N_1 \log\frac{\beta}{N_1} + 2N_2 \log\frac{\beta}{N_2} + \log(\beta+1), \tag{17}
$$

N_1 and N_2 respectively refer to the number of instances of situations $X_i = 0$ and $X_i = 1$ in S.

Proof. For an empty parent node set, the local score $g(v_i, \emptyset)$ can be calculated as

$$g(v_i, \emptyset) = \log\big(\frac{N_1}{\beta}\big)^{N_1} \cdot \big(\frac{N_2}{\beta}\big)^{N_2} - \frac{1}{2}\log(\beta+1). \tag{18}$$

If a non-empty parent node set F_i can maximize the value of $g(v_i, F_i)$, relationship $g(v_i, F_i) \geqslant g(v_i, \emptyset)$ should hold, which can be expressed as

$$\begin{aligned} &\log L(v_i, F_i) - \frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1) \\ &\geqslant N_1 \log\big(\frac{N_1}{\beta}\big) + N_2 \log\big(\frac{N_2}{\beta}\big) - \frac{1}{2}\log(\beta+1) \end{aligned} \tag{19}$$

For the nodes in F_i, there would be $2^{|F_i|}$ possible combinations of their infection statuses. In fact, partial combinations may have no instances in S, and we have denoted the number of these non-existent combinations as ϕ_{F_i}. If the j-th combination has no instances, i.e., $N_{ij} = 0$, then $\log(N_{ij}+1) = 0$; otherwise, $N_{ij} \geqslant 1$, and $\log(N_{ij}+1) \geqslant \log 2 = 1$. Then, the lower bound of $\frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1)$ is as follows.

$$\frac{1}{2}\sum_{j=1}^{2^{|F_i|}} \log(N_{ij}+1) \geqslant \frac{1}{2}\big(2^{|F_i|} - \phi_{F_i}\big). \tag{20}$$

Moreover, since $L(v_i, F_i) \leqslant 1$, we have

$$\log L(v_i, F_i) \leqslant 0. \tag{21}$$

Combining Eqs. (19)–(21), we have relationship

$$\frac{1}{2}\big(2^{|F_i|} - \phi_{F_i}\big) \leqslant -N_1 \log\big(\frac{N_1}{\beta}\big) - N_2 \log\big(\frac{N_2}{\beta}\big) + \frac{1}{2}\log(\beta+1) \tag{22}$$

which can be transformed as

$$|F_i| \leqslant \log\big(\phi_{F_i} + 2N_1 \log\frac{\beta}{N_1} + 2N_2 \log\frac{\beta}{N_2} + \log(\beta+1)\big). \tag{23}$$

Thus, the theorem is correct. ■

Given the local score $g(v_i, F_i)$ and Theorem 2, we can apply a greedy search procedure to find the most probable parent nodes for v_i. The procedure starts from an empty parent node set F_i, and expands F_i by iteratively adding a node combination (i.e., a subset of $V \setminus \{v_i\}$) that increases the value of the current $g(v_i, F_i)$ the most. If no candidate parent node for v_i exists or for any node subset $W \subseteq V \setminus \{v_i\}$, the number of nodes in set $F_i \cup W$ always exceeds the theoretical upper bound, i.e., $|F_i \cup W| > \log(\phi_{F_i \cup W} + \delta_i)$, then the greedy search procedure stops. In this way, we can efficiently achieve a locally optimal F_i. A similar greedy search procedure is used commonly in many other applications, such as influence maximization [30] and classification [28], due to its efficiency and good result quality.

B. Pruning Method

During the greedy search procedure, to find for each node $v_i \in V$ a candidate parent node or node set that can be added to its current parent node set F_i, a straightforward method is to traverse all node combinations from the candidate parent node set $V \setminus \{v_i\}$. This straightforward method is inefficient since there are $\sum_{i=1}^{n-1} \binom{n-1}{i}$ combinations, where n is the number of nodes in the network. Instead, we prune the candidate parent nodes to reduce the number of possible node combinations and avoid redundant computations.

Given the fact that the infections of nodes are only caused by their parent nodes with a certain probability, the infections of the parent nodes and corresponding child nodes should have positive correlations. In contrast, if the infection statuses of two nodes have a negative or extremely low positive correlation, there is a very low probability that these two nodes have an influence relationship between them.

To quantify the correlation between two variables, mutual information (abbreviated as MI) is a commonly used criterion and can be estimated as

$$MI(X_i, X_j) = \hat{P}(X_i, X_j) \log \frac{\hat{P}(X_i, X_j)}{\hat{P}(X_i)\hat{P}(X_j)} \tag{24}$$

$MI(X_i, X_j) \in [0, 1]$. A higher MI value indicates that variables X_i and X_j have a greater correlation. However, the correlation evaluated by MI is not equivalent to the positive correlation of infections. This is because when situations $X_i = 0$ and $X_j = 1$ (or $X_i = 1$ and $X_j = 0$) have a significantly great correlation, i.e., the infection statuses of nodes v_i and v_j have a significantly negative correlation, the value of MI still could be high.

