Diffusion Network Inference from Partial Observations

Ting Gan[1], **Keqi Han**[1], **Hao Huang**[1], **Shi Ying**[1], **Yunjun Gao**[2], **Zongpeng Li**[1]
[1]School of Computer Science, Wuhan University, China
[2]College of Computer Science and Technology, Zhejiang University, China
{ganting, hankeqi, haohuang, yingshi, zongpeng}@whu.edu.cn, gaoyj@zju.edu.cn

Abstract

To infer the structure of a diffusion network from observed diffusion results, existing approaches customarily assume that observed data are complete and contain the final infection status of each node, as well as precise timestamps of node infections. Due to high cost and uncertainties in the monitoring of node infections, exact timestamps are often unavailable in practice, and even the final infection statuses of nodes are sometimes missing. In this work, we study how to carry out diffusion network inference without infection timestamps, using only partial observations of the final infection statuses of nodes. To this end, we iteratively infer the structure of the target diffusion network with observed data and imputed values for missing data, and learn the most likely infection transmission probabilities between nodes w.r.t. current inferred structure, which then help us update the imputation of missing data in turn. Extensive experimental results on both synthetic and real-world networks show that our approach can properly handle missing data and accurately uncover diffusion network structures.

Introduction

The structures of diffusion networks delineate the underlying influence relationships between the nodes. An explicit diffusion network structure is essential for understanding the mechanisms of historical diffusion processes and for developing strategies to control future diffusions on the network. Unfortunately, in most cases, the diffusion network structures are not naturally accessible and need to be inferred from diffusion results observed from history.

To infer the structure of a diffusion network, most existing approaches resort to precise timestamps of historical node infections, and utilize sequences of timestamps (known as cascades) to determine potential parent-child influence relationships between nodes (Mehmood et al. 2013). By transforming the problem of diffusion network inference into a problem of convex optimization (Myers and Leskovec 2010; Gomez-Rodriguez, Balduzzi, and Schölkopf 2011; Du et al. 2012; Gomez-Rodriguez, Leskovec, and Schölkopf 2013a,b; Daneshmand et al. 2014; Wang et al. 2014; Pouget-Abadie and Horel 2015; Narasimhan, Parkes, and Singer 2015; Rong, Zhu, and Cheng 2016; Kalimeris et al. 2018) or submodular optimization (Gomez-Rodriguez, Leskovec, and Krause 2010; Gomez-Rodriguez and Schölkopf 2012), they try to find a diffusion network structure that maximizes the likelihood of given cascades by solving the corresponding optimization problems.

In reality, gathering node infection timestamps as cascades is not always feasible or affordable. In most real-world settings, such as disease propagation, nodes have a wide spatial distribution, and infection monitoring is often labor/resource demanding. Given the limitation of the cascade-based approaches, some infection timestamp-free methods try to learn diffusion network structures from only the final infection statuses of the nodes in historical diffusion processes (Gripon and Rabbat 2013; Amin, Heidari, and Kearns 2014; Huang et al. 2019a,b; Han et al. 2020), since observing the final infection statuses is relatively easier and less expensive than monitoring exact infection timestamps.

Whether the existing approaches use observed cascades or final infection statuses of nodes, they generally assume that the observed data are complete, without missing values on any infection timestamp or status. Unfortunately, this assumption often fails to hold in real-world applications (He et al. 2016; Lokhov 2016). For example, it is virtually impossible to avoid missing readings from non-functioning sensors, or avoid non-response in a survey. In situations where missing data are unavoidable, one can make the data complete via certain ad-hoc strategies, such as filling in the missing parts with average values in the observed data (Sefer and Kingsford 2015), although these strategies often have no guarantee of accuracy and may result in severe biases. So far, a few studies do deal with missing data, but aim at predicting diffusion results (He et al. 2016) or learning only the strengths of influence relationships (Lokhov 2016; Wilinski and Lokhov 2020), under the assumption that the diffusion network structures are known in advance.

In this work, we investigate how to infer the structure of a diffusion network based on only partial observations of the final node infection statuses in a limited number of historical diffusion processes. We propose an effective algorithm called POIND (a re-ordered acronym of **D**iffusion **N**etwork **I**nference with **P**artial **O**bservations) for this problem. POIND consists of two iterative steps, i.e., (1) inferring the structure of the objective network with observed data and imputed values for missing data, where the imputed values

are sampled from estimated probability distributions for the missing data; and (2) learning the most likely infection transmission probabilities between nodes and their parents w.r.t. current inferred structure, which will help POIND update the probability distribution estimation and then the value imputation of missing data in turn.

In summary, our key contributions include the following: (1) To our knowledge, this work is the first to explicitly solve the problem of diffusion network inference with partial observations of final node infection statuses. (2) We propose an effective approach to infer the structure of objective diffusion network and learn the infection transmission probabilities in an iterative way. (3) We present a probabilistic sampling method for the imputation of missing data, which is able to complete the data properly and improve the accuracy of structure inference.

The remainder of the paper is organized as follows. We first present our problem statement, and then introduce our proposed POIND algorithm, followed by reporting experimental results and our findings before concluding the paper.

Problem Statement

A diffusion network can be represented by a directed graph $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ is the set of nodes in the network, and E represents the set of parent-child influence relationships between the nodes. In a diffusion process, infections propagate from an infected parent node to each uninfected child node with a certain probability. Let F_i denote the set of parent nodes of node $v_i \in V$ w.r.t. E, as each node has two possible infection statuses (i.e., infected and uninfected), there are $2^{|F_i|}$ possible combinations of the infection statuses of v_i's parent nodes, where $|F_i|$ refers to the number of nodes in set F_i. Let X_i and X_{F_i} be the infection status variable of node v_i and the infection status variables of nodes in F_i, respectively, $\pi_{i,j}$ be the j-th possible combination of the infection statuses of v_i's parent nodes ($j \in \{1, ..., 2^{|F_i|}\}$), and $\theta_{i,j,k}$ be the probability of v_i's infection status being k (i.e., $X_i = k$, where $k \in \{0, 1\}$, 0 refers to uninfected status, and 1 refers to infected status) under the condition that the infection statuses of v_i's parent nodes are instantiated to the j-th possible combination (i.e., $X_{F_i} = \pi_{i,j}$), we refer to the probabilities $\{\theta_{i,j,1} \mid j \in \{1, ..., 2^{|F_i|}\}\}$ as the infection transmission probabilities between F_i and v_i ($\theta_{i,j,1} = 1 - \theta_{i,j,0}, \forall i, j$).

