Inferring Information Diffusion Networks without Timestamps

Yuchen Wang
wany810@mail.nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Dongpeng Hou
hdp0918@mail.nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Chao Gao
cgao@nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Xianghua Li*
li_xianghua@nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Zhen Wang
w-zhen@nwpu.edu.cn
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Abstract

The topology of diffusion networks plays an essential role in understanding information propagation dynamics and conducting social network analysis. However, diffusion networks are often unobservable in practical applications, leading to wide research on network inference from information cascades over the past decade. At present, novel cascades-based methods have been further developed to recover temporal dynamics and network topology by exploring the utilization of node temporal information, resulting in notable advancements. However, it requires high costs to acquire extensive temporal information, and the performance of network inference may decrease due to potential observational errors. Therefore, this paper specifically focuses on the time-independent scenario to address these limitations. Firstly, this paper models the node statuses of each diffusion process by leveraging the assumption of propagation trees based on the well-known independent cascade model. Subsequently, a gradient-based approach is developed to estimate the influences between nodes, facilitating the inference of network structure. Furthermore, this paper proposes a Monte Carlo EM-based approach to enhance the efficiency of network inference while maintaining comparable accuracy. Extensive experiments are conducted to verify the efficiency and effectiveness of our approaches on both synthetic and real-world networks.

CCS Concepts

• **Human-centered computing** → **Social network analysis**; **Social networks**; **Social network analysis**; • **Security and privacy** → *Social aspects of security and privacy*.

Keywords

Network inference, Information diffusion, Independent cascade model, Propagation tree

CIKM '24, October 21–25, 2024, Boise, ID, USA.

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679798

ACM Reference Format:
Yuchen Wang, Dongpeng Hou, Chao Gao, Xianghua Li, and Zhen Wang. 2024. Inferring Information Diffusion Networks without Timestamps. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3627673.3679798

1 Introduction

During the rapid growth of the Internet, the widespread sharing of information has become an integral part of our daily lives. Understanding the complex dynamics of information spread on social media platforms is crucial for various applications, such as detecting rumors [16] and implementing word-of-mouth marketing strategies [28]. A key aspect of these applications is comprehending the underlying structure that governs the diffusion processes across information networks. However, the specific mechanisms of information dissemination remain hidden in real-world scenarios, making it difficult to access the underlying topology of diffusion networks. As a result, researchers have dedicated significant efforts in the past decade to diffusion network inference, which aims to reconstruct network structures based on observable outcomes from information dissemination. This problem also tries to bridge the gap between the concealed mechanisms of information propagation and the underlying network structure.

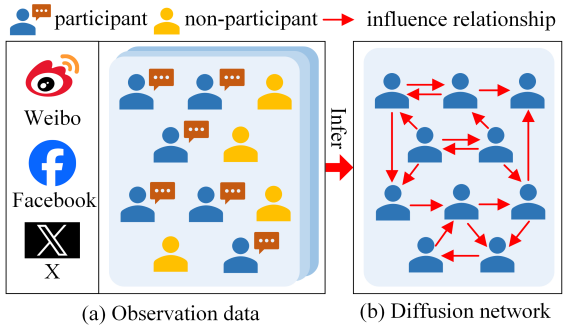

Figure 1: A diagram of diffusion network inference. In this paper, the observation data consists of the final states of nodes in each diffusion process. The goal is to infer the underlying influence relationships between the nodes.

Inspired by epidemiology, the spread of information is conceptualized by considering individuals involved in dissemination as infected, while those not participating remain uninfected. Then, the diffusion process of information is modeled as a stochastic process over a network. Early research primarily relied on temporal information. These cascade-based methods establish likelihoods based on observed infection times of nodes and aim to find a network that maximizes the likelihood. The propagation-tree-based model [11] and survival-analysis-based model [10] emerged as notable approaches in this research, leading to the development of diverse adaptations to improve the accuracy of inference [8, 13, 31]. Additionally, certain methods utilize the order of infected nodes to embed them into low-dimensional spaces, employing the distances between nodes in that space as a metric to quantify the strength of their relationships [3, 4, 9]. There are also methods that cluster pairs of nodes based on statistical differences in time intervals, aiming to identify a group of edges that better align with the patterns of propagation [18, 29]. These methods have achieved remarkable results in network inference but have also given rise to new challenges. Due to the significant resources required for monitoring and acquiring infection times of nodes, there is a growing interest in exploring if network inference can be achieved with less information. Consequently, diffusion network inference without timestamps has gained attention in recent years.

Time-independent methods for inferring diffusion networks face significant challenges due to the lack of explicit information, such as the sequential order of infections. Early studies in this field often simplified the problem by making specific assumptions, such as assuming the presence of all path-connected triples [15] or having prior knowledge of the number of edges [1]. Subsequently, several methods have been proposed based on parent-child influence relationships [7, 17, 20] to overcome certain limitations and enhance the applicability and accuracy of inference. These methods primarily focus on directly extracting potential relationships between nodes from infection statuses, while paying less attention to the dynamics of the diffusion process. To address this limitation, this paper proposes an approach that adopts a perspective of propagation trees from the Independent Cascade (IC) model [23], which has been a fundamental model for studying information and influence propagation in social networks.

This paper focuses on inferring the underlying diffusion networks without timestamps. Firstly, the likelihood is established for each set of infection statuses based on the IC model. In the IC model, it is assumed that the path of a diffusion process that starts from one node forms a tree structure, which is also known as a propagation tree. Then, we propose a gradient-based approach and a Monte Carlo EM-based approach to estimate the influences between nodes. The gradient-based approach utilizes the Matrix-tree theorem to derive the gradients of each directed edge. The MCEM-based method mainly employs Wilson's algorithm for generating samples of spanning trees to approximate the Q-function in the EM algorithm, ensuring accuracy while improving efficiency in inference. Finally, the strategies of cluster pruning and threshold selection are introduced to further enhance the accuracy of inference. The main contributions of this paper are as follows:

- Our approaches model the infection statuses of nodes through the propagation trees of the IC model. In contrast to other algorithms, this is the first approach that directly models these data using propagation trees and ensures the convexity of the objective function. Our code is available at https://github.com/cgao-comp/DNIT.
- We develop a gradient-based approach that utilizes the Matrix-tree theorem to compute gradients for each directed edge. Furthermore, a few optimization strategies are introduced to ensure that the solutions stay within the feasible region.
- Additionally, an MCEM-based approach has been developed, which involves sampling spanning trees from a latent distribution to approximate the Q-function. This approach enables efficient inference of diffusion networks.