In order to exactly evaluate the positive correlation between the infections of nodes, we modify the original MI metric as a new version called infection MI to measure the infection correlation. The infection MI between the infections of nodes v_i and v_j, denoted by $IMI(X_i, X_j)$, is defined as

$$\begin{aligned} &IMI(X_i, X_j) = \\ &MI(X_i = 1, X_j = 1) + MI(X_i = 0, X_j = 0) \\ &- |MI(X_i = 1, X_j = 0)| - |MI(X_i = 0, X_j = 1)|. \end{aligned} \tag{25}$$

Given the above definition of infection MI, when the infection statuses of nodes v_i and v_j have a significantly negative correlation, i.e., $|MI(X_i = 1, X_j = 0)|$ or $|MI(X_i = 0, X_j = 1)|$ has a significantly great value, the value of $IMI(X_i, X_j)$ tends to be negative. When infections of nodes v_i and v_j tend to be independent, the value of infection MI will be close to 0. When the value of $IMI(X_i, X_j)$ is a relatively great positive value, it indicates that the infections of nodes v_i and v_j have a positive correlation.

In a real-world diffusion network, each node v_i often has a limited number of parent nodes. The infections of these parent nodes and the node v_i often have relatively great positive correlations. Except for a few nodes whose infections have negative correlations to v_i's infections, the majority of nodes in the network do not have influence relationships to v_i, resulting

Algorithm 1: The TENDS Algorithm

```
   Input  : Node set $V = \{v_1, \dots, v_n\}$, infection status
            results $S = \{S^1, \dots, S^\beta\}$ observed on $V$.
   Output: The diffusion network $G = (V, E)$.
1  $E \leftarrow \emptyset$;                      // set of inferred directed edges
2  for each $v_i \in V$ do
3      for each $v_j \in V (j \neq i)$ do
4          Calculate the infection MI value $IMI(X_i, X_j)$
           using Eq. (25);
5  Partition all non-negative infection MI values into two
   groups by $K$-means (with $K = 2$ and one mean fixed at
   0) and set $\tau$ to the largest value in the group with mean
   close to 0;
6  for each $v_i \in V$ do
7      $P_i \leftarrow \emptyset$;                // $v_i$'s candidate parent node set
8      $C_i \leftarrow \emptyset$;  //$v_i$'s possible parent node combination set
9      $F_i \leftarrow \emptyset$;                // $v_i$'s inferred parent node set
10     for each $v_j \in V (j \neq i)$ do
11         if $IMI(X_i, X_j) > \tau$ then
12             $P_i \leftarrow P_i \cup \{v_j\}$;        // insert $v_j$ into $P_i$
13     for each $W \subseteq P_i$, $|W| \leqslant \log(\phi_W + \delta_i)$ do
14         Calculate $g(v_i, W)$ using Eq. (13);
15         $C_i \leftarrow \{C_i, W\}$;   // add a new element $W$ to $C_i$
16     while $C_i \neq \emptyset$ do
17         $W^* \leftarrow \arg\max_{W \in C_i} g(v_i, W)$;
18         if $|F_i \cup W^*| \leqslant \log(\phi_{F_i \cup W^*} + \delta_i)$ then
19             $F_i \leftarrow F_i \cup W^*$;
20         $C_i \leftarrow C_i \backslash W^*$;
21     $E \leftarrow \{(v_j, v_i) \mid v_j \in F_i\} \cup E$;   // $(v_j, v_i)$ is directed
```

in very small infection MI values (close to 0). These very small infection MI values form a compact cluster with a very small mean (close to 0).

Inspired by this line of reasoning, we introduce a heuristic pruning method based on the infection MI values with the goal of screening out insignificant candidate parent nodes for each node. After calculating the infection MI value for each two nodes in the network, we remove each negative infection MI value. Then, by performing a modified K-means algorithm with $K = 2$ and one of the two means fixed at 0 through all iterations of K-means, we can efficiently partition all non-negative infection MI values into two groups, where one group has a small mean close to 0. Let τ be the largest value in the group with a mean close to 0. Then, for each $IMI(X_i, X_j) \leqslant \tau$, we regard the corresponding node v_j as an insignificant candidate parent node for node v_i and exclude v_j from the candidate parent node set of v_i. This pruning method allows us to screen out insignificant candidate parent nodes, and thus enables the TENDS algorithm to focus on parent node combinations that are more likely to exist in the real diffusion network.

C. Algorithm

Based on the proposed scoring criterion and pruning method, we propose the TENDS algorithm for the problem of reconstructing diffusion network topologies with only the infection status results.