In the problem of diffusion network inference, the node set V is given, while the structure (i.e., the directed edge set E) of the objective diffusion network is unknown, so are the infection transmission probabilities w.r.t. the structure. To infer the structure of a diffusion network, a set S of diffusion results observed from a number of historical diffusion processes on the network is required. In this work, we deal with the cases that the diffusion results contain only the final infection statuses of nodes in each diffusion process, i.e., $S = \{s_i^\ell \mid i \in \{1, ..., n\}, \ell \in \{1, ..., \beta\}\}$, where $s_i^\ell \in \{0, 1\}$ is the infection status of node $v_i \in V$ observed at the end of the ℓ-th diffusion process , and β is the number of historical diffusion processes. Furthermore, we aim to carry out diffusion network inference in a complex yet more realistic situation, i.e., partial final infection statuses in the diffusion results are unobserved. Let S^{obs} and S^{mis} denote the observed part and missing part in S, respectively, then our problem statement can be formulated as follows.

Given: partial observations S^{obs} of node infection statuses observed on a diffusion network G at the end of β historical diffusion processes.

Infer: the structure of the diffusion network G.

The POIND Algorithm

To infer the structure of G with partial observations S^{obs}, POIND performs the following two steps iteratively, namely (1) inferring the structure with S^{obs} and imputed values for S^{mis}, and (2) learning the most likely infection transmission probabilities w.r.t. the inferred structure.

In the first step, to make the data complete for structure inference, POIND initializes the imputed values for S^{mis} randomly (or with some ad-hoc methods) at the first iteration; in subsequent iterations, it estimates the probability distribution of each unobserved infection status in S^{mis} based on S^{obs}, the latest inferred structure, and the current values of infection transmission probabilities $\Theta = \{\theta_{i,j,1} \mid i \in \{1, ..., n\}, j \in \{1, ..., 2^{|F_i|}\}\}$, and then carries out the value imputation for missing data by sampling values from an estimated probability distribution of each unobserved infection status. With the completed data, POIND can adopt existing infection timestamp-free methods for diffusion network inference to update the result of structure inference.

In the second step, given the latest inferred structure, POIND performs an EM-like approach to approximate the optimal values of infection transmission probabilities Θ w.r.t. this structure in an iterative way. The EM-like approach adopts a carefully designed expectation function, from which the optimal updating result for current values of Θ can be theoretically deduced. EM-like approach iteratively updates the values of Θ until convergence.

The details of the above two steps, as well as a complexity analysis for POIND are provided in what follows.

Structure Inference

Let $E^{(T)}$ and $\Theta^{(T)}$ denote the directed edge set and infection transmission probabilities inferred by POIND after the T-th iteration, then in the next iteration, the goal of structure inference is to find a directed edge set $E^{(T+1)}$ that satisfies the following requirement.

$$E^{(T+1)} = \arg\max_{E} P(E \mid \Theta^{(T)}, S^{obs}). \tag{1}$$

Nonetheless, as S^{obs} is not a complete data set, it is difficult to calculate $P(E \mid \Theta^{(T)}, S^{obs})$ directly. Aiming at an easier estimation for $P(E \mid \Theta^{(T)}, S^{obs})$, we repeatedly sample S^{mis} under the condition of S^{obs}, $E^{(T)}$ and $\Theta^{(T)}$ to obtain a set of samples of missing data $\{\hat{S}_1^{mis}, ..., \hat{S}_m^{mis}\}$, where m is the number of sampling rounds. Then, we can obtain a set of samples of complete data $\{\hat{S}_1, ..., \hat{S}_m\}$, in which the r-th sample $\hat{S}_r$ consists of S^{obs} and $\hat{S}_r^{mis}$, i.e.,

$$\hat{S}_r = (S^{obs}, \hat{S}_r^{mis}). \tag{2}$$

If the sampling is sufficient (i.e., m is great enough), then

$$P(E \mid \Theta^{(T)}, S^{obs}) \simeq P(E \mid \hat{S}_1, ..., \hat{S}_m). \tag{3}$$

Given this, we can utilize $P(E \mid \hat{S}_1, ..., \hat{S}_m)$ to estimate $P(E \mid \Theta^{(T)}, S^{obs})$, and reformulate Eq. (1) as follows.

$$E^{(T+1)} = \arg\max_{E} P(E \mid \hat{S}_1, ..., \hat{S}_m). \tag{4}$$

Note that the problem in Eq. (4) is equivalent to learning diffusion network structures from only the complete data of final node infection statuses. One can utilize existing approaches to this problem, such as TWIND (Huang et al. 2019b), to infer $E^{(T+1)}$. Therefore, how to complete the data through sampling is crucial for structure inference. Next we elaborate the sampling method.