2 Related Work

This section reviews the works related to our proposed approaches, specifically tree-based approaches and time-independent approaches.

2.1 Tree-based Approaches

Propagation trees have attracted considerable attention in the field of network inference as they serve as representations of prevalent patterns in information propagation paths [36]. Gomez-Rodriguez et al. proposed NetINF [11], a method that leverages tree patterns inspired by the IC model to model propagation paths. Then, the authors developed a sub-modular optimization to select k edges to construct a diffusion network. Later, they employed Kirchhoff's matrix tree theorem to consider all propagation trees, aiming to enhance inference accuracy [13]. Gray et al. extended this approach by integrating the tree-pattern-based likelihood with Bayesian inference [14], enabling the inference of network structures and estimation of uncertainties through sampling network distributions consistent with observed dynamics. Xu et al. proposed a model based on directed spanning trees that utilizes contrastive training and unsupervised training to improve the performance of information cascade models [33]. However, these methods need precise infection times of nodes. To the best of our knowledge, there are currently no known approaches specifically addressing time-independent inference using tree patterns without prior knowledge of the underlying networks.

2.2 Time-independent Approaches

While cascade-based methods have played a crucial role in inferring diffusion networks, they do have limitations. These limitations include the presence of unknown observation errors [34] and significant resource consumption for monitoring temporal information [21]. As a result, there has been a growing interest in exploring time-independent approaches.

Earlier studies proposed time-independent approaches for specific scenarios, such as assuming the presence of all path-connected triples [15] or requiring prior knowledge of the number of edges in a network [1]. However, these limitations hinder the broad application of these approaches. In recent years, novel methods

have emerged to facilitate network inference using infection statuses without any prior knowledge. One popular class of time-independent methods focuses on the parent-child influence relationship. These methods extract and analyze the relationship between the infection status of each node and the infection statuses of its parent nodes. Examples of such methods include TWIND [21], SIDN [20], and TENDS [17]. However, for larger-scale networks, the exponential growth in the number of combinations of infection statuses with the number of potential parents results in significant time and space consumption. Furthermore, some approaches estimate the influence relationships between nodes from different perspectives. For instance, Sun et al. developed FINITI [35], a method that infers the strength of links by estimating influence probabilities between nodes. Chen et al. introduced InDNI [6], which leverages a variational autoencoder (VAE) to extract behavioral relationships.

In addition to the aforementioned methods, some approaches propose general frameworks for addressing specific scenarios, such as missing data [7] and multi-aspect networks [19]. These methods often utilize the EM algorithm to treat missing data as latent variables and incorporate existing algorithms like TWIND for network inference. Additionally, Huang et al. introduced PIND [22] to address the problem of inferring diffusion networks. They formulated it as a constrained nonlinear regression that solely utilizes probabilistic information on node infection statuses.

In summary, these methods offer more reliable solutions for network inference in real-world applications. This paper differs from existing methods by modeling the infection statuses of nodes using propagation trees to estimate the influence relationships between nodes. Furthermore, the fundamental issues addressed in our approaches can also be utilized within the frameworks, such as [7], to improve the overall performance of their inference.

3 Methodologies

This section begins by formally defining the problem. The likelihood of node infection statuses is then modeled. Subsequently, a gradient-based optimization approach and a Monte Carlo-EM (MCEM)-based approach for optimization are proposed. Furthermore, the introduction of a pruning strategy and a threshold selection strategy aims to enhance the accuracy of network inference.

3.1 Problem Statement

Suppose that a diffusion network is represented as a directed graph $G=\{V, E_G\}$, where $V=\{v_1, v_2, ..., v_n\}$ is a set of n nodes, and E_G is the a of edges in a network. Meanwhile, $\mathbf{A}$ is the weighted adjacency matrix of G, where $0 \leq a_{ij} \leq 1$ represents the probability of node v_i propagating to node v_j. Under the assumption of the IC model, if there is a directed edge between nodes v_i and v_j in the diffusion network, it indicates that an infected node v_i has an opportunity to infect its uninfected neighbors v_j once, with a probability a_{ij} that contagion propagates over an edge (v_i, v_j) of G. Our problem can be formulated as follows.

Given: a set $S=\{S^1, ..., S^\beta\}$ of infection status on a diffusion network G in β diffusion processes, where $S^l=\{s_1^l, ..., s_n^l\}$ is a n-dimensional vector that records the final infection status $s_i^l \in \{0, 1\}$ (1 for infected status and 0 for uninfected status) of each node $v_i \in V$ observed in the l-th ($l \in \{1, ..., \beta\}$) diffusion process. Meanwhile, denote S_1^l as the set of infected nodes in S^l, and S_0^l as the set of uninfected nodes in S^l.

Infer: the influence probabilities matrix $\mathbf{A}$, and the diffusion network G.

3.2 Likelihood of Statuses of Nodes

In the independent cascade model, the path of propagation from a single node can be viewed as a propagation tree. For the tree T^l corresponding to a infection status S^l, each directed edge $(v_i, v_j) \in E_{T^l}$ represents an actual diffusion process. Then, the likelihood of S^l can be modeled as follows.

$$P(S^l|T^l, \mathbf{A}) = \prod_{(v_i,v_j)\in E_{T^l}} a_{ij} \prod_{v_u \in S_1^l, v_k \in S_0^l} (1 - a_{uk}). \tag{1}$$

However, propagation trees are not directly accessible, it is necessary to model the likelihood $P(S^l|\mathbf{A})$ considering all possible propagation trees that could exist in a diffusion process, which can be formulated as follows.

$$P(S^l|\mathbf{A}) = \sum_{T^l \in \mathcal{T}^l(\mathbf{A})} P(S^l|T^l, \mathbf{A}) P(T^l), \tag{2}$$

where $\mathcal{T}^l(\mathbf{A})$ is the set of all possible trees that can be generated by $\mathbf{A}$. Also, we assume $P(T^l) = \frac{1}{|\mathcal{T}^l(\mathbf{A})|}$ without any priors.

Then, the log-likelihood $\log P(S|A)$ for all diffusion processes S is shown as Eq. (3).

$$\begin{aligned}
\log P(S|\mathbf{A}) &= \sum_{l=1}^{\beta} \log P(S^l|\mathbf{A}) \\
&= \sum_{l=1}^{\beta} \left(\log \left(\sum_{T^l \in \mathcal{T}^l(\mathbf{A})} P(S^l|T^l, \mathbf{A}) \right) - \log |\mathcal{T}^l(\mathbf{A})| \right) \\
&\propto \sum_{l=1}^{\beta} \log \left(\prod_{v_u \in S_1^l, v_k \in S_0^l} (1 - a_{uk}) \left(\sum_{T^l \in \mathcal{T}^l(\mathbf{A})} \prod_{(v_i,v_j)\in E_{T^l}} a_{ij} \right) \right).
\end{aligned} \tag{3}$$

Inferring the diffusion network is equivalent to obtaining $\mathbf{A}^* = \text{argmax}_{\mathbf{A}} \log P(S|\mathbf{A})$ through maximum likelihood estimation (MLE).

