

Finding the Graph of Epidemic Cascades

Praneeth Netrapalli
praneethn@utexas.edu

Sujay Sanghavi
sanghavi@mail.utexas.edu

Abstract

We consider the problem of finding the graph on which an epidemic cascade spreads, given *only* the times when each node gets infected. While this is a problem of importance in several contexts – offline and online social networks, e-commerce, epidemiology, vulnerabilities in infrastructure networks – there has been very little work, analytical or empirical, on finding the graph. Clearly, it is impossible to do so from just one cascade; our interest is in learning the graph from a small number of cascades.

For the classic and popular “independent cascade” SIR epidemics, we analytically establish the number of cascades required by both the global maximum-likelihood (ML) estimator, and a natural greedy algorithm. Both results are based on a key observation: the global graph learning problem decouples into n local problems – one for each node. For a node of degree d , we show that its neighborhood can be reliably found once it has been infected $O(d^2 \log n)$ times (for ML on general graphs) or $O(d \log n)$ times (for greedy on trees). We also provide a corresponding information-theoretic lower bound of $\Omega(d \log n)$; thus our bounds are essentially tight. Furthermore, if we are given side-information in the form of a super-graph of the actual graph (as is often the case), then the number of cascade samples required – in all cases – becomes independent of the network size n .

Finally, we show that for a very general SIR epidemic cascade model, the Markov graph of infection times is obtained via the moralization of the network graph.

Keywords: Epidemics, cascades, network inverse problems, structure learning, sample complexity, Markov random fields

1 Introduction

Cascading, or epidemic, processes are those where the actions, infections or failure of certain nodes increase the susceptibility of other nodes to the same; this results in the successive spread of infections / failures / other phenomena from a small set of initial nodes to a much larger set. Initially developed as a way to study human disease propagation, cascade or epidemic processes have recently emerged as popular and useful models in a wide range of application areas. Examples include

(a) *social networks*: cascading processes provide natural models for understanding both the consumption of online media (e.g. viral videos, news articles[13]) and spread of ideas and opinions (e.g. trending of topics and hashtags on Twitter/Facebook[24], keywords on blog networks[7])

(b) *e-commerce*: understanding epidemic cascades (and, in this case, finding influential nodes) is crucial to viral marketing [9], and predicting/optimizing uptake on social buying sites like Groupon etc.

(c) *security and reliability*: epidemic cascades model both the spread of computer worms and malware [10], and cascading failures in infrastructure networks [11, 23] and complex organizations [18].

(d) *peer-to-peer networks*: epidemic protocols, where users sending and receiving (pieces of) files in a

random uncoordinated fashion, form the basis for many popular peer-to-peer content distribution, caching and streaming networks [14, 3].

Structure Learning: The vast majority of work on cascading processes has focused on understanding how the graph structure of the network (e.g. power laws, small world, expansion etc.) affects the spread of cascades. We focus on the *inverse problem*: if we only observe the states of nodes as the cascades spread, can we infer the underlying graph? Structure learning is the crucial first step before we can *use* network structure; for example, before we find influential nodes in a network (e.g. for viral marketing) we need to know the graph. Often however we may only have crude, prior information about what the graph is, or indeed no information at all.

For example, in online social networks like Twitter or Facebook, we may have access to a *nominal* graph of all the friends of a user. However, clearly not all of them have an equal effect on the user’s behavior; we would like to find the sub-graph of important links. In several other settings, we may have no a-priori information; examples include information forensics that study the spread of worms, and offline settings like real-world epidemiology and social science. The standard practice seems to be to use crude/nominal subgraphs if they exist (e.g. Twitter), or find graphs by other means (e.g. surveys). We propose to take a *data-driven* approach, finding graphs from observations of the cascades themselves.

While structure learning from cascades is an important primitive, there has been very little work investigating it (we summarize below). There are two related issues that need to be addressed: *(a) algorithms*: what is the method, and its complexity, and *(b) performance*: how many observations are needed for reliable graph recovery? The main intellectual contribution of this paper is characterizing the performance of two algorithms we develop, and a lower bound showing they perform close to optimal. To the best of our knowledge, there exists no prior work on performance analysis (i.e. characterizing the number of observations needed) for learning graphs of epidemic cascades.

1.1 Summary of Our Results

We present two algorithms, and information-theoretic lower bounds, for the problem of learning the graph of an epidemic cascade when we are given prior information of a super-graph¹. It is not possible to learn the graph from a single cascade; we study the number of cascades required for reliable learning. Key outcomes of our results are that *(i)* epidemic graph learning can be done in a fast, distributed fashion, *(ii)* with a number of samples that is close to the lower bound. Our results:

(a) Maximum Likelihood: We show that, via a suitable change of variables, the problem of finding the graph most likely to generate the cascades we observe *decouples* into n convex problems – one for each node. Further, for node i , the algorithm requires as input only the infection times of that node’s size- D_i super-neighborhood; it is local both in computation and in the information requirement. Our main result here is to establish that for this efficient algorithm, if d_i is the size of the true neighborhood, then node i needs to be infected $O(d_i^2 \log D_i)$ times before we learn it, for a general graph.

(b) Greedy algorithm: We show that if the graph is a tree, then a natural greedy algorithm is able to find the true neighborhood of a node i with only $O(d_i \log D_i)$ samples. The greedy algorithm involves iteratively adding to the neighborhood the node which “explains” (i.e. could be the likely cause of) the largest number of instances when node i was infected, and removing those infections from further

¹Of course if no super-graph is given, it can be taken to be the complete graph.

consideration.

(c) Lower bounds: We first establish a general information-theoretic lower bound on the number of cascade samples required for approximate graph recovery, for general (but abstract) notions of approximation, and for any SIR process. We then derive two corollaries: one for learning a graph upto a specified edit distance when there is no super-graph information, and another for the case when there is a super-graph, and specified edit distances for each of the nodes. These bounds show that the ML algorithm is at most a factor d away from the optimal.

(d) Markov structure of general cascades: Every set of random variables has an associated Markov graph. In our final result, we show that for a very general SIR epidemic cascade model – essentially any that is causal with respect to time and the directed network graph – the (undirected) Markov graph of the (random) infection times is the *moralized* graph of the true directed network graph on which the epidemic spreads. This allows for learning graph structure using techniques from Markov Random Fields / graphical models, and also illustrates the role of causality.