The TENDS algorithm, outlined in Algorithm 1, takes as inputs node set V of the objective diffusion network G and a set S of infection status results observed on V across β diffusion processes. It first initializes the inferred directed edge set E of G as an empty set (line 1), following by calculating the infection MI value for each node pair (lines 2–4), and performing the modified K-means algorithm on all the non-negative infection MI values (with $K = 2$ and one mean fixed at 0 through all K-means iterations) to find an infection MI threshold τ (line 5), which is used to screen out insignificant candidate parent nodes. Then, the TENDS algorithm infers the incoming edges to each node $v_i \in V$ by the following five steps: (1) Firstly, three empty sets P_i, C_i and F_i are initialized to record v_i's candidate parent nodes, possible parent node combinations and inferred parent nodes, respectively (lines 7–9). (2) Secondly, for each node $v_j \in V(j \neq i)$ (line 10), if the corresponding value of $IMI(X_i, X_j)$ is larger than the infection MI threshold τ (line 11), then the node v_j will be inserted into the candidate parent node set P_i of v_i (line 12), otherwise it will be regarded as an insignificant candidate parent node of v_i. (3) Thirdly, for each possible parent node combination $W \subseteq P_i$ that has a size less than $\log(\phi_W + \delta_i)$ (line 13), the corresponding local score $g(v_i, W)$ is calculated and recorded (line 14), and the node combination W is added into possible parent node combination set C_i as a new element (line 15). (4) Fourthly, if the theoretical upper bound for the size of a parent node set is not exceeded (line 18), the inferred parent node set F_i will be continuously expanded with the parent node combination $W^* \in C_i$ that has the currently greatest value of $g(v_i, W)$ ($W \in C_i$) until no more candidate parent node combinations exist (lines 16–20). (5) Finally, a directed edge from each node in F_i to v_i is added to the inferred edge set E of the objective diffusion network G (line 21).

D. Complexity Analysis

The most computationally expensive process in TENDS consists of the following two parts. (1) To disqualify insignificant candidate parent nodes, calculating infection MI values for each node pair requires $O(\beta n^2)$ time, and performing K-means clustering on non-negative infection MI values takes $O(tn^2)$ time, where n is the number of nodes in the network, β is the number of diffusion processes, and t is the number of K-means iterations ($t \ll n$). (2) To find candidate parent node sets, calculating a local score requires $O(\beta\eta)$, where η is the size of the largest possible parent node combination. Calculating all local scores requires $O(\eta^2\kappa^\eta n\beta)$ time, since there are at most $\sum_{i=1}^{\eta} \binom{\kappa}{i} < \eta\kappa^\eta$ candidate parent node combinations for each node, where κ denotes the maximum number of candidate parent nodes for each node ($\eta \leqslant \kappa$). Since most candidate parent nodes are insignificant (discussed

TABLE II
PROPERTIES OF LFR BENCHMARK GRAPHS USED FOR EXPERIMENTS

Graphs	n	$\mathcal{K}$	$\mathcal{T}$
LFR1-5	100,150,200,250,300	4	2
LFR6-10	200	2,3,4,5,6	2
LFR11-15	200	4	1,1.5,2,2.5,3

in Section IV-B), after screening out these nodes with the proposed infection MI-based pruning method, κ is usually much smaller than n, i.e., $\kappa \ll n$.

In summary, the overall time complexity of TENDS is $O(\beta n^2 + tn^2 + \eta^2 \kappa^\eta n \beta)$, where $t \ll n$, $\eta \leqslant \kappa \ll n$. Therefore, the running time of TENDS depends mainly on the network size and the number of diffusion processes.

V. EXPERIMENTAL EVALUATION

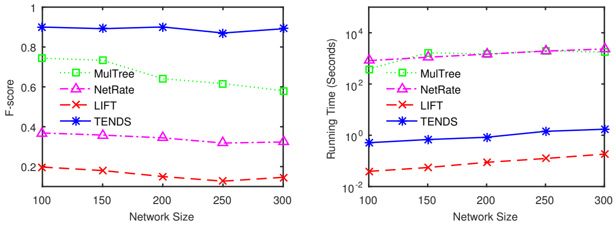
We first introduce the experimental setup, and then report on experiments designed to gain insight into the effectiveness and efficiency of the TENDS algorithm on both synthetic and real-world networks. To this end, we investigate the effects of diffusion network size, the average node degree, the degree dispersion of the diffusion network, the initial infection ratio, the propagation probability, the number of diffusion processes, and the infection MI-based pruning method on the accuracy and running time of TENDS. All algorithms are implemented in Java, running on a desktop PC with an Intel Core i3-6100 CPU at 3.70GHz and 8GB RAM.

