Reconstructing an Epidemic Outbreak Using Steiner Connectivity

Ritwick Mishra[1,2], **Jack Heavey**[1,2], **Gursharn Kaur**[1], **Abhijin Adiga**[1], **Anil Vullikanti**[1,2]

[1] Biocomplexity Institute & Initiative, University of Virginia
[2] Department of Computer Science, University of Virginia
{mbc7bu, jch7jm, fug3aj, abhijin, vsakumar}@virginia.edu

Abstract

Only a subset of infections is actually observed in an outbreak, due to multiple reasons such as asymptomatic cases and under-reporting. Therefore, reconstructing an epidemic cascade given some observed cases is an important step in responding to such an outbreak. A maximum likelihood solution to this problem (referred to as CASCADEMLE) can be shown to be a variation of the classical Steiner subgraph problem, which connects a subset of observed infections. In contrast to prior works on epidemic reconstruction, which consider the standard Steiner tree objective, we show that a solution to CASCADEMLE, based on the actual MLE objective, has a very different structure. We design a logarithmic approximation algorithm for CASCADEMLE, and evaluate it on multiple synthetic and social contact networks, including a contact network constructed for a hospital. Our algorithm has significantly better performance compared to a prior baseline.

Introduction

In most outbreaks (of diseases, pests, or pathogens), such as COVID-19 (Shaman et al. 2020), Hospital Associated Infections (HAIs), such as Methicillin-resistant *Staphylococcus aureus* (MRSA), and biological invasions (Robinson et al. 2017) only a subset of infections is known in a timely manner, due to multiple reasons, including asymptomatic cases (Jang et al. 2021), and lack of resources for tracing or inspection. Such events require a prompt response, such as isolation of infected patients (important in the case of HAIs, since hospitalized patients are very vulnerable) and corresponding measures in the case of crops and livestock. Therefore, reconstructing an epidemic cascade which identifies missing infections (or infestations), and explains the pattern of spread an important problem.

The problems of reconstructing an epidemic cascade and identifying the source have been studied extensively for both SI and SIR models on networks (Rozenshtein et al. 2016; Jang et al. 2021; Zhu and Ying 2014; Shah and Zaman 2011). These works assume partial information is available about the cascade, e.g., a subset of nodes which are known to be infected (Jang et al. 2021), or both infection state and time of infection (Rozenshtein et al. 2016; Zhu and Ying 2014; Shah and Zaman 2011).

In the simplest SIR type network model of epidemic spread (also referred to as the independent cascade (IC) model), where a disease spreads on each edge of a contact network $G = (V, E)$ with probability p, this reconstruction problem (referred to as CASCADEMLE) involves finding a connected subgraph $T = (V(T), E(T))$ such that $S \subseteq V(T)$, where S is the subset of observed infections. Using the natural maximum likelihood estimation (MLE) approach, the goal of the CASCADEMLE problem is to find a connected Steiner subgraph T such that $S \subseteq V(T)$, and $\Pr[T]$ is maximized—this can be expressed as an equivalent problem of minimizing cost of T, denoted by $\overline{\mathrm{Cost}}(T)$. As discussed later in the Preliminaries section, $\overline{\mathrm{Cost}}(T)$ includes cost of edges $e \in E(T)$, as well as cost of edges $= (u, v) \notin E(T)$ not in T, but with $\{u, v\} \cap V(T) \neq \emptyset$ (i.e., at least one end point of e is in $E(T)$). This makes $\overline{\mathrm{Cost}}(T)$ quite different from the standard Steiner tree objective (Karp 1972), which has been very well studied. Most prior work on reconstructing epidemic cascades in the SIR model has mainly been restricted to the regular Steiner tree objective, e.g., (Jang et al. 2021; Rozenshtein et al. 2016); this immediately connects with the vast literature on algorithms for the Steiner tree problem. We note that Zhu and Ying (2014) consider the actual cost, but mainly focus on trees for rigorous analysis (which is then extended to general graphs through various heuristics).

Our Contributions. We study the CASCADEMLE problem of finding an MLE solution for reconstructing an epidemic cascade for the independent cascade (IC) model.

- We show that the solution to CASCADEMLE can have a very different structure from the solution to the regular Steiner tree objective. In particular, there exist instances where the solution to CASCADEMLE has diameter $\Theta(n)$, where as the Steiner tree solution can have constant diameter. We observe a significant difference in real instances as well.
- We study the conditions under which the MLE solution to the cascade reconstruction problem will fail. We find that the MLE based approach is not good in many classes of instances.
- The CASCADEMLE hasn't been studied before. We show that it is NP-hard. For the independent cascade model, we present an algorithm with a logarithmic ap-

proximation factor, under natural assumptions about the structure of social contact networks.

- Finally, we evaluate our formulation and algorithms for several synthetic and realistic contact networks, including a contact network for the University of Virginia (UVA) hospital, constructed using Electronic Health Record (EHR) data. Our results show improved performance compared to a prior baseline in identifying missing infections.

