Learning Network Graph of SIR Epidemic Cascades Using Minimal Hitting Set based Approach

Zhuozhao Li, Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: {zhuozhl, shenh}@clemson.edu

Kang Chen
Department of Electrical and Computer Engineering
Southern Illinois University, Carbondale, IL 62901
Email: kchen@siu.edu

***Abstract*—We consider learning the underlying graph structure of a network in which infection spreads based on the observations of node infection times. We give an algorithm based on minimal hitting set to learn the *exact* underlying graph structure and provide sufficient condition on number of cascades required (i.e. sample complexity) for reliable recovery, which is shown to be $\mathscr{O}(\log n)$, where n is the number of nodes in the graph. We then analytically evaluate performance of minimal hitting set approach in learning the degree distribution and detecting leaf nodes of a graph and provide a sufficient condition for its sample complexity which is shown to be lower than that of learning the whole graph. We also generalize the exact graph estimation problem to the problem of estimating the graph within a certain distortion, measured by edit distance. We show that this edit distance based graph estimator has a lower sample complexity. Our experimental results based on both synthetic network topologies and a real-world network trace show that our algorithm achieves superior performance than a previously proposed algorithm based on maximum likelihood.**

***Index Terms*—epidemic spread, infection, maximum hitting set, minimal hitting set**

I. Introduction

A great extent of work in literature focuses on deriving properties of the networks based on the knowledge of the underlying network graph structure [18], [19]. We consider the inverse problem in this paper, which learns the underlying network structure based on some observations of the network in the context of infection spread (as an example of epidemic cascades) over the network [16]. In addition to the infection spread, various phenomena can be modeled as epidemic cascades that have many applications in different disciplines. Examples include the diffusion of live streaming in peer-to-peer networks (e.g., CoolStreaming, PPLive and UUSee) [1], [11], [16], the propagation of ideas and information in social networks [8], the propagation of Internet worms and malwares (e.g., "Christma Exce", November's Internet Worm, Worm of December 1988) in computer population [10], and information disseminate in mobile ad-hoc networks and wireless sensor networks [9], [12].

We model the network as a weighted directed graph $G = (V, E)$, where V is a set of vertices and E is a set of edges connecting the vertices. We adhere to the Susceptible, Infected and Recovered (SIR) infection spread model in this paper. Initially at time $t = 0$, a subset of nodes (i.e., seeds) get infected. Each infected node at time $t = t_0$ succeeds in infecting its children with some unknown infection probability (i.e., weight of the corresponding edge). Each infected node becomes recovered and has no more effect on the epidemic process in the next time instance, $t = t_0 + 1$. From each cascade, we observe the vector of times that each node gets infected, while for the nodes that never get infected, we set their infection times to infinity. We use the term "sample" for the infection time vector of all nodes in one cascade. We are interested in learning the underlying graph G from a number of samples. It is unlikely to accurately learn the underlying graph through a single cascade even for simplest graphs with two nodes [16]. Therefore, in this paper, we characterize the number of samples (i.e., sample complexity) required to achieve high fidelity in learning the unknown topology of the network.

The maximum likelihood approach (ML) [16] can be used to recover G from infection times. The probability of observing each sample depends on a set of parameters for the weights of the graph edges (infection probabilities). Each set of parameters leads to an associated likelihood of generating the observed sample. ML picks a set of parameters that maximize the sum of associated likelihoods over all observed samples from cascades, then generates an edge in the estimated graph if the corresponding parameter of the edge is bigger than a pre-defined threshold.

However, ML is not sufficiently effective and efficient. First, ML guarantees to detect a subset of the parents of a node rather than its exact parental set because ML only detects a node's neighbors with edge weights higher than a pre-defined threshold. Second, ML requires a relatively high sample complexity for reliable graph recovery when the graph is dense (i.e., nodes have high degree). Third, failing to predefine a suitable threshold for the edge weight can lead to poor performance.

To overcome these problems, we propose a minimal hitting set approach [20] to recover G. A hitting set of a collection of sets is a set that intersects all of the sets in the collection. A minimal hitting set of the collection is a hitting set of the collection of sets and no proper subset of it is a hitting set. Learning the network graph is equivalent to finding parental set of each node of the graph. For each node $i \in V$, we estimate its parental set as follows. In cascade u, suppose the infection time of node i is t_i, we collect the set of nodes (denoted by S_i^u) that have infection time $t_i - 1$. The parental set of node

i is the minimal hitting set of the sets of S_i^u obtained from different cascades.

We have theoretically proved a sufficient condition on the number of samples for correct recovery using the minimal hitting set algorithm (MHS). Specifically, MHS has the following advantages over ML.
1) Unlike ML that has theoretical guarantees to detect a subset of parents of a given node, MHS has theoretical guarantees to detect exact neighbors of each node, and hence can recover the exact underlying graph structure.
(2) For dense networks with high-degree nodes (degree in this paper means indegree), minimal hitting set approach has lower sample complexity than ML.
(3) Unlike ML that depends on an appropriately predefined threshold parameter for high performance, MHS does not depend on such parameters as a priori.

Some applications may tolerate certain distortions in graph estimation and some applications may have limited number of samples. We can sacrifice exact recovery in order to reduce sample complexity. We measure the distortion by edit distance between the estimated graph and the true graph, which is defined as the number of edges in the symmetric difference between the two graphs' edge sets. We provide necessary condition for edit distance based graph estimator to have reliable recovery. We show that we can have lower sample complexity in recovering the graph up to a given edit distance.

The rest of the paper is organized as follows. Section II illustrates preliminaries of the problem. Section III presents our learning algorithms and provides the proof of our algorithms. Section IV shows the sufficient condition for exact graph recovery. Section V compares the sample complexity of our learning algorithms and the maximum likelihood algorithm [16]. Section VI provides necessary conditions for the edit distance based graph estimators over the collection of bounded degree graphs and collection of graphs with bounded total number of edges, respectively. Section VII presents experimental evaluation of our learning algorithm both on synthetic network and Google+ trace compared with ML. Section VIII presents a review of related work. Finally, Section IX concludes this work with remarks on our future work.

II. Preliminaries

A. System Model

We describe the system model [16] including the epidemic model and the observation model. For easy reference, Table I lists main notations used in this paper.