While here we used the $O(\cdot)$ and $\Omega(\cdot)$ notation for compact statement, we emphasize that our results are *non-asymptotic*, and thus more general than a merely asymptotic result. Thus for fixed values of system parameters and probabilities of error, we give precise bounds on the number of cascades we need to observe. If one is interested in asymptotic results under particular scaling regimes for the parameters, such results can be derived as corollaries of our algorithms (with union bounds if one is interested in complete graph recovery).

A nice feature of our results is that both the algorithms work on a *node by node basis*. Thus for recovering the neighbors of a node we only need information about its super-neighborhood, and solve a local problem. We are also able to find the neighborhood of one or a few nodes, without worrying about finding the neighborhoods of other nodes or the entire graph. Similarly, the number of samples required to recover the neighborhood of a node depend only on the sizes of its own neighborhood and super-neighborhood.

1.2 Related Work

Learning graphs of epidemic cascades: While structure learning from cascades is an important primitive, there has been very little work investigating it:

(a) algorithms: A recent paper [22] investigates learning graphs from infection times for the independent cascade model (similar setting as our paper). However, they take an approach that results in an NP-hard combinatorial optimization problem, which they show can be approximated. Another paper [16] shows max-likelihood estimation in the independent cascade model can be cast as a decoupled convex optimization problem (albeit a different one from ours).

(b) performance: To the best of our knowledge, there has been no work on the crucial question of how many cascades one needs to observe to learn the graph; indeed, this question is the main focus of our paper.

Markov graph structure learning: The ideas in this paper are related to those from Markov Random Fields (MRFs, aka Graphical Models) in statistics and machine learning, but there are also important differences. We overview the related work, and contrast it to ours, in Section 6.

2 System Model

Most of the analytical results of this paper are for the classic and popular *independent cascade* model of epidemics; in particular we will consider the simple one-step model first proposed in [6] and recently popularized by Kempe, Kleinberg and Tardos [9].

Standard independent cascade epidemic model [9]: The network is assumed to be a *directed* graph $G = (V, E)$; for every directed edge (i, j) we say i is a parent and j is a child of the corresponding other node. Parent may infect child along an edge, but the reverse cannot happen; we allow bi-directed edges (i.e. it is possible that (i, j) and (j, i) are in E). Let $\mathcal{V}_i := \{j : (j, i) \in E\}$ denote the set of parents of each node i , and for convenience we also include $i \in \mathcal{V}_i$. Epidemics proceed in discrete time; all nodes are initially in the *susceptible* state. At time 0, each node tosses a coin and independently becomes *active*, with probability p_{init} . This set of initially active nodes are called *seeds*. In every time step each active node probabilistically tries to infect its susceptible children; if node i is active at time t , it will infect each susceptible child j with probability p_{ij} , independently. Correspondingly, a node j that is susceptible at time t will become active in the next time step, i.e. $t + 1$, if *any one* of its parents infects it. Finally, a node remains active for only *one* time slot, after which it becomes *inactive*: it does not spread the infection, and cannot be infected again. Thus this is an ‘‘SIR’’ epidemic, where some nodes remain forever susceptible because the epidemic never reaches them, while others transition according to:

susceptible \rightarrow **active for one time step** \rightarrow **inactive**. A sample path of the independent cascade model is illustrated in Figure 1.

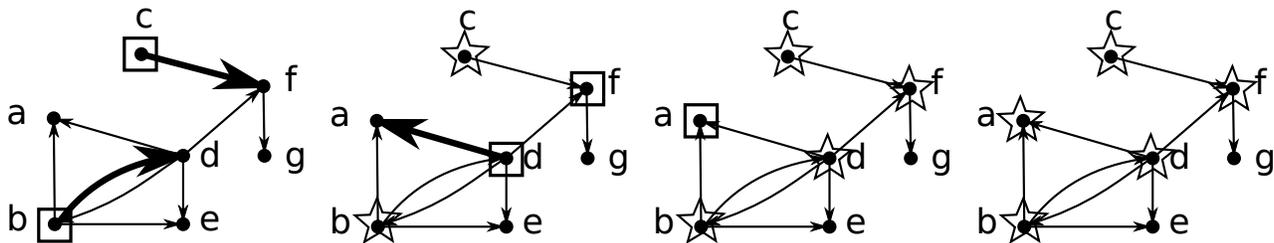


Figure 1: **Illustration of the independent cascade model:** This figure illustrates a sample path of the evolution of the independent cascade model. The four figures above represent the state of the system at time steps 0, 1, 2 and 3 respectively. A node with no box around it means that it is in susceptible state, a node with a square around it means that it is active and a node with a star around it means that it is inactive. At time step 0, nodes b and c are chosen as seeds. They infect d and f respectively and turn inactive. In time step 1, d infects a where as f fails to infect any of its children. In time step 2, a does not have any children to infect. Once a turns inactive in time step 3, the epidemic stops.

Note thus that the set parental set is $\mathcal{V}_i = \{j : p_{ji} > 0\}$, i.e. the set of all nodes that have a non-zero probability of infecting i .

Observation model: For an epidemic cascade u that spreads over a graph, we observe for each node i the time t_i^u when i became active. If i is one of the seed nodes of cascade u then $t_i^u = 0$, and for nodes that are never infected in u we set $t_i^u = \infty$. Let t^u denote the vector of infection times for cascade u . We observe more than one cascade on the same graph; let \mathcal{U} be the set of cascades, and $m = |\mathcal{U}|$ be the number, which we will often refer to as the *sample complexity*. Each cascade is assumed to be generated and observed as above, independent of all others.

(possible) Super-graph information: In several applications, we (may) also have prior knowledge about the network, in the form of a directed *super-graph*² of G . We find it convenient to represent super-graph information as follows: for each node i , we are given a set $\mathcal{S}_i \subset V$ of nodes that contain its true parents; i.e. $\mathcal{V}_i \subset \mathcal{S}_i$ for all i . In terms of edge probabilities, this means that $p_{ji} > 0$ (strictly) for $j \in \mathcal{V}_i$, and $p_{ji} = 0$ for $j \in \mathcal{S}_i \setminus \mathcal{V}_i$. Of course if no super-graph is available we can set $\mathcal{S}_i = V$, the set of all nodes; so from now on we assume a \mathcal{S}_i is always available.

Problem description: Using the vectors of infection times $\{t^u\}$ we are interested in finding the parental neighborhood \mathcal{V}_i , for some or all of the nodes i . That is, we want to find the set of nodes that can infect i . This is not possible when we only observe a single cascade; we will thus be interested in learning the graph from *as few cascades as possible*.