3.3 Gradient-based Approach

In this section, we utilize the matrix-tree theorem to propose a gradient-based approach for optimizing this problem.

Theorem 3.1 (Matrix-tree theorem). *Given a directed graph with non negative weighted adjacency* $\mathbf{A}$ *, construct a matrix* $\mathbf{W}$ *such that* $w_{ij} = \sum_k a_{kj}$*, if* $i = j$ *and* $w_{ij} = -a_{ij}$ *if* $i \neq j$ *and denote the matrix created by removing any row* x *and column* y *from* $\mathbf{W}$ *as* $\mathbf{W}_{\{x,y\}}$*, also known as the minor of a element* w_{xy}*. Then,*

$$(-1)^{x+y} \det(\mathbf{W}_{\{x,y\}}) = \sum_{T \in \mathcal{T}_y(\mathbf{A})} \prod_{(v_i,v_j)\in E_T} a_{ij}, \tag{4}$$

where $\mathcal{T}_y(\mathbf{A})$ *is the set of directed spanning trees in* $\mathbf{A}$ *that starts in* v_y*,* $\det(\mathbf{W})$ *is the determinant of* $\mathbf{W}$*.*

Based on the Matrix-tree theorem, the infected nodes S_1^l of l-th diffusion process can be further derived as Eq. (5).

$$\begin{aligned}\sum_{T\in\mathcal{T}^l(\mathbf{A})}\prod_{(v_i,v_j)\in E_T} a_{ij} &= \sum_{v_k\in S_1^l}\sum_{T\in\mathcal{T}_k^l(\mathbf{A})}\prod_{(v_i,v_j)\in E_T} a_{ij} \\ &= \sum_{v_k\in S_1^l}\det(\mathbf{W}_{\{k,k\}}),\end{aligned} \tag{5}$$

Considering that computing $|S_1^l|$ determinants would increase the complexity, we will continue to construct a new matrix $\hat{\mathbf{W}}$ to simplify this problem inspired by [24]. Each element $\hat{w}_{ij}$ in $\hat{\mathbf{W}}$ is

$$\hat{w}_{ij} = \begin{cases} 1, & \text{if} \quad i = 1; \\ w_{ij}, & \text{if} \quad i > 1. \end{cases} \tag{6}$$

Then, it can be proved that

$$\det(\hat{\mathbf{W}}) = \sum_{i=1}^{n}\det(\mathbf{W}_{\{i,i\}}). \tag{7}$$

Proposition 3.2 (The equation $\det(\hat{\mathbf{W}}) = \sum_{i=1}^{n}\det(\mathbf{W}_{\{i,i\}})$ holds). *Consider the row expansion of $\hat{W}$ with respect to row 1, then,*

$$\begin{aligned}\det(\hat{\mathbf{W}}) &= \sum_{i=1}^{n}(-1)^{1+i}\hat{\mathbf{W}}_{1i}\det(\hat{\mathbf{W}}_{\{1,i\}}) \\ &= \sum_{i=1}^{n}(-1)^{1+i}\det(\hat{\mathbf{W}}_{\{1,i\}}) \\ &= \sum_{i=1}^{n}\det(\mathbf{W}_{\{i,i\}}),\end{aligned}$$

since each element in row 1 is 1, and the minor $\hat{\mathbf{W}}_{\{1,i\}} = \mathbf{W}_{\{1,i\}}$.

The log likelihood $\log P(S|\mathbf{A})$ can be expressed by follows.

$$\begin{aligned}\log P(S|\mathbf{A}) \propto &\sum_{l=1}^{\beta}\sum_{v_u\in S_1^l, v_k\in S_0^l}\log(1-a_{uk}) + \\ &\sum_{l=1}^{\beta}\log\det(\hat{\mathbf{W}}^l),\end{aligned} \tag{8}$$

where $\hat{\mathbf{W}}^l$ is the augmented matrix correspond to S_1^l. Then, the derivative of $\log P(S|A)$ is calculated by the chain rule,

$$\begin{aligned}\frac{\partial\log P(S|\mathbf{A})}{\partial a_{ij}} = &\sum_{l=1}^{\beta}\mathbb{I}(v_i,v_j\in S_1^l)\frac{\partial\log\det(\hat{\mathbf{W}}^l)}{\partial a_{ij}} - \\ &\sum_{l=1}^{\beta}\mathbb{I}(v_i\in S_1^l, v_j\in S_0^l)\frac{1}{1-a_{ij}},\end{aligned} \tag{9}$$

where $\mathbb{I}(\cdot)$ is a indicator function. The derivative of logarithm of determinant can be shown in Eq. (10).

$$\begin{aligned}\frac{\partial\log\det(\hat{\mathbf{W}})}{\partial a_{ij}} &= \frac{\partial\log\det(\hat{\mathbf{W}})}{\partial\hat{\mathbf{W}}}\frac{\partial\hat{\mathbf{W}}}{\partial a_{ij}} \\ &= \sum_{i'=1}^{N}\sum_{j'=1}^{N}\frac{\partial\log\det(\hat{\mathbf{W}})}{\partial\hat{\mathbf{W}}_{i'j'}}\frac{\partial\hat{\mathbf{W}}_{i'j'}}{\partial a_{ij}} \\ &= \frac{\partial\log\det(\mathbf{W})}{\partial\hat{\mathbf{W}}_{jj}} - \frac{\partial\log\det(\mathbf{W})}{\partial\hat{\mathbf{W}}_{ij}} \\ &= (\hat{\mathbf{W}}^{-1})_{jj}^T - (\hat{\mathbf{W}}^{-1})_{ij}^T.\end{aligned} \tag{10}$$

Therefore, the Jacobian matrix $\mathbf{J}(\mathbf{A})$ for $\log P(S|\mathbf{A})$ can be estimated as Eq. (11).

$$\mathbf{J}(\mathbf{A})_{ij} = \begin{cases} \sum_{l=1}^{\beta}\mathbb{I}(v_i,v_j\in S_1^l)\left((\hat{\mathbf{W}}^l)^{-1}\right)_{jj} - (\hat{\mathbf{W}}^l)^{-1})_{ji}\Big) - \\ \sum_{l=1}^{\beta}\mathbb{I}(v_i\in S_1^l, v_j\in S_0^l)\frac{1}{1-A_{ij}}, \text{if} \quad i\neq j; \\ 0, \quad \text{if} \quad i = j. \end{cases} \tag{11}$$

To maximize the log-likelihood $\log P(S|\mathbf{A})$, a gradient ascent method is employed to optimize $\mathbf{A}$. In each iteration t,

$$\mathbf{A}^{t+1} = \mathbf{A}^t + \lambda\mathbf{J}(\mathbf{A}^t), \tag{12}$$

where λ is the step size. It can also be proved that the objective is a convex optimization problem, since all sub-terms of Eq. (8) are convex functions in a feasible region $[0, 1]$.