1) Epidemic Model: We model the network as a directed graph $G=(V,E)$, where vertices correspond to the nodes in the network and edges correspond to the links between nodes. For an directed edge $(j,i)\in E$, we say node j is a parent of node i. Let $\mathscr{V}_i := \{j : (j,i)\in E\}$ be the parental set of node i. There is no constraint on the number of parents of a node. We adopt the SIR model for the epidemic spread model, in which nodes can be in three states: susceptible, infected or recovered. Nodes that have not yet been infected are in susceptible state. Each susceptible node becomes infected if any of its infected parents infects it. Each infected node is in the infected status only for one time unit, after which it becomes recovered and cannot be infected again. Initially, at time $t=0$, each node independent of other nodes gets infected with probability p_{init}. Hence, the expected number of infected nodes at $t=0$ equals $n\cdot p_{init}$, where $n=|V|$ is the number of vertices in the graph. These initially infected nodes are called *seeds*. Learning approach in this paper is independent of the order of magnitude of number of seeds and our results can be applied to the infection models with a bounded number of seeds as well. Infections spread through the network in discrete times. In the infection spread in the network, each infected node i at time $t=k$ tries to infect each of its susceptible children j independently with probability p_{ij}. Node j that was susceptible at time $t=k$ gets infected at time $t=k+1$, if any of its infected parents at time $t=k$ infects it. Each infected node can be in infected status only for one time unit. In other words, parent i that was infected at time $t=k$ gets recovered at time $t=k+1$ and cannot either distribute infections anymore or get infected again. According to the above scenario, it is possible for some nodes not to get infection at all and stay in the susceptible status forever, while others have transmissions of susceptible $\rightarrow$ infected for one time step $\rightarrow$ recovered.

TABLE I: Notations

Parameter	Description
$G=(V,E)$	True underlying graph structure
$\mathscr{G}$	Collection of all possible graph structures
m	Number of cascades(samples)
δ	Specified error probability i
n	Number of nodes in the network
p_{min}, p_{max}	Lower and upper bounds on the probability that infected node succeeds in infecting its children)
u	Notation for cascade
t_i^u	Infection time of node i in cascade u
$\mathscr{G}_{n,d}$	Collection of all graphs over n vertices with maximum node degree being d
r	Allowed edit distance

There are other epidemic models as well in the literature such as SI and SIS model. In the SI model [19] once a node becomes infected remains infected forever. In the SIS model once a node is recovered it immediately becomes susceptible, therefore we have susceptible $\rightarrow$ infected $\rightarrow$ susceptible transition. We adhere to the described SIR model throughout the paper and consider learning problem over other epidemic models as our future work.

2) Observation Model: In each cascade u, seeds start to spread the infection throughout the network under the SIR model. We observe the time each node $i\in V$ gets infected, denoted by t_i^u. We set $t_i^u=0$ for seed nodes, and set $t_i^u=\infty$ for the nodes that never get infected in the cascade u. Then, for a cascade u, we have a vector of infection times $\mathbf{t}^u$ consisting of the infection time of each node (i.e., sample).

Having information only about infection times, we aim at finding the underlying graph structure of the network. This goal cannot be achieved through a single cascade, or equivalently through a single sample time observation vector $\mathbf{t}^u$ for a single cascade u even for simplest graphs [16]. Consequently, we independently run a set of cascades $\mathscr{U}$ with $|\mathscr{U}|=m$. Each cascade is assumed to be generated and observed as above independent of all others. The question we want to answer is: *what is the smallest m (i.e., sample complexity), such that our estimated graph structure is "correct" with high probability?* We consider two different notions of "correctness" in this paper as explained in Section II-B.

B. Graph Estimation

Let $\mathscr{G}$ be a collection of possible underlying graph structures and corresponding edge probabilities (p_{ij}). Given m sample vectors $\mathbf{T}^{\mathscr{U}} = \{\mathbf{t}^i,\ i = 1,...,m\}$ of observations from a true graph $G \in \mathscr{G}$, we consider two different graph estimators and corresponding notions of "correctness". Note that the vertex set V of G is known and we only aim at detecting existence of an edge between any two nodes $i, j \in V$. Based on our described model, there is an edge between arbitrary nodes i and j if and only if either $p_{ij} > 0$ or $p_{ji} > 0$. We also assume that for every edge, the edge probability p_{ij} is bounded by p_{min} and p_{max}, that is, $p_{min} \leq p_{ij} \leq p_{max}$.

1) Single Graph Estimator: Let $\hat{\mathscr{G}}(\mathbf{T}^{\mathscr{U}})$ be a graph estimator that takes as input observation of infection times and outputs a *single* graph. We define the probability of error of the single graph estimator as the probability that the estimated graph is not equal to the true graph, which also considers the randomness in choosing the true graph:

$$P_e(\hat{\mathscr{G}}(\cdot)) := \mathbb{P}[G \neq \hat{\mathscr{G}}(\mathbf{T}^{\mathscr{U}})]$$

Notice that G and $\hat{\mathscr{G}}(\mathbf{T}^{\mathscr{U}})$ can only differ in their edge sets, not in the vertex sets.

2) Edit Distance Based Graph Estimator: We consider estimating the true graph up to a given distortion measured by edit distance.

Definition 2.1: *Edit distance [3].* For any two graphs G and $\hat{G}$, edit distance between them $\Delta(G,\hat{G})$ is minimum number of edge deletions or insertions to convert G to $\hat{G}$. Thus $\Delta(G,\hat{G})$ can also be considered as the number of edges in the symmetric difference between edge sets of G and $\hat{G}$, i.e., $\Delta(G,\hat{G}) = |\{(E(G) - E(\hat{G})) \cup (E(\hat{G}) - E(G))\}|$.

Let $\hat{\mathscr{G}}_r(\mathbf{T}^{\mathscr{U}})$ be a graph estimator that takes as input observation of infection times and outputs a *single graph within edit distance* r of the true graph. We define the probability of error of the edit distance based graph estimator as the probability that the estimated graph is not within edit distance r of the true graph:

$$P_e(\hat{\mathscr{G}}_r(.)) := \mathbb{P}[\Delta(G,\hat{\mathscr{G}}_r(\mathbf{T}^{\mathscr{U}})) \geq r]$$

The controllable distortion on the estimation can reduce the sample complexity. There may exist applications where the observations of the network are limited. Then, we can first find out the graphs within edit distance r from the true graph, and search over these possible graphs to find the true graph.

III. Single Graph Estimator

A. Minimum Hitting Set based Approach

In this section, we introduce a minimum hitting set based approach [20] to recover the underlying graph $G = (V,E)$ of the network.

First, we introduce the definition of *minimum hitting set*. Given a collection C of subsets P ($C = \{P_1,...,P_n\}$), *hitting set H* is a set that intersects ("hits") all the subset in this collection with at least one element. In other words, every subset $P_i \in C$ must contain at least one element in the hitting set H. *Minimum hitting set* [20] is the hitting set of the smallest size. In each observation $\mathbf{t}^u \in \mathbf{T}^{\mathscr{U}}$ from cascade $u \in \mathscr{U}$, assume that the infection time of a node j is $t_j^u, \forall j \in V$. We further define S_j^u as the set of nodes i with $t_i^u = t_j^u - 1$, that is, the infection time of this set of nodes is $t_j^u - 1$,

$$S_j^u = \{i : t_i^u = t_j^u - 1\}.$$

Since there are different epidemic cascades, there are a collection of S_j^u sets, which are obtained from each cascade u.

Then, we propose a network graph recovery algorithm based on the minimum hitting set [20] for the single graph estimator. We can exactly recover the parental set of node j through finding the minimum hitting set of the collection of S_j^u collected from different observations, as m (i.e., the number of cascades) is sufficient. In the following, we provide the proof that the parental set of node j is the unique minimum hitting set of the collection of S_j^u obtained from every cascade, as the number of cascades increases (i.e., $m \to \infty$).