The Sampling Method To sample S^{mis} in $(T+1)$-th iteration, we need to know the conditional probability distribution $P(s_i^\ell \mid S^{obs}, E^{(T)}, \Theta^{(T)})$ for each unobserved infection status $s_i^\ell \in S^{mis}$. Let $D_\ell^{obs} = \{s_i^\ell \in S^{obs} \mid i \in \{1, ..., n\}\}$ denote the set of observed data from ℓ-th historical diffusion process ($\ell \in \{1, ..., \beta\}$), as each historical diffusion process is independent of each other, we have relationship

$$P(s_i^\ell \mid S^{obs}, E^{(T)}, \Theta^{(T)}) = P(s_i^\ell \mid D_\ell^{obs}, E^{(T)}, \Theta^{(T)}). \tag{5}$$

Without loss of generality, let $D = \{s_1, ..., s_n\}$ represent the final infection statuses of nodes in a historical diffusion process, D^{obs} and D^{mis} represent the observed part and missing part in D, respectively. For the sake of simplicity, we adopt E and Θ to represent the latest inferred directed edge set and infection transmission probabilities. The infection probability p_i of node v_i in this historical diffusion process can be estimated as follows.

$$p_i = \begin{cases} 0, & s_i \in D^{obs}, s_i = 0 \\ 1, & s_i \in D^{obs}, s_i = 1 \\ P(X_i = 1 \mid D^{obs}, E, \Theta), & s_i \in D^{mis} \end{cases} \tag{6}$$

When the final infection status s_i of node v_i is missing (i.e., $s_i \in D^{mis}$), we consider the following two cases.

Case 1: If for each of v_i's parent nodes, its infection probability is known already, the infection probability p_i of node v_i can be estimated as follows.

$$\begin{aligned} p_i &= \sum_{j=1}^{2^{|F_i|}} P(X_i = 1, X_{F_i} = \pi_{i,j} \mid D^{obs}, E, \Theta) \\ &= \sum_{j=1}^{2^{|F_i|}} \left(\prod_{\alpha=1}^{|F_i|} \varphi\Big(F_i(\alpha), j\Big) \right) \theta_{i,j,1}, \end{aligned} \tag{7}$$

where $F_i(\alpha)$ refers to the index of the α-th node in F_i and

$$\varphi\Big(F_i(\alpha), j\Big) = \begin{cases} p_{F_i(\alpha)}, & X_{F_i} = \pi_{i,j}, X_{F_i(\alpha)} = 1; \\ 1 - p_{F_i(\alpha)}, & X_{F_i} = \pi_{i,j}, X_{F_i(\alpha)} = 0. \end{cases} \tag{8}$$

Moreover, we would like to point out that the calculation of p_i can be carried out in a more efficient way. Towards this, we divide the set F_i into two parts, namely (1) the observed part F_i^{obs} in which the final infection status of each node is observed, and (2) the missing part F_i^{mis} in which the final infection status of each node is unobserved. Then, Eq. (7) can be rewritten as follows.

$$\begin{aligned} p_i &= \sum_{j=1}^{2^{|F_i|}} \prod_{\alpha=1}^{|F_i|} \varphi\Big(F_i(\alpha), j\Big) \theta_{i,j,1} \\ &= \sum_{j_1=1}^{2^{|F_i^{obs}|}} \sum_{j_2=1}^{2^{|F_i^{mis}|}} \prod_{\alpha_1=1}^{|F_i^{obs}|} \prod_{\alpha_2=1}^{|F_i^{mis}|} \theta_{i,\overline{j_1 j_2},1} \\ &\quad \times \varphi\Big(F_i^{obs}(\alpha_1), j_1\Big) \times \varphi\Big(F_i^{mis}(\alpha_2), j_2\Big), \end{aligned} \tag{9}$$

where $\overline{j_1 j_2} \in \{1, ..., 2^{|F_i|}\}$, and the $\overline{j_1 j_2}$-th possible combination of the infection statues of nodes in F_i corresponds to the j_1-th possible combination of the infection statues of nodes in F_i^{obs} combined with the j_2-th possible combination of the infection statues of nodes in F_i^{mis}.

For the α_1-th node in F_i^{obs}, the value of $p_{F_i^{obs}(\alpha_1)}$ and the value of $\varphi\Big(F_i^{obs}(\alpha_1), j_1\Big)$ for each $j_1 \in \{1, ..., 2^{|F_i^{obs}|}\}$ should be 0 or 1. Without loss of generality, assuming that the infection statuses of nodes in F_i^{obs} are instantiated to the j_i-th possible combination ($j_i \in \{1, ..., 2^{|F_i^{obs}|}\}$), then for each $j_2 \in \{1, ..., 2^{|F_i^{mis}|}\}$, the following relationship holds.

$$\begin{aligned} &\sum_{j_1=1}^{2^{|F_i^{obs}|}} \prod_{\alpha_1=1}^{|F_i^{obs}|} \prod_{\alpha_2=1}^{|F_i^{mis}|} \theta_{i,\overline{j_1 j_2},1} \\ &\quad \times \varphi\Big(F_i^{obs}(\alpha_1), j_1\Big) \times \varphi\Big(F_i^{mis}(\alpha_2), j_2\Big) \\ &= \prod_{\alpha_2=1}^{|F_i^{mis}|} \varphi\Big(F_i^{mis}(\alpha_2), j_2\Big) \theta_{i,\overline{j_i j_2},1}. \end{aligned} \tag{10}$$

Combining Eqs. (9) & (10), we have a much simpler calculating formula for p_i as follows.

$$p_i = \sum_{j=1}^{2^{|F_i^{mis}|}} \prod_{\alpha=1}^{|F_i^{mis}|} \varphi\Big(F_i^{mis}(\alpha), j\Big) \theta_{i,\overline{j_i j},1}. \tag{11}$$

When we start the infection probability estimation, for each node with an observed infection status, we update its infection probability by Eq. (6), and then repeat the following two steps, i.e., (1) checking which nodes satisfy the computation condition of Case 1, and (2) calculating the infection probabilities of these nodes by Eq. (11), until there is no node satisfying the computation condition of Case 1. If there are still a few nodes whose infection probabilities are unknown, it indicates that there are cyclic dependencies among their infection probability estimation. We refer to this kind of situation as Case 2.