Related Work

Our paper is most closely related to the work on cascade reconstruction and source identification on networks where only a partial set of nodes are observed. Rozenshtein et al. (2016) introduce a directed Steiner tree based algorithm, CuLT, for reconstructing an epidemic cascade, where the underlying network is dynamic, and a subset of infections, along with their infection times, is given. They do not make any assumptions about the diffusion model. Jang et al. (2021) consider this problem in the asymptomatic infection setting and propose taking into account the individual's risk factors in the form of node attributes. They formulate this as a directed prize-collecting Steiner tree problem on a temporal contact network with outbreaks starting independently from multiple sources. However, both of these works minimize the regular Steiner tree objective, which consists of only the costs of edges within the subgraph. In contrast, we consider the true MLE cost which includes the costs from edges outside the subgraph corresponding to failed infection attempts.

Shah and Zaman (2011) were the first to study the source detection problem. They consider the SI model, and assume all the infections are given, and the goal is to determine the source. They study the ML estimator for the source detection problem, and show that it can be solved exactly on trees using a notion of rumor centrality. Zhu and Ying (2014) extend this work to the SIR model. They develop an optimal sample-path-based approach to source detection.

A related line of research is the problem of sampling from the space of all cascades that explain the given observations. Xiao, Aslay, and Gionis (2018) map the problem of computing the node infection probabilities, given partial observations, to the problem of sampling Steiner trees. However, in their target sampling distribution, they only consider the contributions of edges within the tree, and ignore the contribution from edges outside the tree corresponding to failed infection attempts.

Preliminaries

Spread model. We consider the simplest form of the Susceptible-Infected-Recovered (SIR) model called the *Independent Cascade* model on a contact graph $G = (V, E)$. In this model, each node is in one of Susceptible (S), Infectious (I) or Recovered (R) state. We start off with all nodes in Susceptible state except the single *source* node which is in Infected state. At each time-step, an infected node u can infect each susceptible neighbor v with probability $p_{u,v}$, independent of other neighbors of v. Each infected node gets only one opportunity to spread the infection following which they enter into Recovered state.

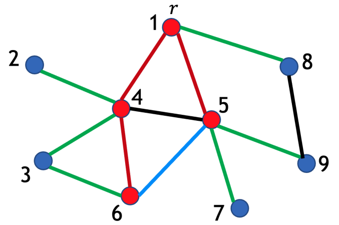

Figure 1: In this example, node 1 is the source, the red nodes are the infections, and the brown edges represent the infection cascade T. Here, δ_T is $\{(2,4),(3,4),(1,8),(5,9),(5,7)\}$, and λ_T is $\{(4,5),(6,5)\}$. Each edge e in T contributes a cost c_e to $\overline{\text{Cost}}(T)$. Edges in δ_T and λ_T contribute d_e costs, except for edge $(4,5)$. Neighbors 4 and 5 get infected at the same time, so they cannot attempt to infect each other.

Probability of a cascade. An outbreak, starting at a node r, is referred to as a *cascade*, and can be represented as a subgraph $T_r = (V(T_r), E(T_r))$, rooted at r. Let δ_{T_r} be the set of edges not in T_r with exactly one endpoint in T_r, i.e., $\delta_{T_r} = \{(u,v) \in E \setminus E(T_r) : u \in V(T_r), v \notin V(T_r)\}$. Let λ_{T_r} be the set of edges not in T_r with both endpoints in T_r, i.e. $\lambda_{T_r} = \{(u,v) \in E \setminus E(T_r) : u, v \in V(T_r)$. Under the IC dynamics, the probability of the cascade T_r is:

$$\overline{P}(T_r) = \prod_{e\in E(T_r)} p_e \prod_{e\in\delta_{T_r}} (1-p_e) \prod_{\substack{e=(u,v)\in\lambda_{T_r},\\ d_{T_r}(r,u)\neq d_{T_r}(r,v)}} (1-p_e) \tag{1}$$

where $d_{T_r}(r,u)$ denotes the distance between root node r and u, in the subgraph T_r. The first term corresponds to the contribution of edges in the subgraph. Since every infected node gets a single chance to infect a susceptible neighbor, we have two kinds of $(1-p_e)$ terms contributed by edges not in T_r: (a) with exactly one endpoint in T_r, and (b) with both endpoints in T_r, as long as they are at different distances from the root. Let λ'_{T_r} denote the set of edges of type (b).

MLE solution. We assume that subsets S_0, S_1 are given where S_0 is a set of nodes which are *known not to be infected* in the outbreak, while S_1 is a set of nodes *known to be infected*. We also assume the outbreak starts at a single node, which need not be in S. We say that a cascade T_r is *consistent* with (S_0, S_1) if $S_1 \subset V(T)$ and $S_0 \subset V \setminus V(T)$. The MLE problem involves finding a connected subgraph $T_r = (V(T_r), E(T_r))$ rooted at a node r which is consistent with the given (S_0, S_1), and maximizes $\overline{P}(T_r)$; this is equivalent to the optimal sample path detection problem as described in (Zhu and Ying 2014). Taking the log of the probabilities in $\overline{P}(T_r)$, we can define the cost of T_r as

$$\overline{\text{Cost}}(T_r) = \sum_{e\in E(T_r)} c_e + \sum_{e\in\delta_{T_r}} d_e + \sum_{\substack{e=(u,v)\in\lambda_{T_r},\\ d_{T_r}(r,u)\neq d_{T_r}(r,v)}} d_e \tag{2}$$

Here, $c_e = -\log p_e$ is the cost of including an edge e in the subgraph and $d_e = -\log(1-p_e)$ is the cost of excluding an edge e from the subgraph. In Figure 1, we have an illustrative example.