(i) We use $\mathscr{V}_j$ denote the parental set of node j in the graph $G = (V,E)$ of the network. First, each infected node must be infected by at least one of its parents, which means that in each cascade u, S_j^u must contain at least one element from the parental set of node j, that is, $\mathscr{V}_j \cap S_j^u \neq \emptyset$, and hence $\mathscr{V}_j$ is a hitting set of S_j^u.

(ii) Next, we prove that $\mathscr{V}_j$ is the unique minimum hitting set. If we assume that $\mathscr{V}_j$ is not the minimum hitting set, then there exists at least one different hitting set $|\hat{\mathscr{V}}_j|$, whose size is smaller than or equal to $\mathscr{V}_j$ (i.e., $|\hat{\mathscr{V}}_j| \leq |\mathscr{V}_j|$). Since $\hat{\mathscr{V}}_j$ is different from $\mathscr{V}_j$, there must exist at least one node k, which is in $\mathscr{V}_j$ but not in $\hat{\mathscr{V}}_j$, that is, $k \in \mathscr{V}_j \setminus \hat{\mathscr{V}}_j$. In this situation, we consider an event in a cascade u (denoted as event K) that only the parent node k of node j is a seed and only node j is infected. Obviously, this event occurs with positive probability. As $m \to \infty$, we know that this event will finally occur. Once this event occurs in one cascade u, it means that in this cascade u, none of the elements of $\hat{\mathscr{V}}_j$ hit S_j^u, i.e., $\hat{\mathscr{V}}_j \cap S_j^u = \emptyset$, which contradicts with assumption that $\hat{\mathscr{V}}_j$ is a hitting set.

Hence, we can conclude that such $k \in \mathscr{V}_j \setminus \hat{\mathscr{V}}_j$ should not exist. Therefore, there does not exist such $\hat{\mathscr{V}}_j$, and the parental set of node j is the unique minimum hitting set as $m \to \infty$.

In this session, we propose an algorithm to find the parental set of any node j (so the true graph) through finding the minimum hitting set of a collection of S_j^u sets, each of which is obtained from each observation $\mathbf{t}^u$. We also prove that the parental set of any node j is the unique minimum hitting set of the collection of S_j^u, as $m \to \infty$.

B. Minimal Hitting Set based Approach

Although we show that it is feasible to recover the graph using the minimum hitting set based algorithm, finding the minimum hitting set is NP-complete [20]. In order to solve this, we propose minimal hitting set based algorithm (MHS) in this section, which has polynomial runtime.

A hitting set of a collection of sets is minimal [6] if and only if no proper subset of it is a hitting set for this collection. Recall that *minimum hitting set* is the hitting set with the smaller size. For example, consider a collection of sets $\{\{1,2\},\{1,3\},\{1,2,4\},\{1,3,5\}\}$. Obviously, we see that $\{2,3\}$ is a minimal hitting set of the collection but not a minimum hitting set. On the other hand, $\{1\}$ is the minimum hitting set and also a minimal hitting set of the collection.

It is easy to prove that a minimum hitting set of a collection of sets is also a minimal hitting set, however the other way is not true. We assume that H is the minimum hitting set of a collection C. Say if H is not a minimal hitting set, then there exists a subset $H' \subset H$ that is a hitting set, according to the definition of minimal hitting set. Since $H' \subset H$, the size of H' is no greater than the size of H, i.e., $|H'| \leq |H|$, and hence H is not the hitting set of the smallest size, which contradicts with the definition of minimum hitting set. Therefore, a minimum hitting set of a collection of sets is also a minimal hitting set.

Now we propose the minimal hitting set based algorithm (MHS). We can exactly recover the parental set of a node $j, \forall j \in V$, by finding the minimal hitting set of the collection of S_j^u obtained from each cascade. The only difference between MHS and the minimum hitting set based algorithm is that we use the minimal hitting set of S_i^u ($\{S_i^u : u \in \mathscr{U}\}$) instead of the minimum hitting set of them. In the following, we prove that as $m \to \infty$, we can find the parental set of node j by finding the minimal hitting set of the collection of S_j^u.

For any node $j \in V$, we suppose that the estimated minimal hitting set is $\tilde{S}_j^{(m)}$ (after m observations) for node j. Now we discuss $\mathscr{V}_j$ and $\tilde{S}_j^{(m)}$. Since $\mathscr{V}_j$ is a minimum hitting set of the collection of S_j^u, we know that $\mathscr{V}_j$ is also a minimal hitting set of the collection of S_j^u.

(i) First, since $\mathscr{V}_j$ is a hitting set as mentioned in Section III-A and $\tilde{S}_j^{(m)}$ is a minimal hitting set, we conclude that $\mathscr{V}_j$ is not a proper subset of $\tilde{S}_j^{(m)}$ ($\mathscr{V}_j \not\subset \tilde{S}_j^{(m)}$), otherwise $\tilde{S}_j^{(m)}$ is not minimal according to the definition of minimal hitting set.

(ii) Second, if we assume that $\mathscr{V}_j$ is not exactly equal to $\tilde{S}_j^{(m)}$, as $\mathscr{V}_j$ is not a proper subset of $\tilde{S}_j^{(m)}$, then there must exist at least one node $k \in \mathscr{V}_j \setminus \tilde{S}_j^{(m)}$. However, recall in Section III-A that it has a positive probability for event K, i.e., node j is only infected by node k but not infected by any nodes in $\tilde{S}_j^{(m)}$ in an observation $\mathbf{t}^u$. Hence, as $m \to \infty$, event K finally occurs. Once event K occurs in one observation, it means that in this observation $\mathbf{t}^u$, $\tilde{S}_j^{(m)}$ does not intersect with S_j^u (i.e., $\tilde{S}_j^{(m)} \cap S_j^u = \emptyset$), which contradicts with the assumption that $\tilde{S}_j^{(m)}$ is a hitting set. Therefore, as $m \to \infty$, there does not exist any node k in $\mathscr{V}_j \setminus \tilde{S}_j^{(m)}$, which indicates that $\mathscr{V}_j \subseteq \tilde{S}_j^{(m)}$.

Combining (i) and (ii) that $\mathscr{V}_j \not\subset \tilde{S}_j^{(m)}$ and $\mathscr{V}_j \subseteq \tilde{S}_j^{(m)}$, we must have $\mathscr{V}_j = \tilde{S}_j^{(m)}$. Therefore, we demonstrate the effectiveness of MHS algorithm.

In this section, since finding the minimum hitting set is NP-complete [20], we propose MHS algorithm, which recovers the parental set of any node $j, j \in V$ by finding the minimal hitting set of a collection of S_j^u sets. We also prove that if $m \to \infty$, recovering the parental set of any node $j, j \in V$ is equivalent with finding the minimal hitting set of the collection of S_j^u.