Case 2: Without loss of generality, let $v_1, ..., v_r \in V$ be the remaining nodes that cannot satisfy the computation condition of Case 1. Then, each remaining node $v_q \in \{v_1, ..., v_r\}$ will have at least one parent node with unknown

infection probability. Hence, we cannot calculate each p_q with Eq. (11) separately. Nevertheless, we can compute infection probabilities jointly for all the remaining nodes by solving the following equations.

$$(p_1, ..., p_r) = f(p_1, ..., p_r). \tag{12}$$

where

$$f(p_1, ..., p_r) = \big(f_1(p_1, ..., p_r), ..., f_r(p_1, ..., p_r)\big), \tag{13}$$

$$f_q(p_1, ..., p_r) = \sum_{j=1}^{2^{|F_q^{mis}|}} \prod_{\alpha=1}^{|F_q^{mis}|} \varphi\Big(F_q^{mis}(\alpha), j\Big)\theta_{q,\overline{j_q j},1}. \tag{14}$$

The above Eq. (12) is a polynomial system of equations w.r.t. $p_1, ..., p_r$, of which the numerical solution can be efficiently obtained by existing tools, such as the `fsolve` and `root` functions in the SciPy package for Python.

When we know all the infection probabilities, we can carry out sampling for each $s_i \in D^{mis}$ based on p_i to complete the data of final infection statuses in historical diffusion processes, and then perform TWIND on the completed data to infer the structure of the objective network G.

Learning Infection Transmission Probabilities

After inferring $E^{(T+1)}$ based on $\Theta^{(T)}$ and S^{obs}, we can learn $\Theta^{(T+1)}$ based on $E^{(T+1)}$ and S^{obs} by solving the following problem.

$$\Theta^{(T+1)} = \arg\max_{\Theta} P(\Theta \mid E^{(T+1)}, S^{obs}). \tag{15}$$

With a given directed edge set $E^{(T+1)}$, the parent node set F_i for each node $v_i \in V$ and the possible combinations of v_i's parent nodes are fixed, so that the elements in $\Theta^{(T+1)}$ are also fixed. What we can do is to find an optimal value for each element $\theta_{i,j,1} \in \Theta$ such that the likelihood of S^{obs} is maximized. Let $H(E^{(T+1)})$ denote the constrained search space of $\Theta^{(T+1)}$ w.r.t. the given $E^{(T+1)}$, the problem of learning $\Theta^{(T+1)}$ can be reformulated as

$$\Theta^{(T+1)} = \arg\max_{\Theta \in H(E^{(T+1)})} P(S^{obs} \mid \Theta). \tag{16}$$

Inspired by the EM algorithm, we propose to update Θ iteratively, and define an expectation function Q as follows.

$$\begin{aligned} &Q(\Theta, \Theta^{[t]}) \\ =&E[\log P(S^{obs}, S^{mis} \mid \Theta) \mid S^{obs}, \Theta^{[t]}] \\ =&\sum_{S^{mis}} P(S^{mis} \mid S^{obs}, \Theta^{[t]}) \log P(S^{obs}, S^{mis} \mid \Theta). \end{aligned} \tag{17}$$

where $\Theta^{[t]}$ refers to the updated Θ after t iterations, and $E[\cdot]$ refers to the expectation value of a variable.

Function Q has a nice theoretical property as follows.

Theorem 1. *If $Q(\Theta, \Theta^{[t]}) > Q(\Theta^{[t]}, \Theta^{[t]})$, then relationship $P(S^{obs} \mid \Theta) > P(S^{obs} \mid \Theta^{[t]})$ holds.*

Proof: The relationship

$$\begin{aligned} &\log P(S^{obs} \mid \Theta) - \log P(S^{obs} \mid \Theta^{[t]}) \\ =&\log \frac{\sum_{S^{mis}} P(S^{obs}, S^{mis} \mid \Theta)}{P(S^{obs} \mid \Theta^{[t]})} \frac{P(S^{mis} \mid S^{obs}, \Theta^{[t]})}{P(S^{mis} \mid S^{obs}, \Theta^{[t]})} \\ =&\log \sum_{S^{mis}} P(S^{mis} \mid S^{obs}, \Theta^{[t]}) \frac{P(S^{obs}, S^{mis} \mid \Theta)}{P(S^{obs}, S^{mis} \mid \Theta^{[t]})} \\ \geq&\sum_{S^{mis}} P(S^{mis} \mid S^{obs}, \Theta^{[t]}) \log \frac{P(S^{obs}, S^{mis} \mid \Theta)}{P(S^{obs}, S^{mis} \mid \Theta^{[t]})} \\ =&Q(\Theta, \Theta^{[t]}) - Q(\Theta^{[t]}, \Theta^{[t]}) \end{aligned}$$

holds, where the inequality is derived from Jensens inequality. Thus, the theorem is correct. ■

According to the above Theorem 1, one can learn the optimal $\Theta^{[t+1]}$ by finding a Θ from $H(E^{(T+1)})$ such that the value of $Q(\Theta, \Theta^{[t]})$ is maximized, i.e.,

$$\Theta^{[t+1]} = \arg\max_{\Theta \in H(E^{(T+1)})} Q(\Theta, \Theta^{[t]}). \tag{18}$$

According to Eq. (3) in the work of Han et al. (2020), the probability $P(S^{obs}, S^{mis} \mid \Theta)$ can be reformulated as

$$\begin{aligned} &P(S^{obs}, S^{mis} \mid \Theta) \\ =&\prod_{i=1}^{n} \prod_{j=1}^{2^{|F_i|}} \theta_{i,j,0}^{N_{i,j,0}} \cdot \theta_{i,j,1}^{N_{i,j,1}} \\ =&\prod_{i=1}^{n} \prod_{j=1}^{2^{|F_i|}} \theta_{i,j,0}^{N_{i,j,0}} \cdot (1 - \theta_{i,j,0})^{N_{i,j,1}}. \end{aligned} \tag{19}$$

where $N_{i,j,0}$ and $N_{i,j,1}$ represent the number of times situations $X_i = 0 \wedge X_{F_i} = \pi_{i,j}$ and $X_i = 1 \wedge X_{F_i} = \pi_{i,j}$ appear in (S^{obs}, S^{mis}), respectively.