Note that multiple seeds begin each cascade $u \in \mathcal{U}$; thus, for a single cascade even at time step 1 we will not be able to say with surety which seed infected which individual.

Correlation decay: Loosely speaking, random processes on graphs are said to have “correlation decay” if far away nodes have negligible effects. For our problem, this means that the cascade from each seed does not travel too far. Formally, all the results in this paper assume that there exists a number $\alpha > 0$ such that for every node i , the sum of all probabilities of incoming edges satisfies $\sum_k p_{ki} < 1 - \alpha$. The following lemma clarifies what this assumption means for the infection times of a node.

Lemma 1. *For any node i and time t , we have*

$$\mathbb{P}[T_i = t] \leq (1 - \alpha)^{t-1} p_{init}$$

Thus, the probability $\mathbb{P}[T_i < \infty]$ that a node is infected satisfies $p_{init} < \mathbb{P}[T_i < \infty] < \frac{p_{init}}{\alpha}$. Also, the average distance from a node to any seed that infected it is at most $\frac{1}{\alpha}$. We discuss the case where there is no correlation decay in the Discussion section.

Interpreting the results: Each cascade we observe provides some information about the graph. Suppose we want to infer the presence, or absence, of the directed edge (i, j) (i.e. if $p_{ij} > 0$ or not). Note that if the parent i is not infected in a cascade, then *that cascade provides no information* about (i, j) : since the parent was never infected, no infection attempt was made using that edge; the “edge activation variable” was never sampled. While our theorems are in terms of the total number m of cascades needed for graph estimation, for a meaningful interpretation of this number one needs to realize that the expected number of times we get *useful* information about any edge is, on average, between mp_{init} and mp_{init}/α . These are also the bounds on the average number of times a particular node is infected in a particular cascade.

We provide both upper bounds (via two learning algorithms), and (information theoretic) lower bounds on the sample complexity. Note that the *execution* of our algorithms does not require knowledge of these parameters like p_{init}, α etc.; these are defined only for the *analysis*.

²For example, on social networks like Facebook or Twitter, we may know the set of all friends of a user, and from these we want to find the ones that most influence the user.

3 Maximum Likelihood

The graph learning problem can be interpreted as a parameter estimation problem: for each cascade, the vector T of infection times is a set of random variables that has a joint distribution which is determined by a set of parameters $p_{ji} \geq 0$ for every i and $j \in \mathcal{S}_i$. We want to find these parameters, or more specifically the identities of the edges where they are non-zero, from samples t^u , $u \in \mathcal{U}$. Each choice of parameters has an associated probability, or likelihood, of generating the infection times we observe. The classical *Maximum-likelihood (ML) estimator* advocates picking the parameter values that maximize this likelihood.

Our crucial insight in this section is that, with an appropriate change of variables the likelihood function has a particularly nice (decoupled, convex) form, enabling both efficient implementation and analysis. In particular, define $\theta_{ij} := -\log(1 - p_{ij})$; note that $p_{ij} = 0 \Leftrightarrow \theta_{ij} = 0$.

Further, for each node i let $\theta_{*i} := \{\theta_{ji}; j \in \mathcal{S}_i\}$ be the set of parameters corresponding to the possible parents \mathcal{S}_i of node i . Let θ be the set of all parameters of the graph. Note that $\theta \geq 0$ (i.e. every parameter is positive or zero). Finally, we define the *log-likelihood* of a vector t of samples to be

$$\mathcal{L}(t; \theta) := \log(\Pr_{\theta}[T = t])$$

The proposition below shows how \mathcal{L} decouples into convex functions with this change of variables.

Proposition 1 (convexity & decoupling). *For any vector of parameters θ , and infection time vector t , the log-likelihood is given by*

$$\mathcal{L}(t; \theta) = \log(p_{init}^s (1 - p_{init})^{n-s}) + \sum_i \mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i})$$

where s is the number of seeds (i.e. nodes with $t_i = 0$), and the node-based term

$$\mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i}) := - \sum_{j: t_j \leq t_i - 2} \theta_{ji} + \log \left(1 - \exp \left(- \sum_{j: t_j = t_i - 1} \theta_{ji} \right) \right)$$

Furthermore, $\mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i})$ is a concave function of θ_{*i} , for any fixed $t_{\mathcal{S}_i}$.

Proof: Please see appendix.

Remark: The overall log-likelihood $\mathcal{L}(t; \theta)$ has now decoupled because it is the sum of n terms of the form $\mathcal{L}_i(t; \theta_{*i})$, each of which depend on a *different* set of variables θ_{*i} . Thus each one can be optimized, and analyzed, in isolation.

The *algorithmic* implications of this proposition are:

- (a) if we are only interested in a small subset of nodes, we can find their parental neighborhood by solving a separate $|\mathcal{S}_i|$ -variable convex program for each one,
- (b) even if we want to find the entire graph, the decoupling allows for parallelization, and speedup: solving n convex programs with n variables each is much faster than solving one program with n^2 variables.
- (c) The function \mathcal{L}_i is fully determined by the times $t_{\mathcal{S}_i}$ of the node's super-neighborhood; it does not need knowledge of the infection times of other nodes.

Proposition 1 is equally crucial *analytically*, as it enables us to derive bounds on the number of cascades required for us to reliably select the neighborhood, via analysis of the first-order optimality conditions of the convex program. In particular, we will see that complementary slackness conditions from convex programming, and concentration results, are key to proving our results on the sample complexity of the ML procedure.

The ML algorithm for finding the parental neighborhood of node i is formally stated below. It involves solving the convex program corresponding to the max-likelihood, and setting small values of θ_{ji} to 0. The threshold for this cut-off is η , which is an input to the procedure.

Algorithm 1 ML Algorithm for Node i

- 1: Find the optimizer of the empirical likelihood, i.e. find

$$\hat{\theta}_{*i} := \arg \max_{\theta_{*i}} \sum_u \mathcal{L}_i(t_{\mathcal{S}_i}^u; \theta_{*i})$$

where $\mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i})$ is as defined in Prop. 1.

- 2: Estimate the parental neighborhood by thresholding:

$$\hat{\mathcal{V}}_i := \{j : \hat{\theta}_{ji} \geq \eta\}$$

- 3: Output $\hat{\mathcal{V}}_i$.
-

Our main analytical result of this section is a characterization of the performance of this ML algorithm, in terms of the number of cascades it needs to reliably estimate the parental neighborhood of any node i .