IV. Lower Bounds

In this section, we turn our attention to establishing the lower bounds on the number of cascades that need to be observed for approximate network graph learning.

In Section III, we have proved that as $m \to \infty$, the parental set of node j, $\mathscr{V}_j$ can be recovered using the minimum hitting set algorithm and MHS algorithm. In the following, we further explore the lower bounds of the sample complexity for these two method, that is, *how many cascades do we need at least to recover the* ***exact*** *parental set of node j?*

A. Minimum Hitting Set based Algorithm

In this section, we explore the lower bound sample complexity for *minimum hitting set algorithm* to guarantee recovering the network graph with probability at least $1-\delta$, for any $\delta > 0$.

The procedure to find a lower bound of the sample complexity is similar to the Theorem 3 in [20]. For any node $j \in V$, we define the error event C_j as the event that the *estimated minimum hitting set* $\hat{S}_j^{(m)}$ is not equal to the parental set of node j, $\mathscr{V}_j$, after m observations. This only occurs when $|\hat{S}_j^{(m)}| \leq |\mathscr{V}_j|$, because otherwise $\mathscr{V}_j$ is a feasible hitting set with a smaller size. When $|\hat{S}_j^{(m)}| \leq |\mathscr{V}_j|$, there must exist $k \in \mathscr{V}_j \backslash \hat{S}_j^{(m)}$. Therefore, the probability of error is equivalent to the aggregate probability of such kind of node k, that is,

$$\begin{aligned} \mathbb{P}(\tilde{C}_j) &= p(\cup_{k \in \mathscr{V}_j} k \notin \tilde{S}_j^{(m)}) \\ &\leq \sum_{k \in \mathscr{V}_j} p(k \notin \tilde{S}_j^{(m)}). \end{aligned}$$

Let us consider the probability that a parent node k of node j does not succeed to infect node j in m observations. In this case, we have $k \notin \tilde{S}_j^{(m)}$. Obviously, in one cascade u, the probability that node k succeeds to infect node j (denoted as p_e) follows $p_e \geq p_{init} * p_{kj}$, where $p_{init} * p_{kj}$ is corresponding to the probability that node k is a seed and it succeeds to infect node j. Recall that p_{kj} is bounded by p_{min} and p_{max}, that is, $p_{min} \leq p_{kj} \leq p_{max}$. Therefore,

$$\begin{aligned} \mathbb{P}(\tilde{C}_j) &\leq \sum_{k \in \mathscr{V}_j} p(k \notin \tilde{S}_j^{(m)}) \leq \sum_{k \in \mathscr{V}_j} (1-p_e)^m \\ &\leq \sum_{k \in \mathscr{V}_j} (1 - p_{init} * p_{kj})^m \\ &\leq d_j (1 - p_{init} * p_{min})^m, \qquad (1) \end{aligned}$$

where d_j is the degree of node j. From Formula (1), we notice that as $m \to \infty$, $\mathbb{P}(\tilde{C}_j)$ is upper bounded by a value that is close to zero, which indicates that $\mathbb{P}(\tilde{C}_j) \to 0$. It means that there is no error as $m \to \infty$, and hence it again proves that the parental set of node j is the unique minimum hitting set of the collection of S_j^u obtained from every observation. Finally, the probability of the estimated graph $\hat{G}_H$ not equal to the true graph G is:

$$\begin{aligned}\mathbb{P}(\hat{G}_H \neq G) &= \mathbb{P}(\cup_j C_j) \leq \sum_{j\in V(G)} \mathbb{P}(C_j) \\ &\leq \sum_{j\in V(G)} d_j(1-p_{init} * p_{min})^m \\ &\leq n^2(1-p_{init} * p_{min})^m. \qquad (2)\end{aligned}$$

Note that in Formula (5), we use the fact that $\sum_{j\in V(G)} d_j \leq n^2$ for any graph. Obviously, in order to guarantee that $\mathbb{P}(\hat{G}_H \neq G_H) < \delta$, the sufficient condition is to guarantee $n^2(1-p_{init} * p_{min})^m < \delta$. Therefore, the lower bound sample complexity of minimum hitting set algorithm is,

$$m \geq \frac{\log\delta - 2\log n}{\log(1-p_{init} * p_{min})}. \qquad (3)$$

Therefore, from Formula 3, the sample complexity for *minimum hitting set algorithm* to guarantee that the estimated graph equals the true graph with probability at least $1-\delta$ is $\mathscr{O}(\log n)$.

B. Minimal Hitting Set based Algorithm

In this section, we aim to find the lower bound sample complexity for *minimal hitting set algorithm* to guarantee recovering the network graph with probability at least $1-\delta$, for any $\delta > 0$.

For any node $j \in V$, we define the error event C_j as the event that the *estimated minimal hitting set* $\hat{S}_j^{(m)}$ is not equal to the parental set of node j, $\mathscr{V}_j$, after m observations. Since $\tilde{S}_j^{(m)}$ is minimal hitting set and $\mathscr{V}_j$ is a hitting set, $\mathscr{V}_j$ is not a proper subset of $\tilde{S}_j^{(m)}$, otherwise $\tilde{S}_j^{(m)}$ is not minimal. On the other hand, the upper bound of Formula (1) is still valid for $\mathbb{P}(\tilde{C}_j)$. Therefore,

$$\begin{aligned}\mathbb{P}(\tilde{C}_j) &= p(\cup_{k\in\mathscr{V}_j} k \notin \tilde{S}_j^{(m)}) \leq \sum_{k\in\mathscr{V}_j} p(k \notin \tilde{S}_j^{(m)}) \\ &\leq \sum_{k\in\mathscr{V}_j} (1-p_e)^m \leq \sum_{k\in\mathscr{V}_j} (1-p_{init} * p_{min})^m \\ &\leq d_j(1-p_{init} * p_{min})^m. \qquad (4)\end{aligned}$$

Again, we see from Formula (4) that, as $m \to \infty$, $\mathbb{P}(\tilde{C}_j) \to 0$, which again demonstrates that we can find the parental set of node j by finding the minimal hitting set of the collection of S_j^u obtained from every cascade.

The rest of the procedure to explore the lower bound sample complexity for MHS is similar as Formula (5).

$$\begin{aligned}\mathbb{P}(\hat{G}_H \neq G) &= \mathbb{P}(\cup_j C_j) \leq \sum_{j\in V(G)} \mathbb{P}(C_j) \\ &\leq \sum_{j\in V(G)} d_j(1-p_{init} * p_{min})^m \\ &\leq n^2(1-p_{init} * p_{min})^m. \qquad (5)\end{aligned}$$

In order to guarantee recovery of the exact graph with a probability at least $1-\delta$, the lower bound sample complexity for MHS is the same as Formula (3),

$$m \geq \frac{\log\delta - 2\log n}{\log(1-p_{init} * p_{min})}. \qquad (6)$$

Therefore, from Formula 6, the lower bound of the sample complexity for MHS to guarantee that the estimated graph equals the true graph with probability at least $1-\delta$ is $\mathscr{O}(\log n)$.