The CASCADEMLE problem. Given subsets S_0, S_1, the goal is to find a connected subgraph T_r rooted at some node r, which is consistent with (S_0, S_1), and minimizes $\overline{\mathrm{Cost}}(T_r)$.

Theorem 1. CASCADEMLE *is NP-hard.*

Proof. (Sketch) We show this by a reduction from the standard Steiner tree problem which is NP-hard (Karp 1972). Consider the class of instances of CASCADEMLE where the homogeneous edge probability is $p = \frac{1}{4n^2}$, n being the number of nodes in the graph. Thus, $c = 2\log 2n$ and $d = -\log(1 - \frac{1}{4n^2}) \leq \frac{2}{4n^2}$ for large n. The CASCADEMLE problem is to find the consistent subgraph which minimizes

$$\begin{aligned}\overline{\mathrm{Cost}}(T_r) &\leq 2\log 2n|E(T_r)| + \frac{1}{2n^2}|\delta_{T_r} \cup \lambda'_{T_r}| \\ &\leq 2\log 2n|E(T_r)| + \frac{1}{2n^2}.n^2 \\ &= 2\log 2n|E(T_r)| + \frac{1}{2}\end{aligned}$$

We use the fact that there can be at most n^2 edges in $\delta_{T_r} \cup \lambda'_{T_r}$. It can be verified that $\overline{\mathrm{Cost}}(T_r) \geq D$ if and only if $|E(T_r)| \geq D$ for any integer $D \leq n-1$. The NP-hardness of CASCADEMLE follows from it.

□

We say a solution T_r is an α-approximation if $\overline{\mathrm{Cost}}(T_r) \leq \alpha\overline{\mathrm{Cost}}(T^\star_{r\star})$, where $T^\star_{r\star}$ is an optimal solution to the instance of CASCADEMLE. Note that the root of T_r and $T^\star_{r\star}$ need not be the same; we only need that T_r be consistent with (S_0, S_1).

Remark. In practice, the costs of exclusion of the edges between same-level nodes in the cascade, $\sum_{(u,v)\in\lambda_T, d_T(r,u)=d_T(r,v)} d_{(u,v)}$, is a very small fraction of $\overline{\mathrm{Cost}}(T_r)$ (as we verify in our experiments). This is also supported by the analysis of (Adcock, Sullivan, and Mahoney 2013) that many realistic social and information networks are tree-like, where this condition will hold. In such a setting, $P(T)$ is a good approximation to $\overline{P_r}(T_r)$:

$$P(T) = \prod_{e\in E(T)} p_e \prod_{e\in\delta_T}(1-p_e) \prod_{e\in\lambda_T}(1-p_e). \tag{3}$$

Observe that $P(T)$ does not depend on the root. We consider the corresponding cost,

$$\mathrm{Cost}(T) = \sum_{e\in E(T)} c_e + \sum_{e\in\delta_T} d_e + \sum_{e\in\lambda_T} d_e \tag{4}$$

and will focus on minimizing $\mathrm{Cost}(T)$.

Algorithm 1 (described in the Approach section) considers the setting where we are given an undirected contact graph $G = (V, E)$ with each edge e having an infection probability p_e , and a set of observed infections $S \subset V$. The set of observed infected nodes S forms the terminal set in the output Steiner tree. For a node u, let $\mathcal{N}_e(u)$ denote the set of edges connecting to its neighbors. In Algorithm 2, we show that with a slight modification, our approach extends to the setting where we are also given the set of observed uninfected nodes.

Difference between CASCADEMLE and Steiner Tree Solutions

As mentioned earlier, previous works (Rozenshtein et al. 2016; Jang et al. 2021) minimize the regular Steiner tree cost which consists of only the first term in $\overline{\mathrm{Cost}}(T_r)$, namely $\mathrm{Cost}_{st}(T) = \sum_{e\in E(T)} c_e$. Here we show that a solution which minimizes $\mathrm{Cost}_{st}(T)$, can have a very different structure than that which minimizes $\overline{\mathrm{Cost}}(T_r)$. Next we study the conditions under which the MLE approach will fail by showing that there exist instances in which a CASCADEMLE solution does not recover the ground truth cascade.

Observation 1. *There exist instances in which a* CASCADEMLE *solution* $T_r = \arg\min_{T'_r} \overline{\mathrm{Cost}}(T'_r)$ *has diameter* $\Theta(n)$*, while a Steiner tree solution* $T_{st} = \arg\min_{T'} \mathrm{Cost}_{st}(T')$ *has diameter* $\Theta(1)$.