Theorem 1. *Consider a node i with true parental degree $d_i := |\mathcal{V}_i|$, and super-graph degree $D_i := |\mathcal{S}_i|$. Let $p_{i,min} := \min_{j \in \mathcal{V}_i} p_{ji}$ be the strength of the edge from the weakest parent. Assume $d_i p_{i,min} < \frac{1}{2}$. Then, for any $\delta > 0$, if the number of cascades $m = |\mathcal{U}|$ satisfies*

$$m > \frac{c}{p_{i,min}} \left(\frac{1}{\alpha^7 \eta^2 p_{i,min}^2} \right) d_i^2 \log \left(\frac{D_i}{\delta} \right) \quad (1)$$

Then, with probability greater than $1 - \delta$, the estimate $\hat{\mathcal{V}}_i$ from the ML algorithm with threshold η will have

- (a) *no false neighbors, i.e. $\hat{\mathcal{V}}_i \subset \mathcal{V}_i$, and*
 (b) *all strong enough neighbors: if $j \in \mathcal{V}_i$ and $p_{ji} > \frac{8}{\alpha}(e^{2\eta} - 1)$, then $j \in \hat{\mathcal{V}}_i$ as well.*

Here c is a number independent of any other system parameter.

Remarks:

(a) This is a *non-asymptotic* result that holds for *all* values of the system variables $d_i, p_{i,min}, \alpha, p_{i,min}, \eta$ and δ . Appropriate asymptotic results can be derived as corollaries, if required. Note that this result on finding the nodes that influence node i *does not depend on n* .

(b) We can learn the entire neighborhood, i.e. $\hat{\mathcal{V}}_i = \mathcal{V}_i$, by choosing the threshold $\eta \leq \frac{1}{2} \log(1 + \frac{\alpha p_{i,min}}{8})$ low enough, and the corresponding number of cascade samples m according to (1). Thus, the number of times node i needs to be infected before we can reliably (i.e. with a fixed small error probability) learn

its neighborhood scales as $O(d_i^2 \log D_i)$ (for fixed values of other system variables). Our result allows for learning stronger edges with fewer samples.

(c) If we want to learn the structure of the *entire* graph with probability greater than ϵ , we can set $\delta = \epsilon/n$ and then take a union bound over all the nodes. So, for example, if every node has true degree at most $|\mathcal{V}_i| \leq d$, and super-graph degree $|\mathcal{S}_i| \leq D$, then the number of samples needed to learn the entire graph (with probability at least $1 - \epsilon$) scales as $O(d^2 \log \frac{Dn}{\epsilon})$ (for fixed values of other system variables).

(d) The average number of parents of i that are seeds is $d_i p_{init}$. If this is large, then in every cascade there will be a reasonable probability of one of them being seeds, and infecting i in the next time slot. This makes it hard to discern the neighborhood of i ; the (mild) assumption $d_i p_{init} < \frac{1}{2}$ is required to counter this effect. Indeed, in most applications p_{init} is likely to be quite small.

(e) Note that our results depend on the *in-degree* of nodes, not the out-degree. So for example it is possible to have high out-degree nodes (as e.g. in power-law graphs), and still be able to learn the graph with small number of samples.

3.1 Generalized Independent Cascade Model

In this paper, for ease of analysis, we restrict our sample-complexity analysis to one-step independent cascade epidemics, where a node is active for only one time slot after it is infected. However, our algorithmic and bounding approaches apply to a more general class of independent cascade models. Specifically, we consider an extension where each parent now has a probability distribution of the amount of time it waits before infecting a child, and prove a generalization of Proposition 1, which was the key result enabling both the implementation and analysis of the ML algorithm.

Formally, let p_{ji}^τ denote the probability that an active node j infects a susceptible child i , τ time steps after j was infected. The time taken for j to infect i is bounded by a parameter \bar{t} i.e., $p_{ji}^\tau = 0$ for $\tau > \bar{t}$. Note that if we have $\bar{t} = 1$, we recover the standard independent cascade model. The total probability that j infects i is given by $\sum_{\tau \in [\bar{t}]} p_{ji}^\tau$ (which can be strictly less than 1) where $[\bar{t}]$ denotes the set of integers between 1 and \bar{t} (including the end points).

Following in the steps of Proposition 1, define $\theta_{ji}^\tau := -\log \left(\frac{1 - \sum_{r \in [\tau]} p_{ji}^r}{1 - \sum_{r \in [\tau-1]} p_{ji}^r} \right)$. Note that given any parameter vector p_{ji}^τ we obtain the corresponding θ_{ji}^τ and vice versa. Moreover $\theta_{ji}^\tau = 0 \Leftrightarrow p_{ji}^\tau = 0$. Suppose each node is seeded with the infection with probability p_{init} and let $\mathcal{L}(t, \theta)$ denote the log-likelihood of the infection time vector t when the parameters of the model are given by θ . We have the following version of Proposition 1 for the generalized independent cascade model.

Proposition 2. *For any vector of parameters θ , and infection time vector t , the log-likelihood is given by*

$$\mathcal{L}(t; \theta) = \log(p_{init}^s (1 - p_{init})^{n-s}) + \sum_i \mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i})$$

where s is the number of seeds (i.e. nodes with $t_i = 0$), and the node-based term

$$\mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i}) := - \sum_{j: t_j \leq t_i - 2} \sum_{\tau \in [t_i - t_j - 1]} \theta_{ji}^\tau + \log \left(1 - \exp \left(- \sum_{j: t_j < t_i} \theta_{ji}^{t_i - t_j} \right) \right)$$

Furthermore, $\mathcal{L}_i(t_{\mathcal{S}_i}; \theta_{*i})$ is a concave function of θ_{*i} , for any fixed $t_{\mathcal{S}_i}$.

Proof: Please see appendix.

4 Greedy Algorithm

We now analyze the sample complexity of a simple iterative greedy algorithm – for the case when the graph is a tree³. The algorithm is of course defined for general graphs.

The idea is as follows: suppose we want to find the parents of node i from a given set of cascades \mathcal{U} . In each cascade u , the set of nodes that could have possibly infected i is the set of nodes j for which $t_j^u = t_i^u - 1$. In the first step, the algorithm thus picks the j which has $t_j^u = t_i^u - 1$ for the largest number of observed cascades. It then *removes* those cascades from further consideration (since they have been “accounted for”) and proceeds as before on the remaining cascades, stopping when all cascades are exhausted.