V. Performance Comparison Discussion

(i) **Minimum hitting set algorithm and maximum likelihood approach [16]**. In the following, we discuss the advantages of the minimum hitting set algorithm over the maximum likelihood approach (ML) proposed in [16].

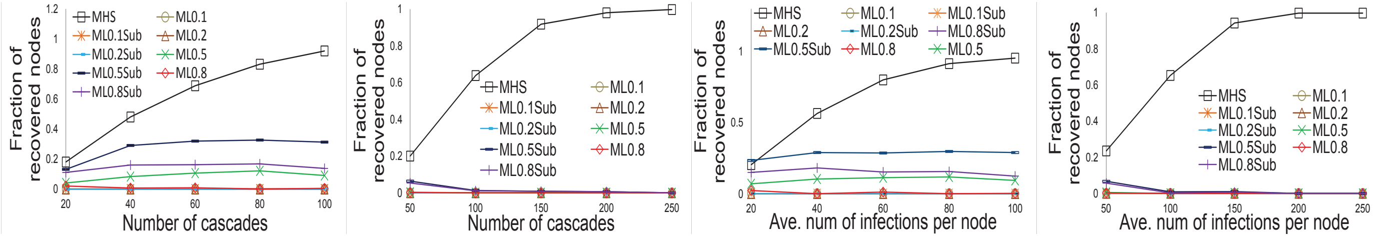
(1) Based on the proof in Section III-A, as the number of cascades goes to infinity, we are able to recover the exact parental set of node $j, \forall j \in V$ in the graph $G = (V, E)$. However, ML [16] can only guarantee that the estimated parental set of node j has no false neighbors and provide theoretical guarantees on recovering strong neighbors of node i, which means that not all the parent nodes of node i are recovered.

(2) According to Formula 3, the number of cascades needed to recover the exact graph is $\mathscr{O}(\log n)$. However, the number of cascades needed for ML [16] to learn the true graph or equivalently for each node i, to learn its parental set is $\mathscr{O}(d_i^2 \log|\mathbb{S}_i|)$, where d_i is degree of node i and $|\mathbb{S}_i|$ is super graph degree. The super graph of node i contains its true parents, i.e., $\mathscr{V}_i \subset \mathbb{S}_i$. For network topologies without the bounded degree assumption, we may have $d_i = \mathscr{O}(n)$, which leads to sample complexity of $\mathscr{O}(n^2\log(n))$ for ML at the worst case. In current era of big data, more users involve in the network and hence the number of nodes increases (i.e., $n \to \infty$). As a result, the minimum hitting set algorithm outperforms ML for dense graphs with high degree nodes.

(3) For the edge probability, ML has a threshold parameter, which is not known a priori. Inappropriate choice of this parameter can lead to poor performance of ML on recovering the graph, that is, increasing the probability of errors. A large value parameter setting leads to the recovery of only some parents for each node recovered (i.e., false negatives), while a small value parameter setting leads to many false parents recovery for each node (i.e., false positives). However, our proposed minimum hitting set algorithm does not need any priori knowledge. In Section VII, our experiments also verify that the inappropriate setting of the parameter causes poor performance.

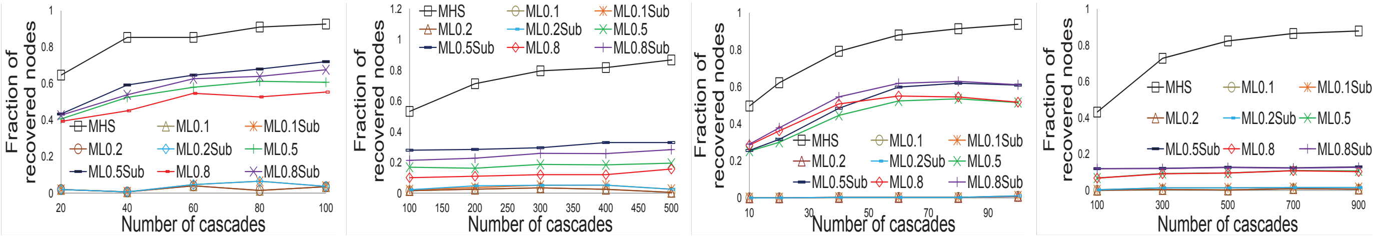
(ii) **Minimum hitting set algorithm and Minimal hitting set algorithm.** We compare the performance of the minimal and minimum hitting set based algorithms in two aspects: (1) sample complexity of the algorithm; and (2) running time of algorithm. From Formulas 3 and 6, we see that the the two algorithms have the same sample complexities (i.e., $\mathscr{O}(\log n)$). However, these two algorithms have different running time. Since minimum hitting set of a collection of sets is also minimal hitting, hence it is obvious that minimal hitting set algorithm has better time complexity than minimum hitting set algorithm to find the true graph.

Moreover, it is known that finding the minimum hitting set of a collection of sets is NP-hard [20], however, we can find a minimal hitting set of the collection in polynomial time. For



(a) Recovery of 5×5 grids (b) Recovery of 10×10 grids (c) Recovery of 5×5 grids (d) Recovery of 10×10 grids

Fig. 1: Single graph estimation of grids.



(a) 25 nodes, max. degree=4 (b) 25 nodes, max. degree=9 (c) 100 nodes, max. degree=4 (d) 100 nodes, max. degree=9

Fig. 2: Single graph estimation of random graphs with different number of cascades.

example, one approach may be first considering the union of the sets, and starting from the smallest element, we remove it if the union without that element is still a hitting set, then go to the next element. It is clear that the resulting set is minimal hitting set and that this procedure has polynomial time complexity.

VI. Edit Distance Based Graph Estimator

Using the minimum hitting set algorithm (or the MHS algorithm), we can get an estimated graph that has a certain edit distance r with the true graph. In this section, we consider a collection of possible graphs $\mathscr{G}$, rather than a single graph as Section III. We consider $\mathscr{G}_{n,d}$, the collection of all graphs on n vertices with maximum degree of each vertex being d. For this collection of possible graphs, we show the following necessary conditions for edit distance based graph estimator, which is not considered in previous network learning literature.

Recall the definition in Section II-B that the probability of error of the edit distance based graph estimator as the probability that the estimated graph is not within edit distance r of the true graph:

$$P_e(\hat{\mathscr{G}}_r(.)) := \mathbb{P}[\Delta(G, \hat{\mathscr{G}}_r(\mathbf{T}^{\mathscr{U}})) \geq r].$$

Necessary Condition for Collection $\mathscr{G}_{n,d}$:

We denote that $\mathscr{G}_{n,d} = \{G_1, ..., G_M\}$ is the collection of graphs on n vertices with maximum degree of each vertex being d. Suppose G is chosen uniformly at random from $\mathscr{G}_{n,d}$. In the following, we derive the necessary condition for the graph estimator, that is, if the number of samples m is less than a number, the probability of error satisfies $P_e^{(m)} \to 1$, as $n \to \infty$. We derive that if the number of samples m satisfies:

$$m < \frac{\frac{nd}{4}\log\frac{n}{8d} - \log(r\binom{\frac{n^2}{2}}{r})}{n}, \qquad (7)$$

then for any arbitrary graph estimator, its probability of error satisfies $P_e^{(m)} \to 1$ as $n \to \infty$. Due to the space limitation, we do not provide the details of the derivation here. Similar derivation can be referred to [3].