A. Experimental Setup

Networks. We adopt the LFR benchmark graphs [31] as the synthetic networks. By using different graph generation parameters, such as the number n of nodes, the average degree $\mathcal{K}$ of each node, and the degree distribution parameter $\mathcal{T}$ (larger $\mathcal{T}$ implies less dispersion of degrees), we generate three series of graphs with properties summarized in Table II. In addition, we adopt two real-world networks, i.e., NetSci [32], which is a coauthorship network containing 379 scientists and 1602 coauthorships, and DUNF [10], which is a microblogging network with 750 users and 2974 following relationships, for the experimental evaluation.

Infection Data. The infection status results S can be obtained by simulating β diffusion processes on each network with randomly selected initially infected nodes in each simulation (the initial infection ratio is α). Corresponding cascades are also recorded for the cascade-based algorithms in the experiments. In each diffusion process, each infected node tries to infect its uninfected child nodes with a given propagation probability, which is subjected to a Gaussian distribution with mean μ and variance 0.05, to ensure that more than 95% of all propagation probabilities are within the range from $\mu - 0.1$ to $\mu + 0.1$.

Performance Criteria. To evaluate the accuracy of the TENDS algorithm on the reconstruction of diffusion network topologies, we report the F-score (i.e., the harmonic mean of precision and recall) of its inferred directed edges, which can be calculated as



(a) *F-score* (b) *Running Time*

Fig. 1. Effect of Diffusion Network Size

$$F\text{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

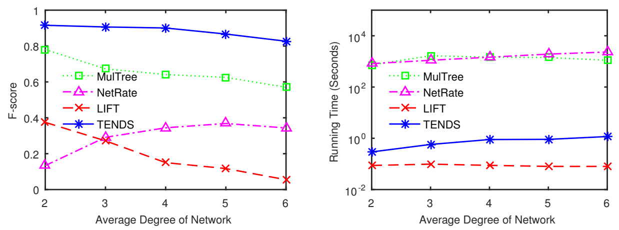
where N_{TP} denotes the number of true positives, i.e., the edges in the real network that are inferred correctly by the algorithm; N_{FP} denotes the number of false positives, i.e., edges that do not exist in the real network, but that are inferred falsely by the algorithm; and N_{FN} denotes the number of false negatives, i.e., edges that exist in the real network, but that are not inferred by the algorithm.

Benchmark Algorithms. Among the existing infection timestamp-based algorithms, embedding-based methods do not infer an explicit diffusion network structure. Therefore, we compare our algorithm with the state-of-the-art convex programming-based approach NetRate [9] and the high performance submodularity-based algorithm MulTree [21]. In addition, as the PATH algorithm [6] requires all path connected node triples, which are difficult to obtain in practice, we choose the infection timestamp-free approach LIFT [7] for comparison. Since NetRate infers the propagation probability between each two nodes in the network, we give NetRate a preferential treatment in accuracy comparisons. Specifically, when calculating the F-scores of edges whose propagation probabilities exceed a threshold, we use different thresholds to find the highest F-score and report it as the accuracy of NetRate. Moreover, since MulTree and LIFT need users to specify the number of edges to be inferred, we provide the real number m of edges in the network to these two algorithms.

B. Effect of Diffusion Network Size

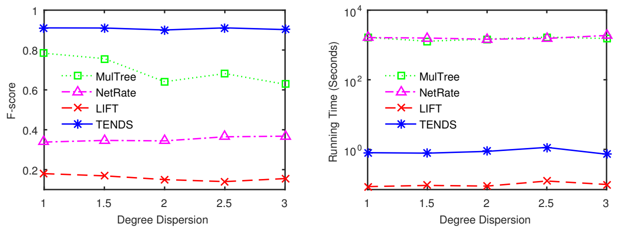
To study the effect of diffusion network size on algorithm performance, we adopt five synthetic networks, i.e., LFR1–5, where the size varies from 100 to 300. We simulate 150 diffusion processes on each network (i.e., $\beta = 150$). In each simulation, $0.15n$ nodes are randomly selected as the initial infected nodes (i.e., $\alpha = 0.15$), and the mean μ of propagation probability is set to 0.3.

Fig. 1 reports the F-score and running time of each algorithm, from which we can observe that (1) a larger diffusion



(a) *F-score* (b) *Running Time*

Fig. 2. Effect of Average Node Degree



(a) *F-score* (b) *Running Time*

Fig. 3. Effect of Node Degree Dispersion of Diffusion Network

network size tends to degrade the accuracy of NetRate, LIFT, and MulTree, while the accuracy of TENDS is reasonably insensitive to the diffusion network size and outperforms the other algorithms. (2) The running time of each algorithm increases with the diffusion network size. LIFT executes the fastest (but with low accuracy), and TENDS is an order of magnitude faster (and has higher accuracy) than both MulTree and NetRate.