Algorithm 2 Greedy Algorithm for Node i

- 1: Initialize unaccounted cascades $U = \mathcal{U}$
 - 2: Initialize $\widehat{\mathcal{V}}_i = \emptyset$
 - 3: **while** $U \neq \emptyset$ **do**
 - 4: Find $k = \arg \max_{j \in \mathcal{S}_i} |\{u \in U : t_j^u = t_i^u - 1\}|$
 - 5: Add it : $\widehat{\mathcal{V}}_i \leftarrow \widehat{\mathcal{V}}_i \cup k$
 - 6: Remove cascades : $U \leftarrow U \setminus \{u : t_k^u = t_i^u - 1\}$
 - 7: **end while**
 - 8: Output $\widehat{\mathcal{V}}_i$
-

Our main result for this section is below.

Theorem 2. *Suppose the graph G is a tree, and the degree of node i is $d_i := |\mathcal{V}_i|$. Suppose also that $p_{\text{init}} < \frac{\alpha^2 p_{\text{min}}}{16ed}$. If Algorithm 2 is given a super-neighborhood of size $D_i := |\mathcal{S}_i|$, then for any $\delta > 0$ if the number of samples satisfies*

$$m > \frac{c}{p_{\text{init}}} \left(\frac{1}{p_{\text{min}}} \right) d_i \log \frac{D_i}{\delta}$$

then with probability at least $1 - \delta$ the estimate from the greedy algorithm will be the same as the true neighborhood, i.e. $\widehat{\mathcal{V}}_i = \mathcal{V}_i$. Here c is a constant independent of any other system parameter.

5 Lower Bounds

We now turn our attention to establishing lower bounds on the number of cascades that need to be observed for even approximately learning graph structure, using *any* algorithm. Clearly, we now cannot focus on learning just one graph, since in that case we could come up with an “algorithm” tailored to find precisely

³We believe (especially since we have correlation decay) that our results can be easily extended to the case of “locally tree-like” graphs; e.g. random graphs from the Erdos-Renyi, random regular or several other popular models.

that one graph. Instead, as is standard practice in information-theoretic lower bounds, we need to consider a collection (or “ensemble”) of graphs, and study how many cascades are needed to (approximately) find *any one* graph from this collection.

We first state a lower bound in a general setting, for any pre-defined ensemble and notion of approximate recovery. We then provide two corollaries specializing it to our independent cascade epidemic model, edit distance approximation, and two natural graph ensembles.

General Setting: Consider any general cascading process generating infection times $\{T_i\}$. Let \mathcal{G} be a fixed collection of graphs and corresponding edge probabilities, and let G be a graph chosen uniformly at random from this collection. We then generate a set \mathcal{U} , with $|\mathcal{U}| = m$, of independent cascades, and observe infection times $T^{\mathcal{U}}$. Let $\widehat{G}(T^{\mathcal{U}})$ be a graph estimator that takes the observations as an input and outputs a graph. Finally, we say that a graph G' approximately recovers graph G if $G \in \mathcal{B}(G')$, where $\mathcal{B}(G') \subseteq \mathcal{G}$ is any pre-defined set of graphs, with one such set defined for every G' .

So for example, if we are interested in exact recovery, we would have $\mathcal{B}(G') = \{G'\}$, i.e. the singleton. If we were interested in edit distance of s , we would have $\mathcal{B}(G')$ be the set of all graphs within edit distance s of G' .

We define the probability of error of a graph estimator $\widehat{G}(\cdot)$ to be

$$P_e(\widehat{G}) := \mathbb{P}[G \notin \mathcal{B}(\widehat{G}(T^{\mathcal{U}}))]$$

where the probability is calculated over the randomness in the choice of G itself, and the generation of infection times in this G . Note that the definition defines error to be when approximate recovery (as defined by the sets \mathcal{B}) fails.

Theorem 3. *In the general setting above, for any graph estimator to have a probability of error of P_e , we need*

$$m \geq \frac{(1 - P_e) \log \frac{|\mathcal{G}|}{\sup_{G'} |\mathcal{B}(G')|} - 1}{\sum_{i \in V} H(T_i)}$$

where $H(\cdot)$ is the entropy function.

Proof. To shorten notation, we will denote $\widehat{G}(T^{\mathcal{U}})$ simply by \widehat{G} . The proof uses several basic information-theoretic inequalities, which can be found e.g. in [5]. In the following $H(\cdot)$ denotes entropy and $I(\cdot; \cdot)$ denotes mutual information.

We can see that the following diagram forms a Markov chain

$$G \longleftrightarrow T^{\mathcal{U}} \longleftrightarrow \widehat{G}$$

We have the following series of inequalities:

$$\begin{aligned}
H(G) &= I(G; \widehat{G}) + H(G | \widehat{G}) \\
&\stackrel{(\varsigma_1)}{\leq} I(G; T^U) + H(G | \widehat{G}) \\
&\stackrel{(\varsigma_2)}{\leq} H(T^U) + H(G | \widehat{G}) \\
&\stackrel{(\varsigma_3)}{\leq} mH(T) + H(G | \widehat{G}) \\
&\stackrel{(\varsigma_4)}{\leq} m \sum_{i \in V} H(T_i) + H(G | \widehat{G})
\end{aligned}$$

where (ς_1) follows from the data processing inequality, (ς_2) follows from the fact that the mutual information between two random variables is less than the entropy of either of them, (ς_3) and (ς_4) follows from the subadditivity of entropy. Since G is sampled uniformly at random from \mathcal{G} , we have that $H(G) = \log |\mathcal{G}|$. We now use Fano's inequality to bound $H(G | \widehat{G})$.