Summary

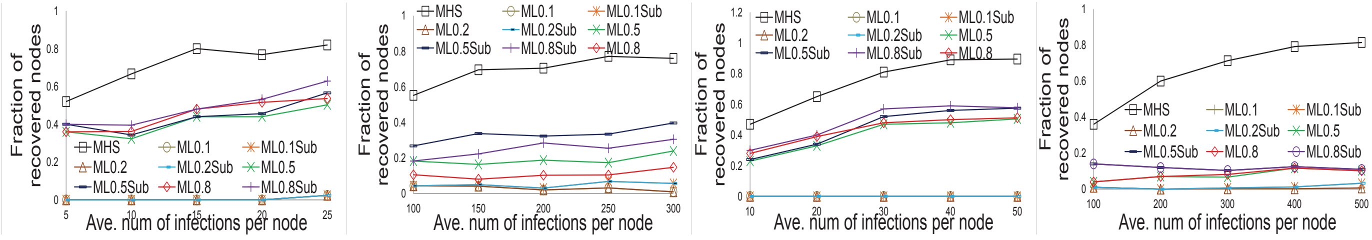
From Formula 7, the necessary condition for the sample complexity is:

$$\begin{aligned} \frac{\frac{nd}{4}\log\frac{n}{8d} - \log(r\binom{\frac{n^2}{2}}{r})}{n} &= \frac{d}{4}\log\frac{n}{8d} - \frac{\log(r\binom{\frac{n^2}{2}}{r})}{n} \\ &\approx \mathscr{O}(\log n) - \frac{\log(r\binom{\frac{n^2}{2}}{r})}{n} \\ &< \mathscr{O}(\log n) \end{aligned}$$

We see that it is less than $\mathscr{O}(\log n)$, which indicates that the edit distance based graph estimator has lower sample complexity than the single graph estimator.

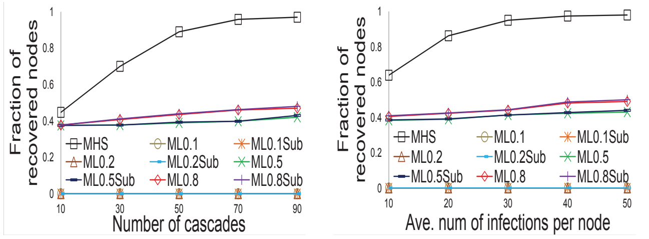
VII. Performance Evaluation

In this section, we validate the performance of the Minimal Hitting Set algorithm (MHS) in comparison with the Maximum Likelihood algorithm (ML) [16] through experimental evaluations on grids, random graphs and subgraphs of the Google+ real world trace. We set $p_{init} = 0.3$ to initialize random seeds and the probability that each active node succeeds to infect its children was set to 0.8. We set the predefined infection probability threshold in ML to x=0.1, 0.2, 0.5 and 0.8, respectively, and use MLx (e.g., ML0.1) to denote corresponding method. We run each experiment for 20 times and report the average experimental result. We measured fraction of recovery as the fraction of the nodes in the graph, whose exact parental set was detected; that is, the set of estimated parents is exactly the same to the set of true parents. We also measured the fraction of nodes that an algorithm recovers a subset of their parental sets; that is, the set of estimated parents is a subset of true parents. These results in the figures are denoted by MLxSub.



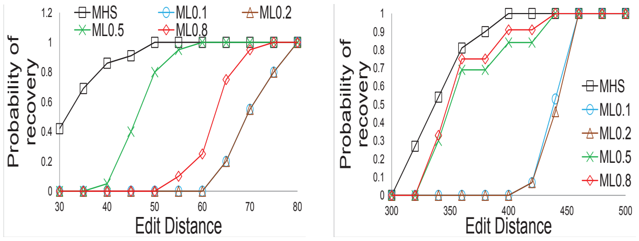
(a) 25 nodes, max. degree=4 (b) 25 nodes, max. degree=9 (c) 100 nodes, max. degree=4 (d) 100 nodes, max. degree=9

Fig. 3: Single graph estimation of random graphs with different average number of infections per node.



(a) Different number of cascades (b) Different number of average infections per node

Fig. 4: Single graph estimation of a sub-graph of Google+.



(a) 5×5 grids (b) 10×10 grids

Fig. 5: Edit distance based estimation of grids with varying edit distance requirements.

A. *Single Graph Estimation*

Grids. We tested on 5×5 and 10×10 grids. Figures 1(a), 1(b), 1(c), 1(d) show the fraction of recovered nodes versus the number of cascades, and the average number of infections per node. As evident from Figures 1(a) and 1(b), in all methods, as the number of cascades increases, the fraction of recovery increases though it is not obvious for ML and ML-Sub. Also, we observe that a larger scale network (10×10 grid compared to 5×5 grid) needs more cascades to have comparable reliable graph recovery. We make the same observations from Figures 1(c) and 1(d). In all figures, MHS produces a higher fraction of recovered nodes than ML and ML-Sub, which validates the superiority of MHS over ML. Also, ML-Sub gives a higher fraction of recovered nodes than ML because ML-Sub recovers a subset of each node's parental set while ML recovers the exact set of each node's parental set. Comparing ML with different thresholds, we note that the 0.5 threshold gives better performance than other thresholds. Inappropriate selection of the threshold parameter can lead to a poor performance of ML. A too small threshold (e.g., 0.1 and 0.2) would lead to many false positives, while a too large threshold (e.g., 0.8) would lead to many false negatives. The results indicate ML's drawback of setting a pre-define threshold in graph recovery.

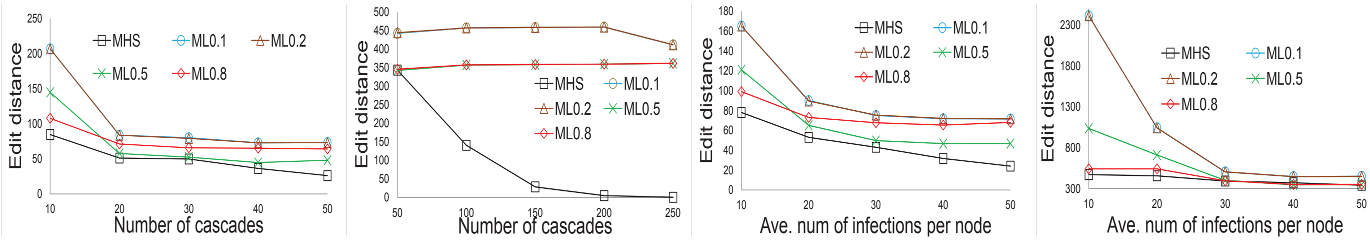
Random Graphs. We tested on random graphs with different combinations on a scale (25 and 100 nodes) and a maximum degree (4 and 9). Figures 2 and 3 show the fraction of recovered nodes versus the number of cascades and the average number of infections per node, respectively. In Figures 2, for all methods, a larger number of cascades produce a higher fraction of recovery generally. We also observe that the number of cascades needed to obtain a given fraction of recovery increases as the network scale increases. Finally, for a given number of nodes, a graph with a higher maximum node degree requires more cascades to achieve a given fraction of recovered nodes. The same results hold in Figures 3 with a varying average number of infections per node. From both sets of figures, we see that MHS produces a higher fraction of recovered nodes than ML, which validates its superiority over ML. Also, ML-Sub gives a higher fraction of recovered nodes than ML because recovering a subset of the graph is less constraint than recovering the whole graph. Also, note that the thresholds 0.5 and 0.8 give better performance than other threshold values.