To ensure that every element in matrix $\mathbf{A}$ is within the feasible region during the optimization, several strategies can be considered, such as projected gradient descent [12] and a self-adjusting step size method [22]. The projected gradient descent method primarily involves projecting the updated variables onto the feasible region, which, in our case, is the interval [0, 1]. This approach is simple to implement and exhibits good convergence properties.

$$\mathbf{A}^{t+1} = (\mathbf{A}^t + \lambda\mathbf{J}(\mathbf{A}^t))^+, \tag{13}$$

where $(x)^+ = \max(0, \min(1, x))$. For the self-adjusting step size method, an adjusted step size λ' can be defined as follows.

$$\lambda'_{ij} = \begin{cases} \dfrac{a_{ij}}{-z_{ij}}, & \text{if}\, z_{ij} < 0; \\ \dfrac{1-a_{ij}}{z_{ij}}, & \text{if}\, z_{ij} > 0; \\ +\infty, & \text{if} \quad z_{ij} = 0. \end{cases} \tag{14}$$

$$z_{ij} = \begin{cases} 0, & \text{if}\, a_{ij} = 0, \mathbf{J}(\mathbf{A})_{ij} < 0; \\ 0, & \text{if}\, a_{ij} = 1, \mathbf{J}(\mathbf{A})_{ij} > 0; \\ \mathbf{J}(\mathbf{A})_{ij}, & \text{otherwise}. \end{cases} \tag{15}$$

Let $\lambda' = \min\{\lambda'_{ij} | i, j \in \{1, ..., n\}\}$, then there exists an integer m that $\log(S|\mathbf{A} + \frac{\lambda}{2^m}\mathbf{J}(\mathbf{A})) \geq \log(S|\mathbf{A})$.

While this strategies effectively ensures that the solution remains within the feasible region, it can be computationally slow when dealing with large-scale networks. Then, the second strategy is to

introduce an additional matrix $\mathbf{X} \in \mathrm{R}^{n\times n}$, which is related to matrix $\mathbf{A}$ as follows.

$$x_{ij} = \begin{cases} -\log(\frac{1}{a_{ij}} - 1), & \text{if } i \neq j; \\ 0, & \text{if } i = j. \end{cases} \tag{16}$$

So the optimization of $\log P(S|\mathbf{A})$ can be transformed into an unconstrained optimization of $\log P(S|\mathbf{X})$ through the inverse of the sigmoid function. This strategy also enables optimization through automatic differentiation in deep learning [2].

3.4 Monte-Carlo EM Approach

Considering that the gradient-based method mentioned above involves matrix inversion, it may encounter computational efficiency bottlenecks when dealing with large-scale networks. Therefore, we also propose a MCEM-based method to solve to this problem approximately. In the framework of EM, it can be divided into two steps, namely E-step and M-step. In the E-step, we define a Q-function $Q(\mathbf{A};\mathbf{A}^t)$=$E[\log P(S,\mathbf{T}|\mathbf{A})|S,\mathbf{A}^t]$ in iteration t, which means computing the expectation of $\log P(S,\mathbf{T}|\mathbf{A})$ based on the current estimation $\mathbf{A}^t$, where $\mathbf{T}$ is the set of propagation trees for all diffusion processes. In M-step, $\mathbf{A}^{t+1} = \mathrm{argmax}_{\mathbf{A}} Q(\mathbf{A};\mathbf{A}^t)$.

However, computing the expectation of a propagation tree conditioned on $\mathbf{A}$, specifically the expectation of edges in the tree, can be computationally inefficient. To address this, we utilize Monte Carlo simulation for computation in the E-step, which is also known as the MCEM algorithm. The Q-function in the MCEM algorithm is expressed in Eq. (16).

$$\begin{aligned} Q(\mathbf{A};\mathbf{A}^t) &\approx \hat{Q}(\mathbf{A};\mathbf{A}^t) \\ &= \frac{1}{M^t}\sum_{k=1}^{M^t} \log P(S,\mathbf{T}^{(k)}|\mathbf{A}) \\ &= \frac{1}{M^t}\sum_{l=1}^{\beta}\sum_{k=1}^{M^t} \log P(S^l, T^{l(k)}|\mathbf{A}) \\ &= \frac{1}{M^t}\sum_{l=1}^{\beta}\sum_{k=1}^{M} \log P(S^l|T^{l(k)},\mathbf{A}) + \log P(T^{l(k)}), \end{aligned} \tag{17}$$

where $\log P(T^{l(k)})$ we assume constant, M^t is the sample size of iteration t, $T^{l(k)}$ is a directed spanning tree sampled from $P(T^l|\mathbf{A}^t, S^l)$.

$$\begin{aligned} P(T^l = T^{l(k)}|\mathbf{A}^t, S^l) &= \frac{P(S^l|T^l = T^{l(k)},\mathbf{A}^t)}{P(S|\mathbf{A}^t)} \\ &= \frac{P(S^l|T^l = T^{l(k)},\mathbf{A}^t)}{\sum_{T\in\mathcal{T}^l(\mathbf{A}^t)} P(S^l|T^l = T,\mathbf{A}^t)} \\ &= \frac{\prod_{(v_i,v_j)\in E_{T^{l(k)}}} a_{ij}}{\sum_{T\in\mathcal{T}^l(\mathbf{A}^t)}\prod_{(v_{i'},v_{j'})\in E_T} a_{i'j'}}. \end{aligned} \tag{18}$$

Then, in M-step, each element a_{ij}^{t+1} in $\mathbf{A}^t$ can be estimated as follows.

$$\begin{aligned} a_{ij}^{t+1} &= \frac{C_{ij}^{++}}{C_{ij}^{++} + M^t C_{ij}^{+-}}, \\ C_{ij}^{++} &= \sum_{l=1}^{\beta}\sum_{k=1}^{M^t} \mathbb{I}((v_i,v_j)\in E_{T^{l(k)}}), \\ C_{ij}^{+-} &= \sum_{l=1}^{\beta} \mathbb{I}(v_i \in S_1^l, v_j \in S_0^l)). \end{aligned} \tag{19}$$