Then, the expectation function Q can be reformulated as

$$\begin{aligned} &Q(\Theta, \Theta^{[t]}) \\ =&\sum_{S^{mis}} \left(\begin{array}{l} P(S^{mis} \mid S^{obs}, \Theta^{[t]}) \times \\ \sum_{i=1}^{n} \sum_{j=1}^{2^{|F_i|}} \left(\begin{array}{l} N_{i,j,0} \log \theta_{i,j,0} + \\ N_{i,j,1} \log(1 - \theta_{i,j,0}) \end{array} \right) \end{array} \right) \\ =&\sum_{S^{mis}} \sum_{i=1}^{n} \sum_{j=1}^{2^{|F_i|}} \left(\begin{array}{l} a_{ij} \log \theta_{i,j,0} + \\ b_{ij} \log(1 - \theta_{i,j,0}) \end{array} \right) \\ =&\sum_{i=1}^{n} \sum_{j=1}^{2^{|F_i|}} h_{ij}(\theta_{i,j,0}), \end{aligned} \tag{20}$$

where

$$\begin{aligned} h_{ij}(\theta_{i,j,0}) &= \left(\begin{array}{l} \sum_{S^{mis}} a_{ij} \log \theta_{i,j,0} + \\ \sum_{S^{mis}} b_{ij} \log(1 - \theta_{i,j,0}) \end{array} \right), \\ a_{ij} &= N_{i,j,0} P(S^{mis} \mid S^{obs}, \Theta^{[t]}), \\ b_{ij} &= N_{i,j,1} P(S^{mis} \mid S^{obs}, \Theta^{[t]}). \end{aligned} \tag{21}$$

The following Theorem can help us find an optimal Θ that maximizes $Q(\Theta, \Theta^{[t]})$.

Theorem 2. *Let $0 < \theta < 1$, $a > 0$ and $b > 0$, then function $f(\theta) = a\log\theta + b\log(1-\theta)$ reaches its maximum value at $\theta = \frac{a}{a+b}$.*

Proof: The derivative of $f(\theta)$ is $f'(\theta) = \frac{a}{\theta} - \frac{b}{1-\theta}$, based on which we have that for $0 < \theta < \frac{a}{a+b}$, relationship $f'(\theta) > 0$ holds, and for $\frac{a}{a+b} < \theta < 1$, relationship $f'(\theta) < 0$ holds. Hence, $f(\theta)$ reaches its maximum value at $\theta = \frac{a}{a+b}$, and the theorem is correct. ■

To maximize $Q(\Theta, \Theta^{[t]})$, we need to maximize the value of each $h_{ij}(\theta_{i,j,0})$. According to Theorem 2, for each $i \in \{1, ..., n\}$ and $j \in \{1, ..., 2^{|F_i|}\}$, function $h_{ij}(\theta_{i,j,0})$ reaches its maximum value at $\sum_{S^{mis}} a_{ij} / (\sum_{S^{mis}} a_{ij} + \sum_{S^{mis}} b_{ij})$. In other words, the optimal updating result $\theta_{i,j,0}^{[t+1]}$ for $\theta_{i,j,0}^{[t]}$ should be

$$\theta_{i,j,0}^{[t+1]} = \frac{\sum_{S^{mis}} a_{ij}}{\sum_{S^{mis}} a_{ij} + \sum_{S^{mis}} b_{ij}}, \tag{22}$$

and $\theta_{i,j,1}^{[t+1]} = 1 - \theta_{i,j,0}^{[t+1]}$.

In order to obtain the value of $\theta_{i,j,1}^{[t+1]}$, we need to calculate $\sum_{S^{mis}} N_{i,j,k} P(S^{mis} \mid S^{obs}, \Theta^{[t]})$ for $k = 0, 1$.

$$\begin{aligned} &\sum_{S^{mis}} N_{i,j,k} P(S^{mis} \mid S^{obs}, \Theta^{[t]}) \\ =&E[N_{i,j,k} \mid S^{obs}, \Theta^{[t]}] \\ =&\sum_{\ell=1}^{\beta} E[N_{i,j,k} \mid D_\ell^{obs}, \Theta^{[t]}]. \end{aligned} \tag{23}$$

If the infection statuses of v_i and its parent nodes can be observed in the ℓ-th historical diffusion process, then $E[N_{i,j,k} \mid D_\ell^{obs}, \Theta^{[t]}]$ can be calculated as

$$\begin{aligned} &E[N_{i,j,k} \mid D_\ell^{obs}, \Theta^{[t]}] \\ =&\begin{cases} 1, & s_i = k \text{ and } \pi_{i,j} \in D_\ell^{obs}; \\ 0, & s_i \neq k \text{ or } \pi_{i,j} \notin D_\ell^{obs}. \end{cases} \end{aligned} \tag{24}$$

Otherwise, $E[N_{i,j,k} \mid D_\ell^{obs}, \Theta^{[t]}]$ can be estimated as

$$\begin{aligned} &E[N_{i,j,k} \mid D_\ell^{obs}, \Theta^{[t]}] \\ =&P(X_i = k, X_{F_i} = \pi_{i,j} \mid D_\ell^{obs}, \Theta^{[t]}) \\ =&\theta_{i,j,k}^{[t]} P(X_{F_i} = \pi_{i,j} \mid D_\ell^{obs}, \Theta^{[t]}) \\ =&\prod_{\alpha=1}^{|F_i|} \varphi\Big(F_i(\alpha), j\Big) \theta_{i,j,k}^{[t]}, \end{aligned} \tag{25}$$

where function φ is defined in Eq. (8) and its computation method has been discussed in the Cases 1 & 2 of our proposed sampling method for missing data.