Google+. We then tested on a sub-graph of the Google+ network with 500 users. Figures 4(a) and 4(b) plot the fraction of nodes that the algorithms can recover their true parents versus the number of cascades and the average number of infections per node, respectively. For all methods, increasing the number of cascades leads to an increase in the fraction of recovery. MHS has better performance than ML as it produces a higher fraction of recovered nodes. Also ML-Sub generates a higher fraction of recovered nodes than ML. Note that setting the threshold of ML to 0.8 gives better performance than other thresholds, and ML0.1 and ML0.2 significantly degrades the performance of ML.

B. *Edit Distance based Graph Estimation*

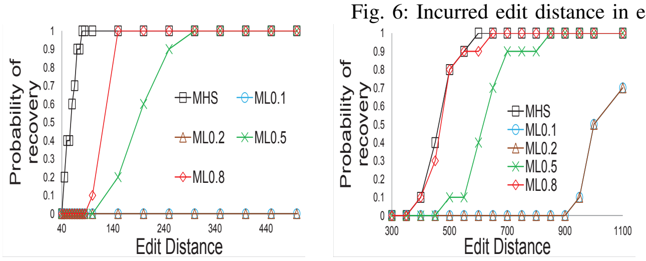
In this experiment, we measure two metrics. One metric is *probability of recovery* which measures the fraction of recovered graphs that are within a certain edit distance r of the true graph. The second metric is the average of edit distances of all recovered graphs from the true graph. Unless otherwise indicated, the number of cascades was set to 50.

Grids. Figures 5 and 6 show the results of experiments for 5×5 and 10×10 grids. Figures 5(a) and 5(b) show that a large network scale requires much higher edit distance to achieve the same probability of recovery for the same number of cascades. We also find that MHS produces a higher fraction of recovery for the same edit distance compared to ML, and ML0.5 has a better performance than other thresholds. All



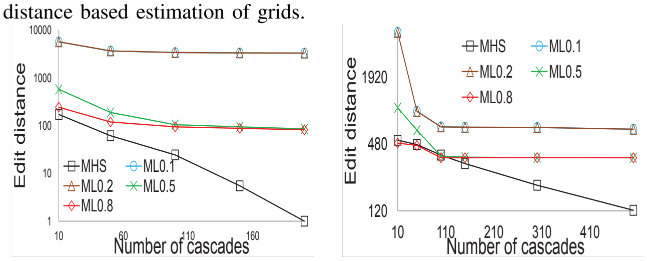
(a) 5×5 grids (b) 10×10 grids (c) 5×5 grids (d) 10×10 grids

Fig. 6: Incurred edit distance in edit distance based estimation of grids.



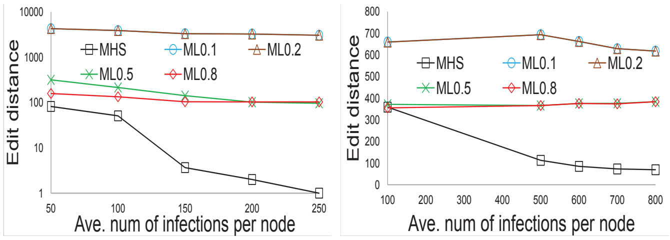
(a) 100 nodes, max degree=4 (b) 100 nodes, max degree=9

Fig. 7: Edit distance based estimation of random graphs with varying edit distance requirements.



(a) 100 nodes, max degree=4 (b) 100 nodes, max degree=9

Fig. 8: Incurred edit distance in edit distance based estimation of random graphs with different number of cascades.



(a) 100 nodes, max degree=4 (b) 100 nodes, max degree=9

Fig. 9: Incurred edit distance in edit distance based estimation of random graphs with different average infections per node.

figures in Figure 6 show that increasing number of cascades (or average number of infections per node) leads to lower edit distance, and that MHS achieved lower edit distance than ML.

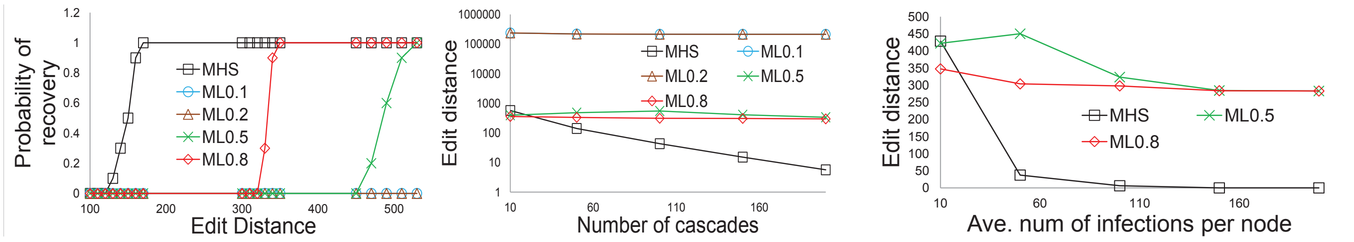
Random Graphs. We consider spreading infections on random graphs with 100 nodes and show the experimental results in Figures 7, 8 and 9. For a fixed number of cascades, we observe that a larger network scale requires a higher allowed edit distance to achieve the given probability of recovery. Finally, for the given number of nodes and the number of cascades, a graph with a lower maximum node degree has a higher probability of recovery for the same edit distance. We further observe that for a fixed edit distance, MHS recovers the graph with a higher probability compared to ML, and the 0.8 and 0.5 thresholds result in better ML performance than other thresholds. Finally, we see that the incurred edit distance decreases as the number of cascades (or average number of infections per node) increases, and that MHS produces better performance than ML.

Google+. Figure 10 shows the experimental results for a subset of Google+ with 500 nodes. MHS performs better than ML with a higher probability of recovery and ML0.8 performs better than ML0.1, ML0.2 and ML0.5. Increasing the number of cascades (or average number of infections per node) leads to lower incurred edit distance. Also, MHS incurs the lowest edit distance compared to ML and ML0.8 achieves a lower edit distance than ML0.1,ML0.2 and ML0.5. From these observations, we conclude that MHS has better performance than ML.