For sampling directed spanning trees, we apply the Wilson's algorithm [32]. Wilson shows that any graph may be converted into a stochastic graph, by adjusting the edge weights to be locally normalized along all incoming edges. Therefore, the edge (v_i, v_j) has a weight a'_{ij} defined by

$$a'_{ij} = \frac{a_{ij}}{\sum_k a_{kj}}. \tag{20}$$

In addition, Wilson has proved that this algorithm has a probabilistic bound of $O(H)$, where H is the mean hitting time of the graph. Therefore, it is generally expected that the complexity of sampling trees is lower than that of computing the inverse of a matrix.

$$H = \sum_{ij} \pi_i \pi_j h(v_i, v_j), \tag{21}$$

where π is the stationary distribution of the Markov chain, and $h(v_i, v_j)$ is the expected number of steps to reach node v_j starting from node v_i.

Despite the lower time complexity compared to gradient descent when dealing with large-scale networks, the MCEM-based approach also has its certain limitations. In a exact EM, it can be proved that the iterates satisfy the monotonicity.

$$Q(\mathbf{A}^{t+1};\mathbf{A}^t) \geq Q(\mathbf{A}^t;\mathbf{A}^t). \tag{22}$$

However, in the MCEM framework, if the sample size M_t remains the same in each iteration, the inequality $\hat{Q}(\mathbf{A}^{t+1};\mathbf{A}^t) \geq \hat{Q}(\mathbf{A}^t;\mathbf{A}^t)$ may not hold, which means that the optimization may not converge. However, in practice, the MCEM-based approach can still exhibit good convergence similar to the gradient-based approach. Alternatively, we can employ strategies such as importance sampling [26] or utilize variant MCEM methods like ascent-based MCEM [5] to enhance the stability of convergence.

3.5 Pruning Candidate Edges

To enhance the accuracy of network inference, a pruning method can be utilized to select candidate edges that satisfy specific criteria and incorporate them into the final results. In this paper, the Mutual Information (MI) metric is employed as a clustering metric for the pruning process.

$$\begin{aligned} MI(v_i,v_j) = \sum_{X_i\in\{0,1\}}\sum_{X_j\in\{0,1\}} & P(s_i = X_i, s_j = X_j)\times \\ & \log\frac{P(s_i = X_i, s_j = X_j)}{P(s_i = X_i)P(s_j = X_j)}. \end{aligned} \tag{23}$$

After calculating the mutual information (MI) of all node pairs, a modified K-means algorithm is used to divide the node pairs into two clusters, with a cluster C_0 always fixed with a centroid of zero. Then, the threshold τ is chosen as the maximum MI value in the cluster C_0. Finally, the set of v_i's neighbors N_i is filtered based on the threshold τ, as shown in the following Eq. (24).

$$N_i = \{v_j \in V | MI(v_i, v_j) > \tau\}. \tag{24}$$

3.6 Threshold for Network Inference

Another crucial factor that influences the performance of network inference is the threshold. In most related studies, a threshold value close to zero, such as 0.01, is commonly selected as the filtering threshold. However, we have discovered that a threshold selection method called predicted break-even point (BEP) [27] is better suited for our approaches, particularly in large-scale networks. This is because predicted BEP prioritizes precision and holds an advantage in larger networks due to their sparsity.

Predicted-BEP is an unsupervised selection procedure for determining the cut-off θ^* such that $\hat{\text{FP}}(\theta^*) = \hat{\text{FN}}(\theta^*)$.

$$\begin{aligned} \hat{\text{FP}}(\theta^*) &= \sum_{(v_i,v_j);a^*_{ij}\geq\theta^*} (1 - a^*_{ij}), \\ \hat{\text{FN}}(\theta^*) &= \sum_{(v_i,v_j);a^*_{ij}<\theta^*} a^*_{ij}. \end{aligned} \tag{25}$$

The final inferred set of edges E_{G^*} of G^* is determined as $E_{G^*} = \{(v_i, v_j) | MI(v_i, v_j) > \tau, a^*_{ij} \geq \theta^*\}$.

3.7 Time Complexity Analysis

Suppose the average number of infected nodes in each diffusion process is denoted by $\bar{n}$. The time complexity of each iteration in the gradient-based approach is approximately $O(\beta\bar{n}^3)$. On the other hand, the time complexity of each iteration t in the MCEM-based approach is approximately $O(\beta M^t \bar{H})$. Regarding the pruning method, it requires $O(\beta n^2)$ time to calculate the MI values for each pair of nodes. The subsequent K-means clustering using these MI values also takes approximately $O(kn^2)$ time, where k represents the number of iterations. In summary, the overall time complexity of the gradient-based method is roughly $O(\beta n^2 + kn^2 + K\beta\bar{n}^3)$. On the other hand, the overall time complexity of the MCEM-based method is approximately $O(\beta n^2 + kn^2 + K\beta M^t \bar{H})$, where K represents the number of optimization iterations.

4 Experiments

In this section, we conduct experiments on both synthetic and real-world datasets to evaluate the performance of our proposed approaches. Additionally, we discuss the convergence, scalability, and incorporation of prior information in our methods.

4.1 Datasets

In this paper, we conduct experiments using both synthetic networks and real-world networks. For the synthetic networks, we generate 10 different networks by the LFR benchmark graph model [25]. The LFR benchmark graph model is highly valuable for generating networks with diverse degrees and community structures. The setting of synthetic networks can also be found in related works [17, 22].

Table 1: The details of synthetic networks.

Networks	Number of nodes	Average degree
G1-G5	2000	4,5,6,7,8
G6-G10	1000,1500,2000,2500,3000	4

In addition to the synthetic networks, we also utilize two real-world networks, namely NetSci and Wiki-vote [30]. NetSci is a network of co-authorships in the area of network science. Wiki-vote is a network which contains all the Wikipedia voting data from the inception of Wikipedia till January 2008.

Table 2: The details of real-world networks.

Network	Number of nodes	Number of edges
NetSci	379	914
Wiki-vote	889	2914

For the infection data, this study utilizes the IC model to simulate information diffusion processes and record the final statuses of each node after β diffusion iterations as the input data S. The influence probabilities a_{ij} of each directed edge are sampled from a uniform distribution $U[0.1, 0.3]$.