C. Effect of Average Node Degree

The edge density of diffusion network can affect the number of influence relationships. The average node degree, i.e., the total number of edges divided by the total number of nodes, is usually used to represent the edge density of a network.

To study the effect of a network's average degree on algorithm performance, we test the algorithms on five synthetic networks, i.e., LFR6–10, where the average degree varies from 2 to 6. We simulate 150 diffusion processes on each network (i.e., $\beta = 150$). In each simulation, $0.15n$ nodes are randomly selected as the initially infected nodes (i.e., $\alpha = 0.15$), and the mean μ of propagation probabilities is set to 0.3.

Fig. 2 illustrates the F-score and running time of each algorithm, from which we can observe that (1) as the average degrees of diffusion networks increase, accuracy of MulTree, TENDS, and LIFT decrease. The accuracy of NetRate increases when the average degree increases from 2 to 5 and then decreases when the average degree reaches 6. Compared with the other tested algorithms, the TENDS algorithm has the best accuracy. (2) The running time of MulTree, NetRate, and TENDS increase with the growth of average degree, and TENDS shows a significant running time advantage over MulTree and NetRate.

D. Effect of Node Degree Dispersion

If a diffusion network has a large degree dispersion, i.e., different nodes have different numbers of edges, then there will be variations in the influence diffusion capabilities of different parts of the network, which can affect the diffusion processes and the final infection statuses of nodes.

To study the effect of the node degree dispersion on algorithm performance, we test the algorithms on five synthetic networks, i.e., LFR11–15, where the degree distribution parameters vary from 1 to 3 (the corresponding standard deviation of the degree varies from about 0.8 to about 0.4). We simulate 150 diffusion processes on each network (i.e., $\beta = 150$). In each of these simulations, $0.15n$ nodes are randomly selected as the initially infected nodes (i.e., $\alpha = 0.15$), and the mean propagation probability μ is set to 0.3.

Fig. 3 reports the F-score and running time of each algorithm, from which we can observe that (1) an increase in the degree distribution parameter tends to reduce the accuracy of MulTree. The accuracy of NetRate, LIFT, and TENDS is reasonably insensitive to degree dispersion, and TENDS performs better than other algorithms. (2) Degree dispersion has little effect on the running times of the algorithms, and TENDS has better running time performance than NetRate and MulTree, and LIFT is the fastest.

E. Effect of Initial Infection Ratio

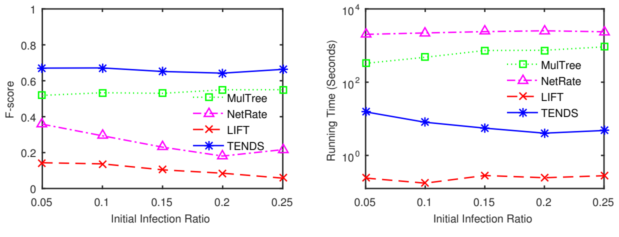
The ratio of initially infected nodes may affect the number of final infected nodes in a diffusion process.

To study the effect of the initial infection ratio on performance, we test the algorithms on real-world networks NetSci and DUNF with different initial infection ratios α (varied from 0.05 to 0.25). For each initial infection ratio, we simulate 150 diffusion processes on each network (i.e., $\beta = 150$) with the mean propagation probability μ fixed at 0.3.

Figs. 4–5 report the F-score and running time of each algorithm on NetSci and DUNF, repectively. From the figures, we can observe that an increase of initial infection ratio tends to improve the accuracy of MulTree, but degrades the accuracy of LIFT and NetRate. TENDS is reasonably insensitive to variations in the initial infection ratio and has the best accuracy. Further, an increase in the initial infection ratio has little effect on the running time of TENDS and LIFT, but results in longer running time for MulTree and NetRate. It is also noted that similar experimental results can be observed on synthetic networks LRF1–15.

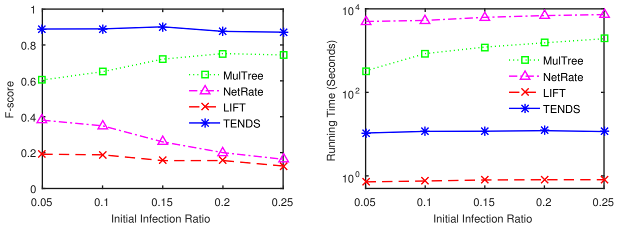
F. Effect of Propagation Probability

The propagation probabilities between nodes may affect the correlations between the infections of parent nodes and corresponding child nodes. Therefore, the propagation probabilities may affect the accuracy of diffusion network reconstruction. Generally, higher propagation probabilities are expected to enhance the correlations between the observed infection statuses of parent nodes and corresponding child nodes, and they