VIII. Related Work

A number of previous works aim to find infection/information sources [18], [19]. Shah *et al.* [18], [19] modeled the virus spread as a variant of SIR model, constructed a maximum likelihood estimator based on rumor centrality for a class of graphs. Chalermsook *et al.* [2] modeled the viral marketing for online advertising as a model of influence spread across the social network. They proposed a polynomial time approximation algorithm to place the seeds for the advertiser to maximize the revenue of the social network provider. Massoulie *et al.* [13] proposed Greedy-Bayes, an algorithm to select the users who are interested in the news whose topic is yet unknown, while not spamming too many uninterested users. Fanti *et al.* [4] presented a message protocol that spreads the message fast and perfectly hide the source in a tree network for anonymous messaging platforms. Our work differs from the above works that we aim to recover the exact graph.

Many works focus on graph learning in epidemic cascades based on different kinds of information [5], [15]–[17]. Netrapalli *et al.* [16] analyzed the sample complexity of graph learning in epidemic cascades solely based on infection times. They proposed an algorithm based on maximum likelihood and compared it with the greedy algorithm. Their learning algorithm finds an approximate graph structure, while we propose a MHS algorithm that finds the exact graph structure. Rabbat *et al.* [17] considered inferring network structure from "co-occurrence" data. They modeled co-occurrence observations as independent realizations of a random walk on the network and derived an expectation-maximization to estimate the random walk parameters. Gripon *et al.* [7] proposed an algorithm for exact graph recovery from indirect observations. Each observation is the unordered set of nodes that are activat-



(a) Fraction of recovery (b) Incurred edit distance vs. number of cascades (c) Incurred edit distance vs. average infections per node

Fig. 10: Edit distance based estimation of a sub-graph of Google+.

ed along a path through the network. They provided necessary and sufficient conditions for the reconstruction algorithm. Milling *et al.* [15] considered diagnosing the causative network of an epidemic or contagion using noisy and highly incomplete data. Milling *et al.* [14] provided sufficient conditions for different graph topologies so that it is possible to distinguish a random infection model from a spreading epidemic model with asymptotically zero probability of misclassification. In contrast to the above works, our work is the first that proposes a *polynomial time algorithm* to recover the underlying structure of the network based on infection times of the nodes.

Yang *et al.* [20] used the minimum hitting set algorithm to learn the interference graph of a wireless network based on passive traffic monitoring. They provided both necessary and sufficient conditions on the number of samples required for the learning problem in both static networks and time-varying networks. Their minimum hitting set approach cannot be directly applied to our scenario due to the environmental differences including assumptions on edge probabilities and nodal degree which results in different bounding procedure. Due to different setup of the problems, we relax some of the constrained assumptions in [20] and use different bounding procedures to derive the sufficient condition in Formula 3, which leads to different results. In particular, they assumed that each sensor node has bounded number of direct and hidden interferers. We do not have such bound assumption on node degrees, which is more practical for dense networks. Since finding minimum hitting set is NP-hard, we use minimal hitting set (MHS) that has a polynomial runtime.

IX. Conclusions and Future Work

We consider learning the underlying graph structure of an epidemic cascade based on infection times of nodes. For a cascade in which a given node is infected, we consider a set of nodes with infection time one less than the infection time of the node. We estimate parental set of the node as the minimal hitting set of these sets. We show that the estimation error probability goes to zero as we increase the number of cascades and derive the sufficient condition for its sample complexity. We then consider learning the graph up to a given edit distance and provide a necessary condition on the sample complexity. However, our algorithms require the knowledge of infection times of all nodes in the network, which may not be practical in real scenarios due to hidden and missing data. We will study learning the graph structure with missing infection times in our future work.

Acknowledgements

We would like to thank Mr. Nematollah Iri for his insightful advise and the anonymous reviewers for their valuable feedback. This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, IBM Faculty Award 5501145 and Microsoft Research Faculty Fellowship 8300751.

References

[1] T. Bonald, L. Massoulie, F. Mathieu, D. Perino, and A. Twigg. Epidemic Live Streaming: Optimal Performance Trade-Offs. In *Proc. of SIGMETRICS*, 2008.

[2] P. Chalermsook, A. D. Sarma, A. Lall, and D. Nanongkai. Social network monetization via sponsored viral marketing. In *Proc. of SIGMETRICS*, 2015.

[3] A. K. Das, P. Netrapalli, S. Sanghavi, and S. Vishwanath. Learning Markov Graphs Up To Edit Distance. In *Proc. of IEEE ISIT*, 2012.

[4] G. Fanti, P. Kairouz, S. Oh, and P. Viswanath. Spy vs. spy: Rumor source obfuscation. In *Proc. of SIGMETRICS*, 2015.

[5] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *TKDD*, 5(4), 2012.

[6] R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41:79–88, 1989.

[7] V. Gripon and M. Rabbat. Reconstructing a graph from path traces. *CoRR*, abs/1301.6916, 2013.

[8] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information Diffusion Through Blogspace. In *Proc. of WWW*, 2003.

[9] K. Han, L. Xiang, J. Luo, M. Xiao, and L. Huang. Energy-Efficient Reliable Data Dissemination in Duty-Cycled Wireless Sensor Networks. In *Proc. of MobiHoc*, 2013.

[10] J. O. Kephart and S. R. White. Directed-Graph Epidemiological Models of Computer Viruses . In *Proc. of IEEE Symposium on Security and Privacy*, 1991.

[11] Y. Lin and H. Shen. Cloud fog: Towards high quality of experience in cloud gaming. In *Proc. of ICPP*, 2015.

[12] J. Liu, L. Yu, H. Shen, Y. He, and J. Hallstrom. Characterizing data deliverability of greedy routing in wireless sensor networks. In *Proc. of SECON*, 2015.

[13] L. Massoulié, M. I. Ohannessian, and A. Proutière. Greedy-bayes for targeted news dissemination. In *Proc. of SIGMETRICS*, 2015.

[14] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai. Network Forensics: Random Infection vs Spreading Epidemic. In *Proc. of SIGMETRICS*, 2012.

[15] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai. Detecting Epidemics Using Highly Noisy Data. In *Proc. of MobiHoc*, 2013.

[16] P. Netrapalli and S. Sanghavi. Learning the Graph of Epidemic Cascades. In *Proc. of SIGMETRICS*, 2012.

[17] M. G. Rabbat, M. A. T. Figueiredo, and R. D. Nowak. Network Inference from Co-Occurrences. In *Advances in Neural Information Processing Systems 19*, 2007.

[18] D. Shah and T. Zaman. Detecting Sources of Computer Viruses in Networks: Theory and Experiment. In *Proc. of SIGMETRICS*, 2010.

[19] D. Shah and T. Zaman. Rumor Centrality: A Universal Source Detector. In *Proc. of SIGMETRICS*, 2012.

[20] J. Yang, S. C. Draper, and R. Nowak. Passive learning of the interference graph of a wireless network. In *Proc. of IEEE ISIT*, 2012.