4.2 Baselines

To demonstrate the validity and novelty of our approaches, we compare our approaches with other state-of-the-art time-independent methods, including K-Lifts [1], TENDS [20], InDNI [6], and PIND [22]. K-Lifts defines a lift metric to quantify the relationships between nodes and selects the K largest lifts in a greedy manner. TENDS introduces a novel scoring criterion to assess the relationships between nodes and their parent nodes. InDNI utilizes deep generative models to uncover latent relationships between nodes. PIND is a method that evaluates the probabilities of influence between nodes while considering the uncertainty of node states. In the problem addressed in this paper, where the observation data contains precise infection statuses of each node, PIND theoretically guarantees no loss in performance. The settings of the comparative algorithms are consistent with their respective source papers to maintain consistency. To ensure the efficiency, we set the number of samples M=10 for our MCEM-based approach.

4.3 Evaluation Metrics

In this paper, we evaluate the performance of algorithms by the *F1-score* metric, as formulated as Eq. (26).

$$\begin{aligned} Precision &= \frac{N_{TP}}{N_{TP} + N_{FP}}, \; Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}, \\ F1 - score &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \end{aligned} \tag{26}$$

where N_{TP} represents the number of correctly inferred edges, N_{FP} represents the number of incorrectly inferred edges, N_{FN} represents the number of true edges which are not correctly inferred by

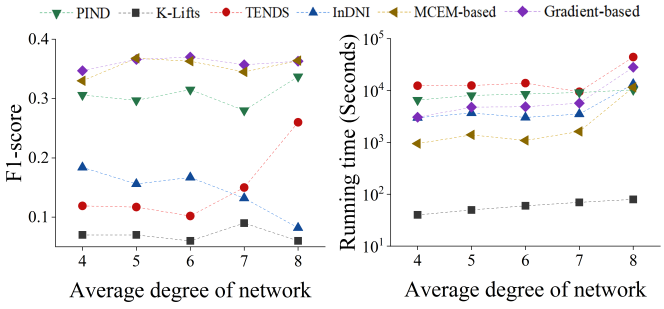

Figure 2: Effect of networks' average degree.

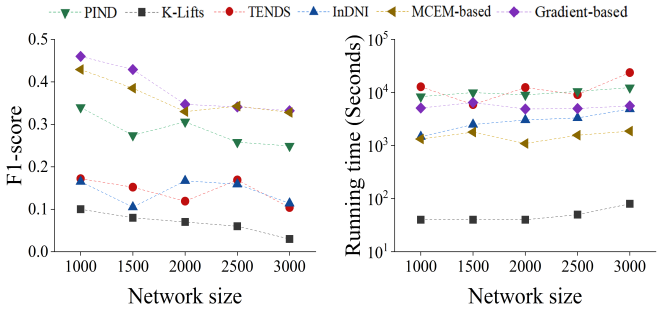

Figure 3: Effect of network size.

the algorithm. Additionally, we will record the algorithm's runtime to provide a visual comparison of computational efficiency.

4.4 Experimental Results

In this section, the effect of average degree, network size, and the amount of infection data is discussed. The first two experiments are conducted on G1-G5 and G6-G10 networks with $\beta = 1000$ infection data [1]. Then, we experiment with different amounts of infection data to show the effect of data quantity on two real-world networks and a synthetic network.

4.4.1 Effect of Average Degree. In Fig. (2), our proposed approaches and PIND show better performance compared to other approaches on networks G1-G5, and overall changes are relatively stable. Specifically, the gradient-based method has slightly higher accuracy than the MCEM-based method, but it requires more running time. TENDS performs better in inferring networks with higher degrees, but it consumes the most running time. This is because, in large networks, the number of node combinations grows exponentially, resulting in high algorithmic complexity. InDNI's performance gradually decreases as the network degree increases. One possible reason is that networks with higher degrees often have a larger number of infected nodes in a single propagation process, and InDNI, which relies solely on the frequency of occurrence between nodes, struggles to handle this situation. Although K-Lifts has the shortest running time, it is difficult to infer the correct network.

4.4.2 Effect of Network Size. In Fig. (3), the results on networks G6-G10 indicate that as the network size increases, all algorithms show a certain degree of performance decline, primarily due to insufficient infected data. A larger network requires more infected data. Specifically, the gradient-based approach performs the best, followed by the MCEM-based approach. TENDS and InDNI have similar performance. In terms of running time, the MCEM-based approach requires less time while maintaining high accuracy.

4.4.3 Effect of Amount of Diffusion Processes. The experiments are conducted on both the NetSci network and the Wiki-vote network,

[1]It should be noted that 1000 infection data are typically insufficient to satisfy the condition for complete network recovery. In fact, even with temporal information, some work suggests that it can recover the network by $O(d^3 \log(n))$ cascades, where d is the maximum number of parents of a node. This paper adopts a dataset setup similar to related works, and the discussion of the quantity condition for complete network recovery is beyond the scope of this paper.

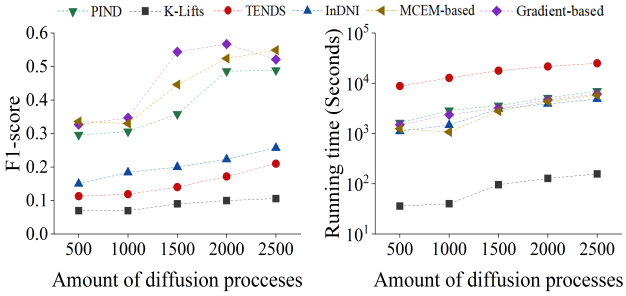

Figure 4: Effect of the amount of diffusion processes on G1.

with diffusion processes ranging from 100 to 500. We also test algorithms on the G1 network with 500-2500 diffusion processes.

In Fig. (4), the experimental results demonstrate that most approaches show improved performance as the data quantity increases. The gradient-based approach continues to outperform, while TENDS and MCEM-based approaches achieve similar levels of performance. PIND shows a noticeable improvement as the data quantity increases. In terms of runtime, TENDS still has a higher execution time, while the gradient-based approach and MCEM approach have similar runtime, since in smaller-sized networks, the computational cost of matrix inversion is not significant. In comparison to other algorithms, InDNI and K-Lifts demonstrate relatively weaker performance, with no significant improvements observed as the amount of data increases.

In Fig. (5), our approaches continue to perform well. Additionally, TENDS and PIND also demonstrate competitiveness. Compared to larger-scale networks, TENDS shows significant improvements in performance on smaller networks. InDNI is comparable to PIND in 100-300 diffusion processes but does not show a noticeable improvement with increasing data quantity. In terms of runtime, all algorithms, except for K-Lifts, perform similarly without significant differences. K-Lifts still struggles to achieve high accuracy in network inference.