Given the discussion above, the optimal values of Θ w.r.t. a given directed edge set can be iteratively approximated by repeatedly using Eq. (18).

Complexity Analysis

POIND infers the diffusion network structure and updates infection transmission probabilities iteratively.

Graphs	Number of Nodes	Average Degree
LFR1-5	100,150,200,250,300	4
LFR6-10	200	2,3,4,5,6

Table 1: Properties of LFR benchmark graphs.

In structure inference, the most computationally expensive process consists of the following two parts, namely (1) sampling missing data, and (2) performing TWIND on completed data. In the sampling process, calculating the nodes' infection probabilities in β historical diffusion processes by Eq. (6) requires $O(\beta n^2 + 2^\eta \beta n)$ time, where n is the number of nodes in objective diffusion network, and η is the upper bound of the number of parent nodes ($\eta \ll n$). Performing TWIND requires about $O(\beta n^2)$ time.

The updating of infection transmission probabilities is carried out in an iterative way. In each updating iteration, the most computationally expensive process is calculating Eq. (23) with Eq. (25), which takes $O(2^\eta \eta \beta n)$ time. Let t be the number of all updating iterations, the time complexity of infection transmission probability updating is $O(2^\eta t \eta \beta n)$.

In summary, the overall time complexity of POIND algorithm is $O(T\beta n^2 + 2^\eta T \beta n + 2^\eta T t \eta \beta n)$, where T refers to the number of iterations of POIND.

Experimental Evaluation

In this section, we first introduce the experimental setup, and then verify the effectiveness and efficiency of our POIND algorithm on synthetic as well as real-world networks. To this end, we investigate the effects of diffusion network size, diffusion network's average degree, the ratio of missing data, the amount of diffusion processes, and the iterations of POIND on the accuracy and running time performance of POIND. All algorithms in the experiments are implemented in Python, running on a desktop PC with Intel Core i3-6100 CPU at 3.70GHz and 8GB RAM.

Experimental Setup

Network. We adopt LFR benchmark graphs (Lancichinetti, Fortunato, and Radicchi 2008) as the synthetic networks. By setting different generation parameters, such as the number of nodes and the average degree of each node, we generate two series of LFR benchmark graphs with properties summarized in Table 1. In addition, we adopt two real-world networks: NetSci (Newman 2006), a co-authorship network containing 379 scientists and 1602 co-authorships, and DUNF (Wang et al. 2014), a microblogging network containing 750 users and 2974 following relationships.

Infection Data. The infection status results S can be obtained by simulating β times of diffusion processes on each network with randomly selected initially infected nodes in each simulation (the ratio of initially infected nodes is 15%). Corresponding cascades are also recorded for cascade-based tested algorithms in the experiments. In each diffusion process, each infected node tries to infect its uninfected child nodes with an infection transmission probability that follows a Gaussian distribution with mean of 0.3 and standard deviation of 0.05, to make about 95% of infection

transmission probabilities are within a range from 0.2 to 0.4. We randomly remove a few infection status data from S as the missing data S^{mis} (let γ denote the ratio of missing data), and use the remaining partial observations S^{obs} for diffusion network inference.

Performance Criterion. To evaluate the accuracy performance of POIND algorithm, we report the F-score (the harmonic mean of precision and recall) of its inferred directed edges, computed as follows.

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

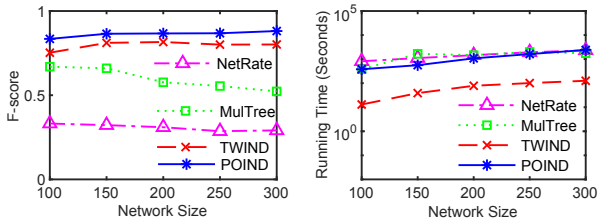
where $Precision = \frac{N_{TP}}{N_{TP}+N_{FP}}$, $Recall = \frac{N_{TP}}{N_{TP}+N_{FN}}$, N_{TP} refers to the number of true positives, i.e., the true edges which are correctly inferred by the algorithm; N_{FP} refers to the number of false positives, i.e., the wrong inferred edges which are not in the real network; and N_{FN} refers to the number of false negatives, i.e., the true edges which are not correctly inferred by the algorithm.

Benchmark Algorithms. We compare POIND with a classical convex optimization-based approach NetRate (Gomez-Rodriguez, Balduzzi, and Schölkopf 2011), a state-of-the-art submodular optimization-based approach MulTree (Gomez-Rodriguez and Schölkopf 2012), and a high performance infection timestamp-free approach TWIND (Huang et al. 2019b) for performance comparison, where TWIND is also used to infer the structure of objective diffusion network with completed data in our POIND algorithm. For POIND, the maximum number T of iterations is set to 5, the number m of sampling round is set to 6, and the stop condition for the iterative updates of infection transmission probabilities Θ is $\|\Theta^{[t+1]} - \Theta^{[t]}\| \leqslant 0.05$. As the three benchmark algorithms require complete infection data, we complete the S^{obs} with the following ad-hoc method for these benchmark algorithms: we estimate the average infection probability of each node in S^{obs}, and then sample for the missing data 6 times based on the average infection probabilities. Since NetRate infers the transmission rate between each two node in the network, we give NetRate a privilege in accuracy performance comparison, i.e., by calculating the F-score of edges whose transmission rates are greater than a threshold, we use different thresholds to find a highest F-score and report this F-score as the final accuracy performance of NetRate. Moreover, as MulTree requires users to specify the number of edges to be inferred, we use the actual number m of edges in the network as an input of MulTree.