Proof. Consider the class of graphs in Figure 2 where $A_1 \cup A_2$ form a complete bipartite graph with $|A_1| = |A_2| = N+1$ and terminal node set $S = \{r, t\}$. Assume the homogeneous setting i.e. $c_e = c, d_e = d$ for every $e \in E$. Consider trees $T_1 = (r, w_1, w'_1, t)$, and $T_2 = (r, u_1, \ldots, u_N, t)$. Observe that $\mathrm{Cost}_{st}(T_1) = 3c$ and $\mathrm{Cost}_{st}(T_2) = (N+1)c$. It can be verified that T_1 minimizes the $\mathrm{Cost}_{st}(\cdot)$ objective. On the other hand, we have $\overline{\mathrm{Cost}}(T_1) = 3c + 2Nd + 2d$ and $\overline{\mathrm{Cost}}(T_2) = (N+1)c + 2d$. It can be verified that there exists a sufficiently large value of p for which T_2 minimizes $\overline{\mathrm{Cost}}(\cdot)$, and thus, is a CASCADEMLE solution. Hence, there exist regimes in which the CASCADEMLE solution has a diameter $\Theta(n)$, while the Steiner tree solution has a diameter $\Theta(1)$. □

Observation 2. *There exist instances in which a* CASCADEMLE *solution does not recover the true cascade.*

Proof. Consider the class of graphs in Figure 2. Suppose T_1 is the ground-truth cascade comprising nodes $\{r, t, w_1, w'_1\}$. Given terminal set $S = \{r, t\}$, the MLE approach will pick T_2 over T_1, unless the cost of excluding the bipartite edges is insignificant, i.e., p is small enough. Thus, there exist regimes in which the MLE solution fails to recover any part of the ground truth cascade. □

Our Approach

Assumption 1. *For every edge in the network,* $p_e \leq 1/2$*. Equivalently,* $c(e) \geq d(e)$*, for all* $e \in E$.

Lemma 1. *Under Assumption 1, a* CASCADEMLE *solution* $T^\star$ *is a tree.*

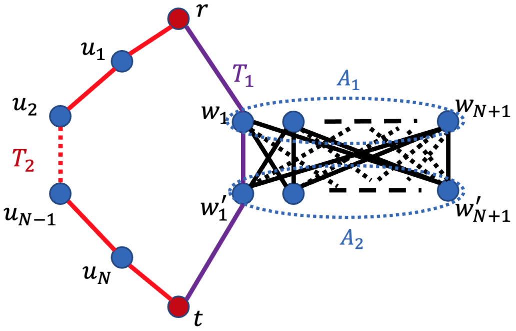

Figure 2: In this example, node r is the root, and $S = \{r, t\}$ is the set of terminals. $A_1 \cup A_2$ is a complete bipartite graph on nodes $w_1, \ldots, w_{N+1}$, and nodes $w'_1, \ldots, w'_{N+1}$. T_1 is the purple path between r, t through w_1, w'_1, while T_2 is the red path between r, t through nodes $u_1, .., u_N$.

Assumption 1 states that the cost of including an edge is greater than than the cost of excluding it. This implies that an optimal subgraph is a tree, as we can always reduce the cost of the subgraph by excluding (rather than including) any edge that forms part of a cycle. Without this assumption, there exist instances where an optimal solution could have cycles. For example, in the homogeneous setting where all edges have the same costs c and d and $c < d$, an optimal solution could be one which spans the whole graph. Thus, under Assumption 1, we leverage a reduction from the CASCADEMLE problem to the *node-weighted Steiner tree problem* in our algorithm MINCOSTSTEINERTREE.

In Algorithm 1, we construct a node and edge-weighted graph by weighing each node with the sum of the costs of exclusion of each of its incident edges, and each edge with the difference between its costs of inclusion and exclusion. Under Assumption 1, these edge weights are non-negative. Converting this to a purely node-weighted graph, this becomes a node weighted Steiner tree problem, where the goal is to find the minimum-weighted Steiner tree with terminal set S.

Algorithm 1 runs in polynomial time as we can construct the node-weighted graph in polynomial time and the Klein and Ravi (1995) algorithm has a polynomial time implementation. We now prove that this algorithm has a logarithmic approximation factor.

Theorem 2. *Let $\hat{T}$ be the tree returned by Algorithm 1, and let T^* be an optimal solution to the* CASCADEMLE *instance. Then $\hat{T}$ is consistent with S, and*

$$\text{Cost}(T^*) \le \text{Cost}(\hat{T}) \le 4 \ln |S| \cdot \text{Cost}(T^*) \tag{5}$$

Proof. For any Steiner tree T,

$$\sum_{u\in V(T)} w(u) + \sum_{e\in E(T)} w(e)$$
$$= \sum_{u\in V(T)} \sum_{e\in\mathcal{N}_e(u)} d_e + \sum_{e\in E(T)} (c_e - d_e)$$

Algorithm 1: MINCOSTSTEINERTREE

```
Input: An undirected contact graph $G = (V, E)$ with edge probabilities $p_e$ and a set of observed infected nodes $S$
Output: Tree $T_r$ consistent with $S$
for each edge $e$ do
    Compute the cost of inclusion $c_e = -\log p_e$ and cost of exclusion $d_e = -\log(1 - p_e)$
end for
Construct a node and edge-weighted graph $G'$ from $G$, by assigning weights as below:
for each node $u$ do
    $w(u) \leftarrow \sum_{e\in\mathcal{N}_e(u)} d_e$
end for
for each edge $e$ do
    $w(e) \leftarrow c_e - d_e$
end for
Convert $G'$ to a purely node-weighted graph $\hat{G}$ by splitting each edge with a new node having the same weight.
Find the minimum weighted Steiner tree $\hat{T}$ in $\hat{G}$ with terminal set $S$, using Klein and Ravi's (1995) algorithm for the node-weighted Steiner tree problem.
Let $r$ be any node in $\hat{T}$
return $\hat{T}$, with root $r$
```