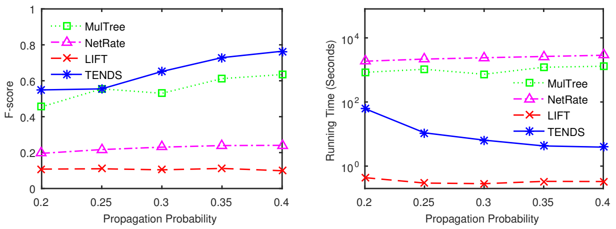
(a) *F-score* (b) *Running Time*

Fig. 4. Effect of Initial Infection Ratio on NetSci



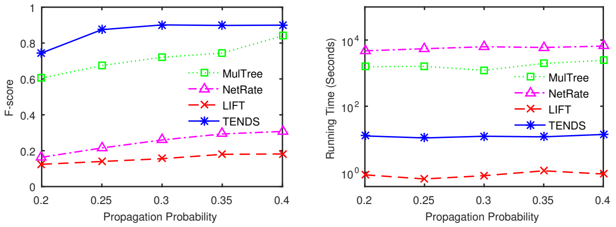
(a) *F-score* (b) *Running Time*

Fig. 5. Effect of Initial Infection Ratio on DUNF



(a) *F-score* (b) *Running Time*

Fig. 6. Effect of Propagation Probability on NetSci



(a) *F-score* (b) *Running Time*

Fig. 7. Effect of Propagation Probability on DUNF

will likely help the algorithms identify influence relationships between nodes more effectively, resulting in an accuracy improvement for the algorithms.

To study the effect of the propagation probability on algorithm performance, we test the algorithms on real-world networks NetSci and DUNF with different propagation probability settings, where we vary the mean propagation probability μ from 0.2 to 0.4. For each propagation probability setting, we simulate 150 diffusion processes on each network (i.e., $\beta = 150$). In each simulation, $0.15n$ nodes are randomly selected as the initial infected nodes (i.e., $\alpha = 0.15$).

Figs. 6–7 report the F-score and running time of each algorithm on NetSci and DUNF, repectively. We can observe that the accuracy of each algorithm increase as the propagation probability increases. Further, TENDS generally achieves the best accuracy, with MulTree being close in one setting. The running times are similar to what is observed in pervious experiments. Similar experimental results can also be observed on synthetic networks LRF1–15.

G. Effect of The Number of Diffusion Processes

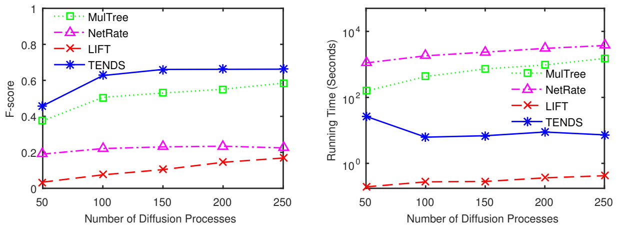
The topology reconstruction of a diffusion network is based on the observed results of diffusion processes. Hence, the number of diffusion processes may affect the accuracy of the reconstructed topology. Generally, more diffusion processes will expose more information about a diffusion network, and this may help diffusion network reconstruction algorithms achieve more accurate inference results.

To study the effect of the amount of diffusion processes on algorithm performance, we test the algorithms on real-world networks NetSci and DUNF with different number β of diffusion processes (β varies from 50 to 250). In each diffusion process, we randomly select $0.15n$ nodes as the initially infected nodes ($\alpha = 0.15$), and the mean propagation probability μ is set to 0.3.

Figs. 8–9 show the F-score and running time of each algorithm on NetSci and DUNF, respectively. We can observe that a larger number of diffusion processes often helps the algorithms achieve more accurate results on network structure inference. TENDS achieves the best accuracy when compared with the other algorithms. To analyze the infection statuses collected from more diffusion processes, the algorithms usually require longer running time, except for that TENDS takes relatively more time when the number of diffusion processes is 50. This is because the more diffusion processes, the more infection data to analyze, resulting in relatively greater computation costs. Nonetheless, an insufficient number of diffusion processes will decrease the statistical significance of the real influence relationships, so that TENDS tends to take into account more candidate parent nodes to find the most probable parent nodes. Compared with MulTree and NetRate, TENDS shows a significant advantage in terms of running time, while LIFT has the lowest running time. Similar experimental results can also be observed on synthetic networks LRF1–15.

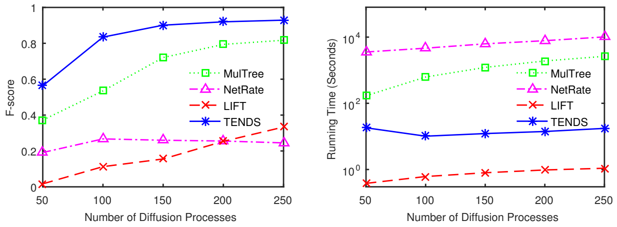
H. Effect of Infection MI-based Pruning Method

To screen out insignificant candidate parent nodes and eliminate redundant computations during the influence relationship inferencing, TENDS adopts an infection MI-based pruning method, which finds an infection MI threshold τ for the identification of insignificant candidate parent nodes.