Effect of Diffusion Network Size

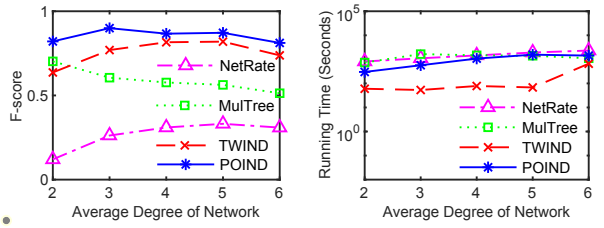
To study the effect of diffusion network size on algorithm performance, we adopt five synthetic networks, LFR1–5, whose sizes vary from 100 to 300. We simulate 200 times of diffusion processes on each network (i.e., $\beta = 200$), and randomly remove 15% of infection status observations as missing data (i.e., $\gamma = 0.15$).

Fig. 1 illustrates the F-score and execution time of each tested algorithm, from which we can observe that (1) among the three benchmark algorithms, TWIND achieves the highest accuracy and best running time performance; (2) compared with TWIND, POIND achieves a relatively higher accuracy at the price of a longer running time; (3) the running time of each tested algorithm increases with the growth of diffusion network size.



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

Figure 1: Effect of diffusion network size



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

Figure 2: Effect of average degree of diffusion network

Effect of Average Degree of Diffusion Network

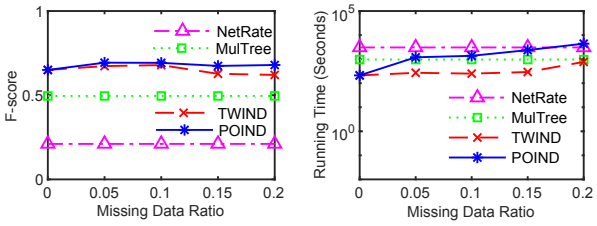
To study the effect of diffusion network's average degree on algorithm performance, we test the algorithms on five synthetic networks, LFR6–10, whose average node degree varies from 2 to 6. We simulate 200 times of diffusion processes on each network (i.e., $\beta = 200$), and randomly remove 15% of infection status observations as missing data (i.e., $\gamma = 0.15$).

Fig. 2 illustrates the F-score and running time of each algorithm, from which we can observe that (1) the POIND leads its competitors in accuracy, and is reasonably insensitive to the network's average degree; (2) the running time of each tested algorithm increases with the growth of average degree, and TWIND is faster than POIND as there is no iteration in TWIND.

Effect of Missing Data Ratio

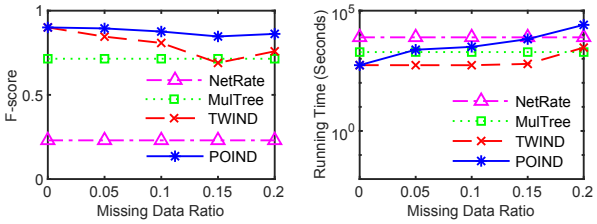
To study the effect of missing data ratio on algorithm performance, we test the algorithms on two real-world networks, NetSci and DUNF (with $\beta = 200$), varying the missing data ratio γ from 0 to 0.2.

Figs. 3 & 4 illustrate the F-score and running time of each tested algorithm on NetSci and DUNF, respectively. From the figures we can observe that (1) compared with TWIND, which often has the best accuracy among the benchmark algorithms, POIND has slightly higher accuracy on NetSci



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

Figure 3: Effect of missing data ratio on NetSci



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

Figure 4: Effect of missing data ratio on DUNF

and reasonably high accuracy on DUNF; (2) the increase of missing data ratio has a rather mild effect on the running time of NetRate, MulTree and TWIND, but results in longer running time for POIND. This is because POIND needs longer running time to infer more missing data, while the three benchmark algorithms use completed data directly.

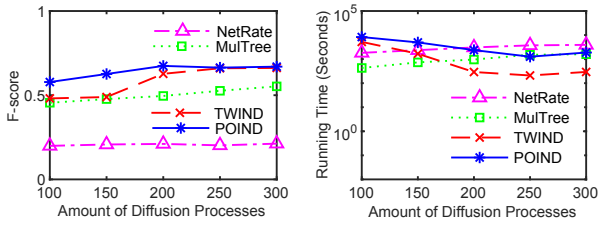
Effect of Amount of Diffusion Processes

To study the effect of the amount of diffusion processes on algorithm performance, we test the algorithms on NetSci and DUNF with different number β of diffusion processes, varying from 100 to 300. In the diffusion result obtained with each β, we randomly remove 15% of infection status observations as missing data (i.e., $\gamma = 0.15$).

Figs. 5 & 6 illustrate the F-score and running time of each algorithm on NetSci and DUNF, respectively. From the figures we can observe that (1) a greater amount of diffusion processes often helps the tested algorithms to achieve more accurate results on diffusion network structure inference. POIND often has a better accuracy performance compared with the other tested algorithms; (2) to analyze the infection status results observed from more diffusion processes, NetRate and MulTree often require more running time. A greater amount of diffusion processes does not always increase the running time of TWIND and POIND.

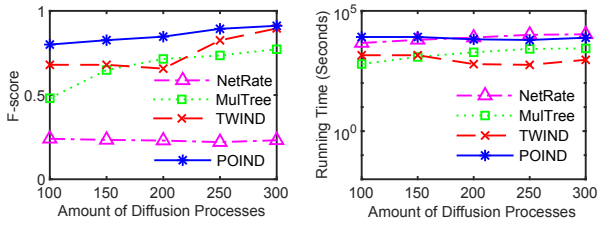
Effect of Iterations of POIND

To study the effect of POIND's iterations on the performance of POIND, we test POIND on NetSci and DUNF (with $\beta = 200$, $\gamma = 0.15$), varying the maximum number T of iterations from 1 to 5.