$$\begin{aligned}
H(G | \widehat{G}) &\stackrel{(\varsigma_1)}{\leq} H(G, Err | \widehat{G}) \\
&\stackrel{(\varsigma_2)}{\leq} H(Err | \widehat{G}) + H(G | Err, \widehat{G}) \\
&\stackrel{(\varsigma_3)}{\leq} H(Err) + H(G | E, \widehat{G}) \\
&\stackrel{(\varsigma_4)}{\leq} 1 + P_e \log |\mathcal{G}| + (1 - P_e) \log \sup_{\widehat{G}} |\mathcal{B}_s(\widehat{G})|
\end{aligned}$$

where Err is the error indicator random variable (i.e., is 1 if $G \notin \mathcal{B}(\widehat{G})$ and 0 otherwise), so that $P_e = \mathbb{E}[Err]$. (ς_1) follows from the monotonicity of entropy, (ς_2) follows from the chain rule of entropy, (ς_3) follows from the monotonicity of entropy with respect to conditioning and (ς_4) follows from Fano's inequality. Combining the above two results, we obtain

$$\begin{aligned}
m \sum_{i \in V} H(T_i) &\geq (1 - P_e) \log \frac{|\mathcal{G}|}{\sup_{\widehat{G}} |\mathcal{B}(\widehat{G})|} - 1 \\
\Rightarrow m &\geq \frac{(1 - P_e) \log \frac{|\mathcal{G}|}{\sup_{\widehat{G}} |\mathcal{B}(\widehat{G})|} - 1}{\sum_{i \in V} H(T_i)} \tag{2}
\end{aligned}$$

□

To apply this result to a particular ensemble \mathcal{G} and notion of approximation \mathcal{B} , we need to find a lower bound on $|\mathcal{G}|$, and upper bounds on $|\mathcal{B}(G')|$ for all G' and $H(T_i)$ for all i . The following lemma states an upper bound on $H(T_i)$ for our independent cascade model when we have correlation decay coefficient α . Both our corollaries assume this is the case for all graphs in their respective ensembles.

Lemma 2. *For any graph with correlation decay coefficient α , for any node i , and when $p_{init} < \frac{1}{e}$, we*

have that

$$\begin{aligned}
H(T_i) &\leq \frac{p_{init}}{1-\alpha} \left(\log \frac{1}{p_{init}} + \left(\frac{1-\alpha}{\alpha} \right)^2 \log \frac{1}{1-\alpha} \right) \\
&\quad - \left(1 - \frac{p_{init}}{\alpha} \right) \log \left(1 - \frac{p_{init}}{\alpha} \right) \\
&=: p_{init} \bar{H}(\alpha, p_{init})
\end{aligned}$$

Note that the *edit distance* between two graphs is the number of edges present in only one of the two graphs but not the other (i.e. the number of edges in the symmetric difference of the two graphs). Our first corollary is for the case when there is no super-graph information, and we want to approximate in global edit distance.

Corollary 1. *Let \mathcal{G}_d denote the set of all graphs with in-degrees bounded by d , and $\mathcal{B}_\gamma(G')$ be the set of all graphs within edit distance γ of G' . Let $p_{init} < \frac{1}{e}$. Then for any algorithm to have a probability of error of P_e , we need*

$$m > \frac{(1-P_e)}{p_{init}} \frac{1-\alpha}{\bar{H}(\alpha, p_{init})} \left(d \log \frac{n}{d} - \frac{\gamma}{n} \log \frac{n^2}{\gamma} \right) - 1$$

Proof. We have that

$$\begin{aligned}
\log |\mathcal{G}_d| &= \log \binom{n}{d}^n = (1+o(1)) nd \log \frac{n}{d} \\
\log |\mathcal{B}_\gamma(G')| &\leq \log \binom{\binom{n}{2}}{\gamma} \leq \gamma \log \frac{n^2}{\gamma}
\end{aligned}$$

Using the above two equations along with Theorem 3 and Lemma 2 gives us the result. \square

Note that the number of times a node is infected thus needs to be $\Omega((d - \frac{2\gamma}{n}) \log n)$ (since it is of the same order as mp_{init}). For exact recovery, i.e. $\gamma = 0$, we see that our result on the performance of our ML algorithm – specialized to the no prior information case $D = n$ – is off by just a factor d in terms of the number of samples required.

The second corollary is for the case when we do have prior supergraph information. In particular, we assume that we are given sets \mathcal{S}_i , of size $|\mathcal{S}_i| = D$, for each node i . We consider the ensemble $\mathcal{G}_{D,d}$ of all in-degree- d subgraphs of this fixed supergraph. Thus for each node, we need to learn the d parents it has, from a given super-set of size D . Finally, for each node i we allow s_i errors; let $\mathcal{B}_s(G')$ be the corresponding set of all subgraphs of the given supergraph.

Corollary 2. *For any estimator to have a probability of error of P_e in the setting above, the number of samples m must be bigger than*

$$\frac{(1-P_e)}{p_{init}} \frac{1-\alpha}{\bar{H}(\alpha, p_{init})} \left(d \log \frac{D}{d} - \frac{1}{n} \sum_i s_i \log \frac{eD}{s_i} + \log \max(s_i, 1) \right) - 1$$

Remark: Specializing this result to exact recovery (i.e. $s_i = 0$) removes dependence on n , and again shows us that the ML algorithm is within a factor d of optimal for the case when we have a super-graph.

Proof. We have the following bound on the size of the ensemble:

$$\log |\mathcal{G}_d| = \log \binom{D}{d}^n = (1 + o(1)) nd \log \frac{D}{d}$$

Similarly,

$$\begin{aligned} \log |\mathcal{B}_s(\widehat{G})| &\leq \log \prod_{i \in V} \left(\sum_{l=0}^{s_i} \binom{D}{l} \right) \\ &\leq \log \prod_{i \in V} \left(\max(1, s_i) \binom{D}{s_i} \right) \\ &\leq \sum_{i \in V} \log \left(\max(1, s_i) \left(\frac{De}{s_i} \right)^{s_i} \right) \\ &= \sum_{i \in V} \log \max(1, s_i) + \sum_{i \in V} s_i \log \frac{De}{s_i} \end{aligned} \quad (3)$$

where

$$\mathcal{B}_s(\widehat{G}) = \{ \widetilde{G} \in \mathcal{G}_d : \widetilde{\mathcal{V}}_i \Delta \widehat{\mathcal{V}}_i \leq s_i \forall i \in V \}$$

Note that in the second inequality we assume $s_i \leq \frac{D}{2}$ because otherwise if $d < \frac{D}{2}$, we can choose $\widehat{\mathcal{V}}_i = \Phi$ and if $d \geq \frac{D}{2}$, we can choose $\widehat{\mathcal{V}}_i = \mathcal{V}_i$. Using Theorem 3, (3) and Lemma 2 gives us the first part of the result. \square

6 General SIR Epidemics: Markov Graphs and Causality

In this section we consider a much more general model for SIR epidemics/cascades on a directed graph, and establish a connection to the classic formalism of Markov Random Fields (MRFs) – see e.g. [12] for a formal introduction. Specifically, we show that the (undirected) Markov graph of infection times of an SIR epidemic is obtained via the *moralization* of the true (directed) network graph on which the cascade spreads. A moralized graph, as defined below, is obtained by adding edges between all parents of a node (i.e. “marrying” them), and removing all directions from all edges. Graph moralization also arises in Bayesian networks, and we comment on the relationship, and the role of causality, after we present our result.