In Fig. (6), as the number of diffusion increases, the performance of all algorithms also improves. Among them, TENDS performs slightly better than the other algorithms. Fig. (5) and (6) demonstrate that TENDS is more competitive in small-scale networks. PIND and our gradient-based method exhibit similar performances.

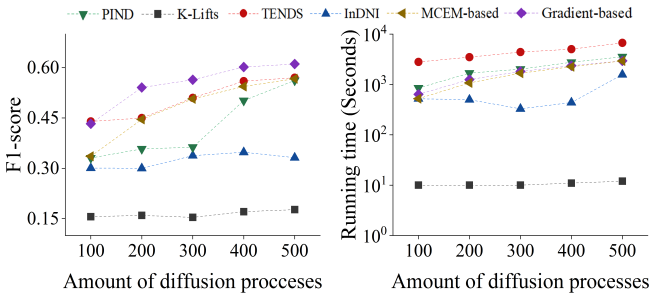

Figure 5: Effect of the amount of diffusion processes on the NetSci network.

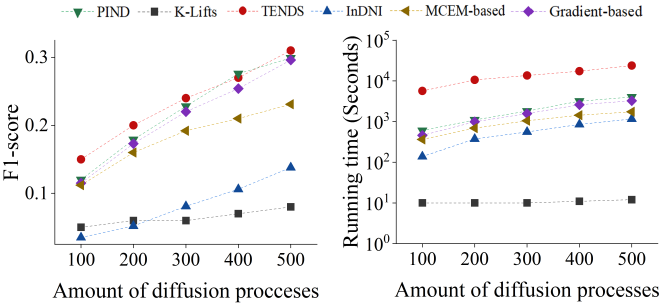

Figure 6: Effect of the amount of diffusion processes on the Wiki-vote network.

The MCEM-based method shows slightly lower performance, possibly due to insufficient sampling, which prevents convergence to the global maxima. Regarding runtime, consistent with previous results, it is evident that the TENDS algorithm still requires more time for execution. In summary, our proposed approaches demonstrate strong competitiveness across networks of varying sizes and different amounts of data. The gradient-based approach exhibits higher accuracy, while the MCEM-based methods enhance inference efficiency while maintaining accuracy.

4.5 Discussion

4.5.1 Convergence. In this section, we conduct experiments on a LFR network with 100 nodes and average degree 4 to show the convergence of both the gradient-based and MCEM-based approaches.

In Fig. (7), it can be observed that the gradient-based method converges quickly. The red line in the figure represents the log-likelihood calculated under the true network topology and influence probabilities. The final convergence result closely approximates the likelihood of the true values, indicating that our proposed method plays a positive role in inferring networks even with limited information. It also can be found that the changes in the Q-function are shown for sampling sizes M=10 and M=50. The Q-function is slightly higher for M=50 compared to M=10. Overall, increasing the value of M generally leads to enhanced stability in the convergence of MCEM, but it comes at the cost of reduced efficiency.

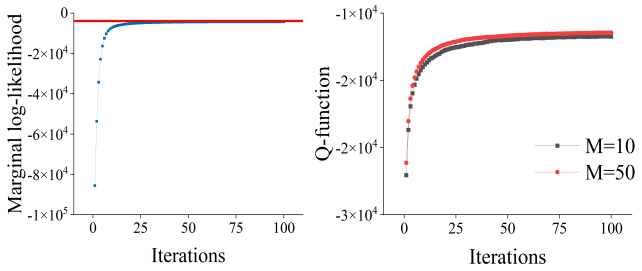

Figure 7: The convergence of the gradient-based approach and the MCEM-based approach.

4.5.2 Scalability. The experiments presented in Section 4.3 demonstrate that our method achieves improved accuracy while maintaining an acceptable timeframe for network inference. Moreover, for larger-scale networks, the utilization of the MCEM-based approach provides computational efficiency advantages. This is attributed to the lower upper limit of the mean hitting time compared to computing the matrix inverse as the number of nodes increases. Additionally, our approaches can leverage multiprocessing to compute each diffusion process separately. In the case of the MCEM-based method, multiprocessing can also be employed to handle tree sampling, further enhancing scalability.

4.5.3 Incorporating Prior Information. If we have access to prior knowledge, such as the distribution of influence rates, we can convert the maximum likelihood estimation (MLE) to maximum a posteriori (MAP) estimation to incorporate this prior knowledge. For the gradient-based method, we can incorporate the prior term $P(\mathbf{A})$ into the objective function and compute the gradient.

$$\log P(\mathbf{A}|S) \propto \log P(S|\mathbf{A}) + \log P(\mathbf{A}). \tag{27}$$

$$Q(\mathbf{A}; \mathbf{A}^t) = E[\log P(S, \mathbf{T}|\mathbf{A})|S, \mathbf{A}^t] + \log P(\mathbf{A}). \tag{28}$$

In the MCEM-based approach, we observe that incorporating prior information into the EM algorithm primarily affects the M-step. In this section, we primarily evaluate our gradient-based approach on an LFR network with an average degree of 4. The influence probabilities, represented by the matrix $\mathbf{A}$, follow a normal distribution $N(0.3, 0.05)$. Additionally, we also examine the performance of a weight-inverse graph prior on a scale-free network with $\alpha = 2$. The error in estimating $\hat{\mathbf{A}}$ is measured by mean absolute error (MAE).

$$\begin{aligned} P(\mathbf{A}) &= \prod_{i=1}^{n} f(\sum_{j} \frac{1}{1 - a_{ji}})), \\ f(x) &= Cx^{-\alpha}, \end{aligned} \tag{29}$$

where C is a constant and α is the parameter of the power-law distribution. Based on this prior knowledge, we demonstrate the effect of incorporating the prior information on the inference results.

Table 3: Comparison of algorithm performance with incorporated prior information.

Network	Prior	F1-score	MAE
LFR average=4	-	0.748	0.03
(n=100)	normal prior	0.768 (2% ↑)	0.026 (14% ↓)
Power-law	-	0.748	0.032
(n=100,α=2)	graph prior	0.756 (1% ↑)	0.031 (4% ↓)

From the results, incorporating prior information leads to estimated $\hat{\mathbf{A}}$ values that are closer to the true values $\mathbf{A}$ and further improves the accuracy of network inference.