$$= \sum_{u\in V(T)} \sum_{e\in\mathcal{N}_e(u)\cap E(T)} d_e + \sum_{u\in V(T)} \sum_{e\in\mathcal{N}_e(u)\cap\lambda_T} d_e$$
$$+ \sum_{u\in V(T)} \sum_{e\in\mathcal{N}_e(u)\cap\delta_T} d_e + \sum_{e\in E(T)} (c_e - d_e)$$
$$= 2\sum_{e\in E(T)} d_e + 2\sum_{e\in\lambda_T} d_e + \sum_{e\in\delta_T} d_e + \sum_{e\in E(T)} (c_e - d_e)$$
$$= \sum_{e\in E(T)} c_e + \sum_{e\in E(T)} d_e + 2\sum_{e\in\lambda_T} d_e + \sum_{e\in\delta_T} d_e$$
$$= \text{Cost}(T) + \sum_{e\in E(T)} d_e + \sum_{e\in\lambda_T} d_e \tag{6}$$
$$\Rightarrow \text{Cost}(T) \le \sum_{u\in V(T)} w(u) + \sum_{e\in E(T)} w(e)$$

Continuing from (6) and using Assumption 1,

$$\sum_{u\in V(T)} w(u) + \sum_{e\in E(T)} w(e)$$
$$\le \text{Cost}(T) + \sum_{e\in E(T)} c_e + \sum_{e\in\lambda_T} d_e$$
$$= 2\text{Cost}(T) - \sum_{e\in\delta_T} d_e \le 2\,\text{Cost}(T)$$

Thus, for any Steiner tree T, we have shown that

$$\text{Cost}(T) \le \sum_{u\in V(T)} w(u) + \sum_{e\in E(T)} w(e) \le 2\,\text{Cost}(T) \tag{7}$$

This holds for the Steiner tree returned by the algorithm, $\hat{T}$.

$$\text{Cost}(\hat{T}) \le \sum_{u\in V(\hat{T})} w(u) + \sum_{e\in E(\hat{T})} w(e) \tag{8}$$

Klein and Ravi's algorithm has a worst-case approximation factor of $2 \ln |S|$. Hence,

$$\begin{aligned}
&\sum_{u \in V(\hat{T})} w(u) + \sum_{e \in E(\hat{T})} w(e) \\
&\quad \le 2 \ln |S| \Big(\sum_{u \in V(T^*)} w(u) + \sum_{e \in E(T^*)} w(e) \Big) \\
&\quad \le 4 \ln |S| \mathrm{Cost}(T^*) \qquad (9)
\end{aligned}$$

Combining (8) and (9),

$$\mathrm{Cost}(\hat{T}) \le 4 \ln |S| \mathrm{Cost}(T^*) \qquad (10)$$

□

Observed Uninfected Nodes

Our approach extends easily to the setting where we are given the set of observed uninfected nodes S_0 in addition to the set of observed infected nodes S_1. Our goal is to find an optimal subgraph consistent with S_0 and S_1. Here we assume the observed uninfected nodes were never part of the cascade, i.e., they remained uninfected throughout the spreading process.

Let $\kappa = \{(u, v) \in E \setminus E(T) : u \in S_0, v \in T\}$ be the set of edges with one endpoint in S_0 and the other in the cascade T. In this setting, define δ_T as the set of edges not in cascade, with one endpoint in cascade and the other in $V \setminus (S_0 \cup V(T))$, i.e., $\delta_T = \{(u, v) \in E \setminus E(T) : u \in V(T), v \in (V \setminus (S_0 \cup V(T))\}$. Here λ_T is the same as before, i.e, $\lambda_T = \{(u, v) \in E \setminus E(T) : u, v \in T\}$. Then the cost of a subgraph T, under IC dynamics, can be defined as:

$$\mathrm{Cost}(T) = \sum_{e \in E(T)} c_e + \sum_{e \in \lambda_T} d_e + \sum_{e \in \delta_T} d_e + \sum_{e \in \kappa} d_e \qquad (11)$$

Our goal is to find a connected subgraph consistent with (S_0, S_1), and which minimizes $\mathrm{Cost}(T)$. Here we present Algorithm 2, MINCOSTSTEINERTREE-OBS-UNINFECTED, which is only a slight modification of our previous algorithm: we remove the nodes known to be uninfected (and their edges), after constructing the node and edge-weighted graph. This ensures that the graph weighting takes into account the removed edges and the returned tree is consistent with S_0 and S_1.

Algorithm 2 has the same approximation factor as the previous algorithm.

Theorem 3. *Let $\hat{T}$ be the tree returned by Algorithm 2, and let T^* be an optimal solution to the* CASCADEMLE *instance. Then $\hat{T}$ is consistent with S_0, S_1, and*

$$\mathrm{Cost}(T^*) \le \mathrm{Cost}(\hat{T}) \le 4 \ln |S_1| \cdot \mathrm{Cost}(T^*) \qquad (12)$$

The proof is similar to that for Theorem 2.