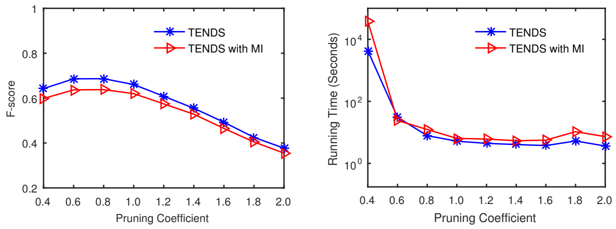
(a) *F-score* (b) *Running Time*

Fig. 8. Effect of Number of Diffusion Processes on NetSci



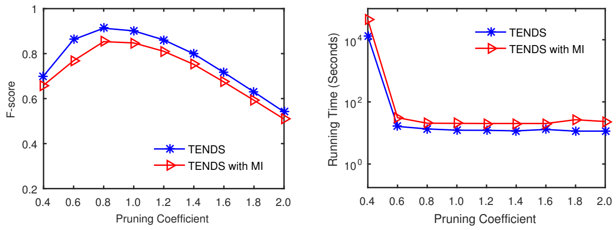
(a) *F-score* (b) *Running Time*

Fig. 9. Effect of Number of Diffusion Processes on DUNF



(a) *F-score* (b) *Running Time*

Fig. 10. Effect of Infection MI-based Pruning Method on NetSci



(a) *F-score* (b) *Running Time*

Fig. 11. Effect of Infection MI-based Pruning Method on DUNF

To study the effect of the infection MI-based pruning method on the performance of TENDS, we test TENDS on real-world networks NetSci and DUNF with different infection MI thresholds. Since the running time of TENDS with a very small infection MI threshold (i.e., TENDS without effective pruning on candidate parent node) on the networks is prohibitively long and beyond acceptable, we omit to report the corresponding performance results. We vary the MI threshold from 0.4τ to 2τ, and for each MI threshold, we simulate 150 diffusion processes on each network (i.e., $\beta = 150$) with $0.15n$ initially infected nodes that are randomly selected in each simulation (i.e., $\alpha = 0.15$) and the mean propagation probability μ fixed at 0.3.

Figs. 10–11 report the F-score and running time of TENDS with different infection MI thresholds on NetSci and DUNF, respectively. We can observe that the infection MI threshold τ found by the infection MI-based pruning method is able to help TENDS achieve a nearly optimal accuracy. When the infection MI threshold is less than 0.6τ, the smaller MI threshold the lower accuracy of TENDS. When the MI threshold is more than τ, the larger infection MI threshold the lower accuracy of TENDS. This is because a smaller infection MI threshold has a weaker effect of pruning, and thus leaves more insignificant candidate parent nodes for parent node selection, causing precision degradation for TENDS; in contrast, if the infection MI threshold is too large, it may screen out the real candidate parent nodes, resulting in a lower recall for TENDS. Further, compared with using a small infection MI threshold less than 0.6τ, using the infection MI threshold τ found by the infection MI-based pruning method markedly reduces the running time of TENDS. Similar experimental results can also be observed on synthetic networks LRF1–15.

In addition, the infection MI measurement is modified from traditional MI metric to help better reflect the positive correlations of node infections. To verify the effectiveness of this modification, we also execute TENDS with the traditional MI instead of the infection MI, and report the corresponding performance in Figs. 10–11. From the figures, we can observe that compared with using traditional MI, using infection MI enables our approach to achieve a reasonably better accuracy and slightly higher efficiency. The reason behind is that infection MI can distinguish positive and negative correlations of node infections, while traditional MI mixes up these two types of correlations. Therefore, using infection MI will find each node relatively less candidate parent nodes, which are more reasonable.

VI. Conclusion

In this paper, we have investigated the problem of how to reconstruct the topology of a diffusion network based only on final node infection statuses observed from a set of diffusion processes. To this end, we have designed a decomposable scoring criterion, which balances the likelihood of inferred topology and statistical error, and transforms the problem of diffusion network reconstruction into finding for each node in the network a set of most probable parent nodes. Furthermore, we have presented a heuristic pruning method to eliminate redundant computations during the search of most probable parent nodes. Extensive experiments on both synthetic and real-world networks offer evidence that the proposed approach is effective and efficient.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFB1004003), the NSFC Grants (61976163, 61972338 and U1609217), the Advance Research Projects of Civil Aerospace Technology ("Intelligent Distribution Technology for Domestic Satellite Information"), the Technological Innovation Major Projects of Hubei Province (2017AAA125), the Natural Science Foundation of Hubei Province (ZRMS2020000714), the Natural Science Foundation of Zhejiang Province (LQ19F020001), the Science and Technology Program of Wuhan City (2018010401011288), the ZJU-Hikvision Joint Project, and the Xiaomi-WHU AI Lab. Hao Huang is the corresponding author of the work.