We first briefly describe our general model for SIR epidemics, then define its Markov graph, and finally present our result.

General SIR epidemics: We now describe a general model for SIR epidemics propagating on a directed graph. Nodes can be in one of three *states*: **0** for susceptible, **1** for infected and active, and **2** for resistant and inactive; we restrict our attention to discrete time in this paper. Let $X_i(t)$ be the state of node i at time t , and $X(t)$ to be the vector corresponding to the states of all nodes. We require that this process be causal, and governed by the true directed graph G , in the sense that for any time step t ,

$$\mathbb{P}[X(t) = x(t) | x(0 : t-1)] = \prod_i \mathbb{P}[X_i(t) = x_i(t) | x_{\mathcal{V}_i}(0 : t-1)] \quad (4)$$

where the notation $x(1:t) = \{x(s), 1 \leq s \leq t\}$ is the entire history upto time t , and as before \mathcal{V}_i is the set of parents of node i , and includes $i \in \mathcal{V}_i$ as well. Note the above encodes that the probability distribution of each node's next state depends only on the history of itself and its neighbors, but is otherwise independent of the history or current state of the other nodes. We assume that the cascade is initially seeded arbitrarily, i.e. $x(0)$ can be any fixed initial condition.

For each node i , let $T_i^{(1)}$ be the (random) time when its state transitioned from $\mathbf{0}$ to $\mathbf{1}$, and $T_i^{(2)}$ for the time from $\mathbf{1}$ to $\mathbf{2}$ (of course, if neither happened then we can take them to be ∞). Let $T_i = (T_i^{(1)}, T_i^{(2)})$ be the summary for node i 's participation in the cascade.

Markov Graphs: Markov random fields (MRFs, also known as Graphical Models) are a classic formalism, enabling the use of graph algorithms for tasks in statistics, physics and machine learning. The central notion therein is that of the Markov graph of a probability distribution; in particular, every collection of random variables has an associated graph. Every variable is a node in the graph, and the edges encode conditional independence: conditioned on the neighbors, the variable is independent of all the other variables. For our purposes here, the random variables are the $T := \{T_i, i \in V\}$. We say that an undirected graph G' is the Markov graph of the variables T if their joint probability distribution, for all t , factors as follows

$$\mathbb{P}[T = t] = \prod_{c \in \mathcal{C}'} f_c(t_c)$$

for some functions f_c ; here \mathcal{C}' is the set of cliques of G' , and for a clique $c \in \mathcal{C}'$, $t_c := \{t_i, i \in c\}$ is the vector of node times for nodes in c .

We need one more definition before we state our result.

Moralization: Given a directed graph G , its moralized graph \bar{G} is the undirected graph where two nodes are connected if and only if they either have a parent-child relationship in G , or if they have a common child, or both. Formally, undirected edge (i, j) is present in \bar{G} if and only if at least one of the following is true

- (a) directed edges (i, j) or (j, i) are present in G , or
- (b) there is some node k such that (i, k) and (j, k) are present in G (i.e. k is a common child).

Figure 2 illustrates the process of moralization with an example.

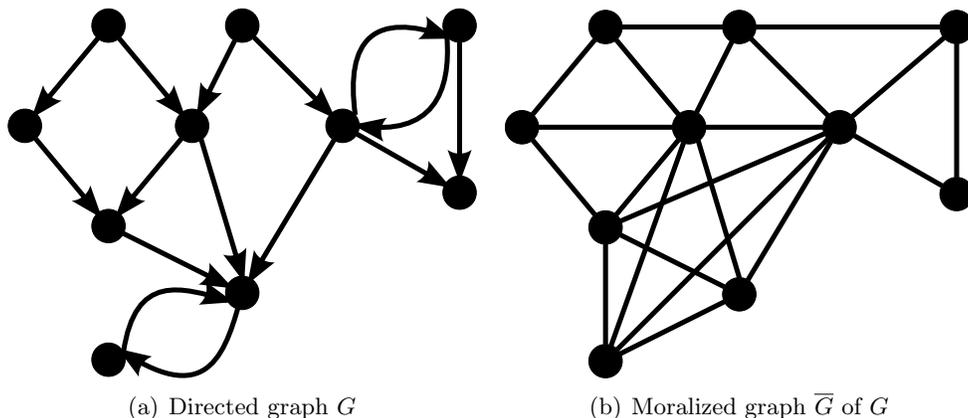


Figure 2: An example of moralization

Theorem 4. *Suppose infection times T are generated from a general SIR epidemic, as above, propagating on a directed network graph G . Let \bar{G} be the (undirected) moralized graph of G . Then, \bar{G} is the Markov graph of T .*

Remarks: The main appeal of this result arises from the generality of the model; indeed, it may be possible to learn the moralized graph even when we may not know what the precise epidemic evolution model is, as long as it satisfies (4). In particular, related to the focus of this paper, there has been substantial work on learning the Markov graph structure of random variables from samples. In our setting, each cascade is a sample from the joint distribution of T , and hence one can imagine using some of these techniques. Markov graph learning techniques can generally be divided into

(a) those that assume a specific class of probability distributions: see e.g. [15, 21] for Gaussian MRFs, [20, 2] for Ising models, [8] for general discrete pairwise distributions. These typically require knowledge of the precise parametric form of the dependence, but then enable learning with a smaller number of samples. (b) distribution-free algorithms, usually for discrete distributions and based on conditional independence tests [1, 4, 17]. These do not need to know the parametric form a-priori, but typically have higher computational and sample complexity.

Causality: It is interesting to contrast Theorem 4 with the other results in this paper. In particular, on the one hand, Theorems 1 and 2 utilize the fact precise causal process that generates T to find the exact true directed network graph. On the other hand, applying a Markov graph learning technique directly to the samples of T , without leveraging the process that generated them, only allows us to get to the moralized graph. It thus serves as a motivating example to extend the study of graph learning from samples to causal phenomena, in a way that explicitly takes into account time dynamics.