Experimental Results

Dataset and Methods

We experimentally study the CASCADEMLE problem and evaluate the performance of MINCOSTSTEINERTREE algorithm on several real-world and synthetic networks. The networks are listed in Table 1 and described here.

```
Algorithm 2: MINCOSTSTEINERTREE-OBS-UNINFECTED
Input: An undirected contact graph $G = (V, E)$ with edge probabilities $p_e$, set of observed uninfected nodes $S_0$, a set of observed infected nodes $S_1$
Output: Tree $T_r$ consistent with $S_0, S_1$.
for each edge $e$ do
    Compute the cost of inclusion $c_e = -\log p_e$ and cost of exclusion $d_e = -\log(1 - p_e)$
end for
Construct a node and edge-weighted graph $G'$ from $G$, by assigning weights as below:
for each node $u$ do
    $w(u) \leftarrow \sum_{e \in \mathcal{N}_e(u)} d_e$
end for
for each edge $e$ do
    $w(e) \leftarrow c_e - d_e$
end for
Remove all the nodes in $S_0$ (and their edges) from $G'$.
Convert $G'$ to a purely node-weighted graph $\hat{G}$ by splitting each edge with a new node having the same weight.
Find the minimum weighted Steiner tree $\hat{T}$ in $\hat{G}$ with terminal set $S_1$, using Klein and Ravi's (1995) algorithm for the node-weighted Steiner tree problem.
Let $r$ be any node in $\hat{T}$
return $\hat{T}$, with root $r$
```

1. arXiv High Energy Physics-Theory (HEP-TH): This is an academic collaboration network in the High Energy Physics-Theory community based on the citations in the arXiv preprints published between January 1993 and April 2004 (Gehrke, Ginsparg, and Kleinberg 2003; Leskovec and Krevl 2014). Taking the largest connected component, we generate a subgraph with $n = 500$ nodes, obtained by BFS starting from a random node. We refer to it as `arxiv`.
2. Erdős-Rényi random graphs: We generate several $G(n, q)$ graphs for evaluating the performance of our method and include results for $G(n = 300, q = 0.02)$.
3. Hospital ICU network: This is a contact network of patients and healthcare providers built using the Electronic Health Records (EHR) of the UVA Hospital's ICU between Jan 1, 2018 and Jan 8, 2018. We choose the largest connected component for our experiments and refer to it as `hospital-icu`.
4. Power-law networks: We generate power-law networks with $n = 1000$ nodes, varying the exponent γ in the range $[1.5, 3.5]$.

First, we study the error in approximation of the cost by comparing the true $\overline{\mathrm{Cost}}$ and our approximation Cost. The MINCOSTSTEINERTREE algorithm is evaluated with respect to network structure, diffusion model parameters, and the observation set. We compare our method against CULT (Rozenshtein et al. 2016) which is the state-of-the-art Steiner tree-based cascade reconstruction method. Since CULT takes an additional time-of-report information while

MinCostSteinerTree does not, we consider three variants of this method:

1. CuLT-Del: all nodes are reported as infected at the last time step,
2. CuLT-Rand: each node is reported as infected at a time step chosen randomly in between the time of infection and the last time step, and
3. CuLT-NoS: all nodes reported as infected at the time step of infection (NoS means No-Shift).

Note that these CuLT variants are ordered in the increasing amount of information provided to the algorithm.

We use the homogeneous probability setting for our experiments where we set the diffusion probability p across all edges to be the same. We generate the infection cascades under IC dynamics for a single source chosen uniformly at random. In our experiments for evaluating the algorithm, we have considered cascade sizes to be within $(0.02n, 0.1n)$, where n is the network size so that sufficient number of observed nodes can be extracted from the cascade.

Next, to create the observation node sets from the generated infection cascades, we use two different schemes:

(a) `random`: We randomly sample a fixed % of nodes from the infected node set to form the observed node set.

(b) `frontier`: Here, nodes in the cascade at a distance at least d from the source are chosen as observed. This corresponds to the scenario in which we have observed the more recent infections and our goal is to infer the rest.

These schemes are inspired from Rozenshtein et al. (2016), and can help evaluate the performance of our method in two distinct observational settings.

We choose *Matthews correlation coefficient* (MCC) (Matthews 1975) and F1-score as in (Rozenshtein et al. 2016; Jang et al. 2021), to evaluate the quality of the reconstructed cascades with the ground-truth. All reported values are averaged over 100 trials.

Results

Difference between $\mathrm{Cost}(T)$ and $\overline{\mathrm{Cost}}(T)$. We created several power-law networks on 1000 nodes for various values of the power-law exponent γ in the range $[1.5, 3.5]$. We generated cascades starting from a source chosen uniformly at random, varying the probability p from 0.05 to 0.49. For each such cascade, we computed the error between the two costs. Representative results are in Figure 3 (the results are consistent across replicates of the networks). We observe that the difference in the costs depends on both the probability p as well as the network structure (which is decided by γ). We recall that the difference between the two costs is $d = -\log(1-p)$ times the number of node pairs in the cascade that satisfy the property that both nodes are at equal distance from the source. For very low values of p, the difference is low across networks as the cost of including an edge $c = -\log(p) \gg d$. As p increases, we observe that the network structure comes into play. For very low values of the power-law exponent γ, there are several nodes with high degree leading to the presence of dense subgraphs. This increases the chances of node pairs where the nodes at equal distance from the source of the cascade, and in turn, leads to a larger difference in the two costs. On the other hand, for lower γ (2.5–3.5), the graph is more tree-like, and therefore, we see a very low error even for probability approaching 0.5. For $\gamma = 2$ in particular, we note that the error is 10% for p as high as 0.3.