References

[1] S. Gao, H. Pang, P. Gallinari, J. Guo, and N. Kato, "A novel embedding method for information diffusion prediction in social network big data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2097–2105, 2017.

[2] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu, "HawkesTopic: A joint model for network inference and topic modeling from text-based cascades," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015, pp. 871–880.

[3] J. Wallinga and P. Teunis, "Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures," *American Journal of Epidemiology*, vol. 160, no. 6, pp. 509–516, 2004.

[4] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web*, vol. 1, no. 1, p. 5, 2007.

[5] Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen, "CSI: Community-level social influence analysis," in *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2013)*, 2013, pp. 48–63.

[6] V. Gripon and M. Rabbat, "Reconstructing a graph from path traces," in *Proceedings of 2013 IEEE International Symposium on Information Theory(ISIT 2013)*, 2013, pp. 2488–2492.

[7] K. Amin, H. Heidari, and M. Kearns, "Learning from contagion(without timestamps)," in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, 2014, pp. 1845–1853.

[8] S. Myers and J. Leskovec, "On the convexity of latent social network inference," in *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 2010, pp. 1741–1749.

[9] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, 2011, pp. 561–568.

[10] S. Wang, X. Hu, P. Yu, and Z. Li, "MMRate: Inferring multi-aspect diffusion networks with multi-pattern cascades," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, 2014, pp. 1246–1255.

[11] Y. Rong, Q. Zhu, and H. Cheng, "A model-free approach to infer the diffusion network from event cascade," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016)*, 2016, pp. 1653–1662.

[12] N. Du, L. Song, A. Smola, and M. Yuan, "Learning networks of heterogeneous influence," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012, pp. 2780–2788.

[13] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Structure and dynamics of information pathways in online media," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM 2013)*, 2013, pp. 23–32.

[14] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf, "Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm," in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, 2014, pp. 793–801.

[15] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," in *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, 2013, pp. 666–674.

[16] J. Pouget-Abadie and T. Horel, "Inferring graphs from cascades: A sparse recovery framework," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015, pp. 977–986.

[17] P. Netrapalli and S. Sanghavi, "Learning the graph of epidemic cascades," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2012)*, 2012, pp. 211–222.

[18] H. Narasimhan, D. C. Parkes, and Y. Singer, "Learnability of influence in networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015, pp. 3186–3194.

[19] D. Kalimeris, Y. Singer, K. Subbian, and U. Weinsberg, "Learning diffusion using hyperparameters," in *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018, pp. 2420–2428.

[20] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*, 2010, pp. 1019–1028.

[21] M. Gomez-Rodriguez and B. Schölkopf, "Submodular inference of diffusion networks from multiple trees," in *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012, pp. 489–496.

[22] T. Kurashima, T. Iwata, N. Takaya, and H. Sawada, "Probabilistic latent network visualization: Inferring and embedding diffusion networks," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, 2014, pp. 1236–1245.

[23] S. Bourigault, C. Lagnier, S. Lamprier, L. Denoyer, and P. Gallinari, "Learning social network embeddings for predicting information diffusion," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining (WSDM 2014)*, 2014, pp. 393–402.

[24] B. Abrahao, F. Chierichetti, and R. Kleinberg, "Trace complexity of network inference," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*, 2013, pp. 491–499.

[25] E. Sefer and C. Kingsford, "Convex risk minimization to infer networks from probabilistic diffusion data at multiple scales," in *Proceedings of the 31st IEEE International Conference on Data Engineering (ICDE 2015)*, 2015, pp. 663–674.

[26] X. He, K. Xu, D. Kempe, and Y. Liu, "Learning influence functions from incomplete observations," in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016, pp. 2065–2073.

[27] A. Lokhov, "Reconstructing parameters of spreading models from partial observations," in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016, pp. 3467–3475.

[28] Q. Yan, H. Huang, Y. Gao, W. Lu, and Q. He, "Group-level influence maximization with budget constraint," in *Proceedings of the 22nd International Conference on Database Systems for Advanced Applications (DASFAA 2017)*, 2017, pp. 625–641.

[29] R. Nishii, "Maximum likelihood principle and model selection when the true model is unspecified," *Journal of Multivariate Analysis*, vol. 27, no. 2, pp. 392–403, 1988.

[30] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *SIGMOD 2014*, 2014, pp. 75–86.

[31] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, 2008.

[32] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, 2006.