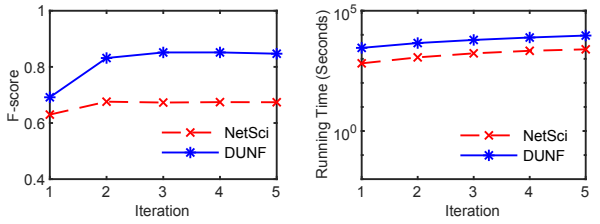
(a) *F-score* (b) *Running time*

Figure 5: Effect of amount of diffusion processes on NetSci



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

Figure 6: Effect of amount of diffusion processes on DUNF



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

Figure 7: Effect of iterations

Fig. 7 illustrates the tested results, from which we can observe that (1) POIND converges quickly; (2) more iterations result in more running time for POIND as a rule.

Conclusion

We have investigated the problem of how to infer the structure of a diffusion network from only the partial observations of final infection statuses of nodes. Towards this, we have proposed an effective approach called POIND to carry out structure inference and learning infection transmission probabilities in an iterative way. To infer the structure, POIND executes a probabilistic sampling for missing data, and then performs an existing infection timestamp-free method on the completed data. To learn infection transmission probabilities, POIND adopts an EM-like approach to iteratively approximate the optimal solution. Extensive experimental results on both synthetic and real-world networks have verified that our approach can effectively uncover the diffusion network structures from the partial observations.

Acknowledgments

This work was supported in part by the NSFC Grants 61976163, 61902284, 62025206, 61972338 and 61672392, the Technological Innovation Major Projects of Hubei Province under Grant No. 2017AAA125, the Science and Technology Program of Wuhan City under Grant No. 2018010401011288, and the Fundamental Research Funds for the Central Universities. Hao Huang is the corresponding author.

References

Amin, K.; Heidari, H.; and Kearns, M. 2014. Learning from Contagion(Without Timestamps). In *ICML 2014*, 1845–1853.

Daneshmand, H.; Gomez-Rodriguez, M.; Song, L.; and Schölkopf, B. 2014. Estimating Diffusion Network Structures: Recovery Conditions, Sample Complexity & Soft-thresholding Algorithm. In *ICML 2014*, 793–801.

Du, N.; Song, L.; Smola, A.; and Yuan, M. 2012. Learning Networks of Heterogeneous Influence. In *NIPS 2012*, 2780–2788.

Gomez-Rodriguez, M.; Balduzzi, D.; and Schölkopf, B. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *ICML 2011*, 561–568.

Gomez-Rodriguez, M.; Leskovec, J.; and Krause, A. 2010. Inferring Networks of Diffusion and Influence. In *KDD 2010*, 1019–1028.

Gomez-Rodriguez, M.; Leskovec, J.; and Schölkopf, B. 2013a. Modeling Information Propagation with Survival Theory. In *ICML 2013*, 666–674.

Gomez-Rodriguez, M.; Leskovec, J.; and Schölkopf, B. 2013b. Structure and Dynamics of Information Pathways in Online Media. In *WSDM 2013*, 23–32.

Gomez-Rodriguez, M.; and Schölkopf, B. 2012. Submodular Inference of Diffusion Networks from Multiple Trees. In *ICML 2012*, 489–496.

Gripon, V.; and Rabbat, M. 2013. Reconstructing a Graph from Path Traces. In *ISIT 2013*, 2488–2492.

Han, K.; Tian, Y.; Zhang, Y.; Han, L.; Huang, H.; and Gao, Y. 2020. Statistical Estimation of Diffusion Network Topologies. In *ICDE 2020*, 625–636.

He, X.; Xu, K.; Kempe, D.; and Liu, Y. 2016. Learning Influence Functions from Incomplete Observations. In *NIPS 2016*, 2065–2073.

Huang, H.; Yan, Q.; Chen, L.; Gao, Y.; and Jensen, C. S. 2019a. Statistical Inference of Diffusion Networks. *IEEE Transactions on Knowledge and Data Engineering* .

Huang, H.; Yan, Q.; Gan, T.; Niu, D.; Lu, W.; and Gao, Y. 2019b. Learning Diffusions without Timestamps. In *AAAI 2019*, 582–589.

Kalimeris, D.; Singer, Y.; Subbian, K.; and Weinsberg, U. 2018. Learning Diffusion using Hyperparameters. In *ICML 2018*, 2420–2428.

Lancichinetti, A.; Fortunato, S.; and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E* 78(4).

Lokhov, A. Y. 2016. Reconstructing Parameters of Spreading Models from Partial Observations. In *NIPS 2016*, 3467–3475.

Mehmood, Y.; Barbieri, N.; Bonchi, F.; and Ukkonen, A. 2013. CSI: Community-Level Social Influence Analysis. In *ECML PKDD 2013*, 48–63.

Myers, S.; and Leskovec, J. 2010. On the Convexity of Latent Social Network Inference. In *NIPS 2010*, 1741–1749.

Narasimhan, H.; Parkes, D. C.; and Singer, Y. 2015. Learnability of Influence in Networks. In *NIPS 2015*, 3186–3194.

Newman, M. E. J. 2006. Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Physical Review E* 74(3): 036104.

Pouget-Abadie, J.; and Horel, T. 2015. Inferring Graphs from Cascades: A Sparse Recovery Framework. In *ICML 2015*, 977–986.

Rong, Y.; Zhu, Q.; and Cheng, H. 2016. A Model-Free Approach to Infer the Diffusion Network from Event Cascade. In *CIKM 2016*, 1653–1662.

Sefer, E.; and Kingsford, C. 2015. Convex Risk Minimization to Infer Networks from Probabilistic Diffusion Data at Multiple Scales. In *ICDE 2015*, 663–674.

Wang, S.; Hu, X.; Yu, P.; and Li, Z. 2014. MMRate: Inferring Multi-aspect Diffusion Networks with Multi-pattern Cascades. In *KDD 2014*, 1246–1255.

Wilinski, M.; and Lokhov, A. Y. 2020. Scalable Learning of Independent Cascade Dynamics from Partial Observations. *arXiv preprint* arXiv:2007.06557.