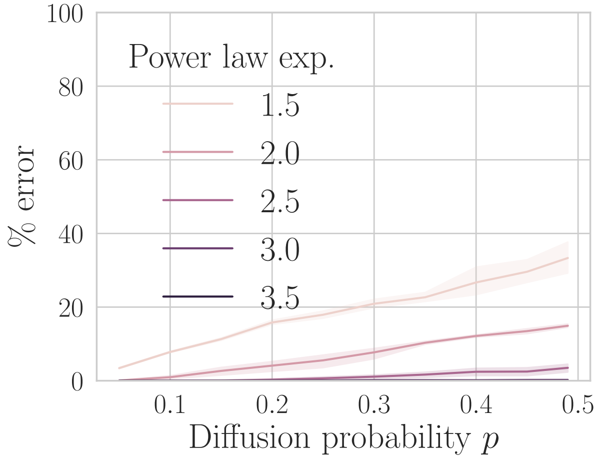

Figure 3: Percentage error between $\mathrm{Cost}(T)$ and $\overline{\mathrm{Cost}}(T)$ for cascades on random power-law graphs with varying diffusion probability p and power-law exponent.

Performance of MinCostSteinerTree. In Figure 4, the MCC scores are plotted for MinCostSteinerTree and CuLT for the two observation schemes and a diffusion probability of 0.1. We observe that MinCostSteinerTree performance is superior compared to CuLT-Del across observation schemes, networks and diffusion probabilities. We recall that the MinCostSteinerTree algorithm accounts for the diffusion model while the versions of CuLT do not. However, CuLT-NoS and, to some extent CuLT-Rand, account for the time of infection. In particular, for the $G(300, 0.02)$ graph and the `hospital-icu`, we observe that the performance of MinCostSteinerTree is much better than that of CuLT. We note that `arxiv` has a large average shortest path length (and low diameter) compared to the other two networks even though its clustering coefficient is large. Even though `hospital-icu` has a large clustering coefficient, it has a small average shortest path length like $G(300, 0.02)$. In Figure 5, we have representative plots of the MCC and F1-scores under diffusion probabilities 0.05 and 0.20. The performance is similar to that in Figure 4 for $G(300, 0.02)$ and `hospital-icu`. For the higher probability, we observe inferior performance in the case of `arxiv` as the distance from source increases under the `frontier` observation scheme.

Impact of different types of observations. For the `random` observation scheme, we observe that MinCostSteinerTree performance drastically increases with increase in the number of observed nodes. This is particularly true for the real-world networks. Typically, we see good per-

Graph Name	**Nodes**	**Edges**	**Clustering coefficient**	**Average shortest path length**
$G(n, q)$ random graph	300	897*	0.015	3.45
`arxiv`	500	895	0.52	12.5
`hospital-icu`	879	3575	0.59	4.31
Power-law networks	1000	660–6613	–	–

*Average value reported.

Table 1: Networks and their properties

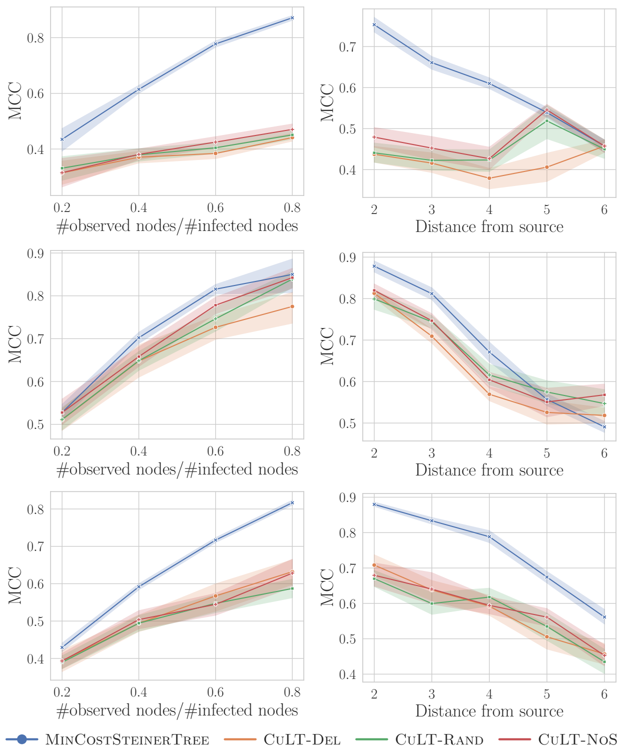

Figure 4: Performance of MinCostSteinerTree for `random` and `frontier` observation schemes. (a) $G(300, 0.02)$; (b) `arxiv`; and (c) `hospital-icu`, with fixed $p = 0.10$

formance when at least 40% of the infected nodes are observed. In the case of FRONTIER observation scheme, we observe that when the distance from the source is ≥ 4, the cascades constructed are quite inferior. This puts emphasis on early discovery of the outbreak.