5 Conclusion

In conclusion, this paper focuses on inferring underlying diffusion networks without timestamps. The study begins by establishing the likelihood for each set of infection statuses using the IC model, which assumes a diffusion process follows a tree structure known as a propagation tree. Then, a gradient-based approach and a Monte Carlo EM-based approach are proposed to estimate the influences between nodes. The gradient-based approach utilizes the Matrix-tree theorem and an augmented graph matrix to compute gradients for each directed edge. On the other hand, the MCEM-based method employs Wilson's algorithm to generate samples of spanning trees, approximating the Q-function in the EM algorithm. Experiments conducted on synthetic and real-network datasets demonstrate the superiority of our approaches in terms of both inference accuracy and efficiency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62271411, U22A2098, 62261136549, U22B2036), Fok Ying-Tong Education Foundation, China (No. 171105), the Fundamental Research Funds for the Central Universities (No. G2024WD0151) and the Tencent Foundation and XPLORER PRIZE.

References

[1] Amin, K., Heidari, H., and Kearns, M. Learning from contagion (without timestamps). In *International Conference on Machine Learning* (2014), PMLR, pp. 1845–1853.

[2] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of machine learning research 18*, 153 (2018), 1–43.

[3] Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM international conference on Web search and data mining* (2014), pp. 393–402.

[4] Bourigault, S., Lamprier, S., and Gallinari, P. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the Ninth ACM international conference on Web Search and Data Mining* (2016), pp. 573–582.

[5] Caffo, B. S., Jank, W., and Jones, G. L. Ascent-based monte carlo expectation–maximization. *Journal of the Royal Statistical Society Series B: Statistical Methodology 67*, 2 (2005), 235–251.

[6] Chen, G., Wang, Y., Shao, J., Shi, B., Shen, H., and Cheng, X. Indni: An infection time independent method for diffusion network inference. In *China Conference on Information Retrieval* (2022), Springer, pp. 63–75.

[7] Gan, T., Han, K., Huang, H., Ying, S., Gao, Y., and Li, Z. Diffusion network inference from partial observations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 7493–7500.

[8] Gao, C., Wang, Y., Wang, Z., Li, X., and Li, X. Pairwise-interactions-based bayesian inference of network structure from information cascades. In *Proceedings of the ACM Web Conference 2023* (2023), pp. 102–110.

[9] Gao, S., Pang, H., Gallinari, P., Guo, J., and Kato, N. A novel embedding method for information diffusion prediction in social network big data. *IEEE Transactions on Industrial Informatics 13*, 4 (2017), 2097–2105.

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

[11] Gomez-Rodriguez, M., Leskovec, J., and Krause, A. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD) 5*, 4 (2012), 1–37.

[12] Gomez Rodriguez, M., Leskovec, J., and Schölkopf, B. Structure and dynamics of information pathways in online media. In *Proceedings of the sixth ACM international conference on Web search and data mining* (2013), pp. 23–32.

[13] Gomez Rodriguez, M., and Schölkopf, B. Submodular inference of diffusion networks from multiple trees. In *29th International Conference on Machine Learning (ICML 2012)* (2012), International Machine Learning Society, pp. 489–496.

[14] Gray, C., Mitchell, L., and Roughan, M. Bayesian inference of network structure from information cascades. *IEEE Transactions on Signal and Information Processing over Networks 6* (2020), 371–381.

[15] Gripon, V., and Rabbat, M. Reconstructing a graph from path traces. In *2013 IEEE International Symposium on Information Theory* (2013), IEEE, pp. 2488–2492.

[16] Guo, B., Ding, Y., Yao, L., Liang, Y., and Yu, Z. The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR) 53*, 4 (2020), 1–36.

[17] Han, K., Tian, Y., Zhang, Y., Han, L., Huang, H., and Gao, Y. Statistical estimation of diffusion network topologies. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (2020), IEEE, pp. 625–636.

[18] Hu, S., Cautis, B., Chen, Z., Chan, L., Geng, Y., and He, X. Model-free inference of diffusion networks using rkhs embeddings. *Data Mining and Knowledge Discovery 33* (2019), 499–525.

[19] Huang, H., Han, K., Xu, B., and Gan, T. Multi-aspect diffusion network inference. In *Proceedings of the ACM Web Conference 2023* (2023), pp. 82–90.

[20] Huang, H., Yan, Q., Chen, L., Gao, Y., and Jensen, C. S. Statistical inference of diffusion networks. *IEEE Transactions on Knowledge and Data Engineering 33*, 2 (2019), 742–753.

[21] Huang, H., Yan, Q., Gan, T., Niu, D., Lu, W., and Gao, Y. Learning diffusions without timestamps. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 582–589.

[22] Huang, H., Yan, Q., Han, K., Gan, T., Jiang, J., Xu, Q., and Yang, C. Learning diffusions under uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2024), vol. 38, pp. 20430–20437.

[23] Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), pp. 137–146.

[24] Koo, T., Globerson, A., Carreras Pérez, X., and Collins, M. Structured prediction models via the matrix-tree theorem. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (2007), pp. 141–150.

[25] Lancichinetti, A., Fortunato, S., and Radicchi, F. Benchmark graphs for testing community detection algorithms. *Physical review E 78*, 4 (2008), 046110.

[26] Levine, R. A., and Casella, G. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics 10*, 3 (2001), 422–439.

[27] Li, X., and Li, X. Reconstruction of stochastic temporal networks through diffusive arrival times. *Nature communications 8*, 1 (2017), 15729.

[28] Robledo, S., Duque, P., and Aguirre, A. M. G. Word of mouth marketing: A scientometric analysis. *Journal of Scientometric Research 11*, 3 (2023), 436–446.

[29] Rong, Y., Zhu, Q., and Cheng, H. 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* (2016), pp. 1653–1662.

[30] Rossi, R., and Ahmed, N. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence* (2015), vol. 29.

[31] Wang, S., Hu, X., Yu, P. S., and Li, Z. 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* (2014), pp. 1246–1255.

[32] Wilson, D. B. Generating random spanning trees more quickly than the cover time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996), pp. 296–303.

[33] Xu, S., and Smith, D. Contrastive training for models of information cascades. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), vol. 32.

[34] Young, J.-G., Cantwell, G. T., and Newman, M. Bayesian inference of network structure from unreliable data. *Journal of Complex Networks 8*, 6 (2020), cnaa046.

[35] Yueming, S., Yunjia, Z., Qian, Y., Lu, C., Hao, H., and Yunjun, G. Fast inference algorithm of diffusion networks without infection temporal information. *Journal of Frontiers of Computer Science & Technology 13*, 4 (2019), 541.

[36] Zhan, X.-X., Hanjalic, A., and Wang, H. Information diffusion backbones in temporal networks. *Scientific reports 9*, 1 (2019), 6798.