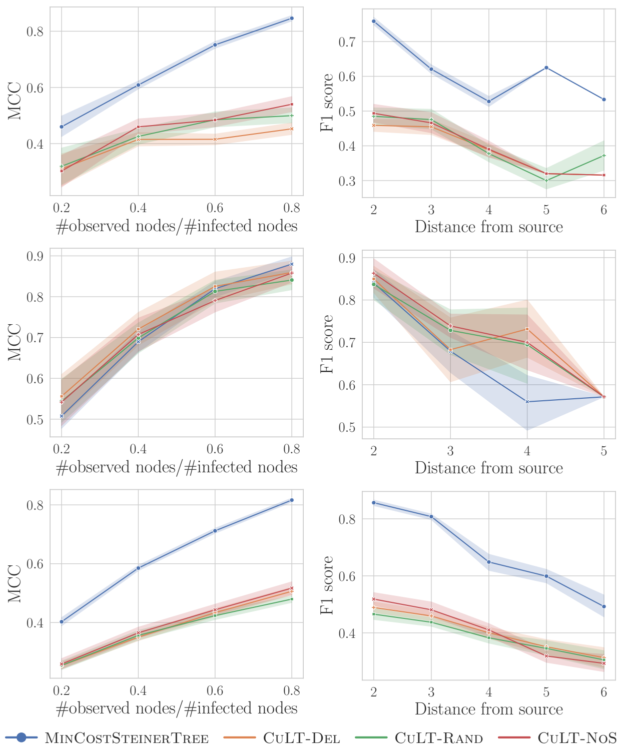

Figure 5: Performance of MinCostSteinerTree for (a) $G(300, 0.02)$ with $p = 0.05$; (b) `arxiv` with $p = 0.20$; and (c) `hospital-icu` with $p = 0.05$.

Conclusions

We studied the problem of reconstructing an epidemic cascade given a subset of infections as observed nodes under IC dynamics. We presented an algorithm with a logarithmic approximation factor using a node-weighted Steiner tree approach, and evaluated its performance on several synthetic and real-world networks. An important future direction is to extend our approach to reconstruct cascades resulting from more complex SEIR processes with delayed recovery, SI, and the SIS models. Another direction is to incorporate additional information about the cascade such as reporting time or order of infections that can help overcome the limits of the MLE problem studied here.

Acknowledgements

This work was supported in part by the United States Agency for International Development under the Cooperative Agreement no. AID-OAA-L-15-00001, Feed the Future Innovation Laboratory for Integrated Pest Management, Agricultural AI for Transforming Workforce and Decision Support (AgAID) grant no. 2021-67021-35344 from the USDA National Institute of Food and Agriculture, Network Models of Food Systems and their Application to Invasive Species Spread, grant no. 2019-67021-29933 from the USDA National Institute of Food and Agriculture, UVA, NSF grants Expeditions CCF-1918656, IIS-1955797 NIH 2R01GM109718-07, and CDC MIND U01CK000589.

References

Adcock, A. B.; Sullivan, B. D.; and Mahoney, M. W. 2013. Tree-Like Structure in Large Social and Information Networks. In *2013 IEEE 13th International Conference on Data Mining*, 1–10.

Gehrke, J.; Ginsparg, P. H.; and Kleinberg, J. M. 2003. Overview of the 2003 KDD Cup. *SIGKDD Explor.*, 5: 149–151.

Jang, H.; Pai, S.; Adhikari, B.; and Pemmaraju, S. V. 2021. Risk-aware Temporal Cascade Reconstruction to Detect Asymptomatic Cases : For the CDC MInD Healthcare Network. In *2021 IEEE International Conference on Data Mining (ICDM)*, 240–249.

Karp, R. M. 1972. Reducibility among Combinatorial Problems. In Miller, R. E.; Thatcher, J. W.; and Bohlinger, J. D., eds., *Complexity of Computer Computations. The IBM Research Symposia Series*, 85–103. Boston, MA: Springer US. ISBN 978-1-4684-2001-2.

Klein, P.; and Ravi, R. 1995. A Nearly Best-Possible Approximation Algorithm for Node-Weighted Steiner Trees. *Journal of Algorithms*, 19(1): 104–115.

Leskovec, J.; and Krevl, A. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. Accessed: 2022-7-25.

Matthews, B. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2): 442–451.

Robinson, A.; Walshe, T.; Burgman, M.; and Nunn, M. 2017. *Invasive Species: Risk Assessment and Management*. Cambridge University Press. ISBN 9781108158282.

Rozenshtein, P.; Gionis, A.; Prakash, B. A.; and Vreeken, J. 2016. Reconstructing an Epidemic Over Time. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1835–1844. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322.

Shah, D.; and Zaman, T. 2011. Rumors in a Network: Who's the Culprit? *Information Theory, IEEE Transactions on*, 57: 5163 – 5181.

Shaman, J.; et al. 2020. An estimation of undetected COVID cases in France. *Nature*, 590: 38–39.

Xiao, H.; Aslay, C.; and Gionis, A. 2018. Robust Cascade Reconstruction by Steiner Tree Sampling. In *2018 IEEE International Conference on Data Mining (ICDM)*, 637–646.

Zhu, K.; and Ying, L. 2014. Information source detection in the SIR model: A sample-path-based approach. *IEEE/ACM Transactions on Networking*, 24(1): 408–421.