Moralization also arises in Bayesian networks; this is an alternative formulation that associates an *acyclic* directed graph with a probability distribution. In that setting, the undirected Markov graph is also the moralization of this directed graph. We note however that our original true network graph G can have directed cycles; in our setting the moralization arises from (ignoring the) causality in time.

7 Experiments

As an initial empirical illustration of our results, in this section, we present – via Figures 3, 4, 5 and 6 – empirical evaluations of both the ML and Greedy algorithms on synthetic graphs, and sub-graphs of the Twitter graph. In all cases, for the ML algorithm the threshold η was picked via cross-validation.

8 Summary and Discussion

This paper studies the problem of learning the graph on which epidemic cascades spread, given only the times when nodes get infected, and possibly a super-graph. We studied the sample complexity – i.e. the number of cascade samples required – for two natural algorithms for graph recovery, and also established a corresponding information-theoretic lower bound. To our knowledge, this is the first paper to study the sample complexity of learning graphs of epidemic cascades. Several extensions suggest themselves; we discuss some below.

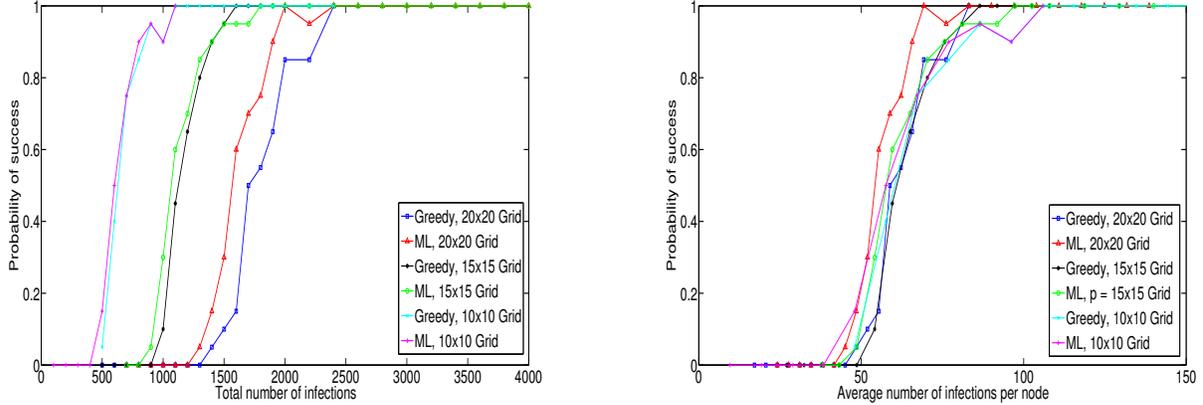


Figure 3: **Interpreting sample complexity:** As mentioned in Section 2, and re-inforced by our theorems, consistent structure recovery is governed not so much by the total number of infections m in the network, as by the number of times a node is infected (which is approx. $p_{init}m$). This figure provides some empirical validation of this claim; the plots on the left and right are from the same set of experiments using the ML algorithm (and no super-graph information). On the left we plot the probability of successful recovery of the entire graph as a function of m , while on the right we plot it as a function of the average number of times a node was infected; for several different sizes of 2-d grids. On the left, we see that the total number of cascades varies noticeably with grid size, but the average number of infections does not. This squares with Theorem 1, since in all these graphs the d is the same, and $\log n$ does not vary much either.

Observation Model: In this paper it is assumed that we have access to the times when nodes get infected. However, this may not always be possible. Indeed a weaker assumption is to only know the infected set in each cascade. To us this seems like a much harder problem, e.g. it is now not clear that there is a decoupling of the global graph learning problem.

Decoupling: A key step in our ML results is to show that the global graph finding problem decouples into n local problems. Our proof of this fact can be extended to any causal network process – i.e. any process where the state $x_i(t)$ depends only on $x_{\mathcal{V}_i}(t-1)$ – under the assumption that we can reconstruct the entire process trajectory from our observations (so e.g. the weaker observation model above would not fall into this class). In particular, it holds for more general models of epidemic cascade propagation as well; we focused on the discrete-time one-step model as a first step.

Correlation decay: Our results are for the case of correlation decay, i.e. when the cascade from one seed reaches a constant depth of nodes before extinguishing. Equally interesting and relevant is the case without correlation decay, when the cascade from each seed can reach as much as a constant fraction of the network. We suspect, based on experiments, that our algorithms would be efficient in this case as well; however, a proof would be technically quite different, and interesting.

Greedy algorithms: As can be seen in our experiments, the greedy algorithm performs quite well even when the graph is far from being a tree (i.e. has several small cycles). It would be interesting to develop an alternate and more general proof of the performance of the greedy algorithm. We also note that one can easily formulate greedy algorithms in more general epidemic settings; this would involve iteratively choosing the parameter that gives the biggest change in the corresponding likelihood function.

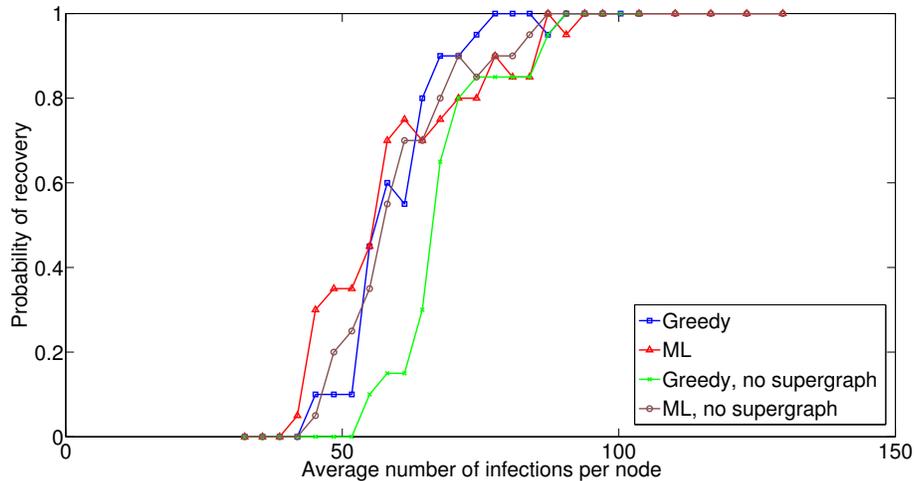


Figure 4: **Effect of super-graph information:** The presence of super-graph information can reduce the number of node infections (and hence cascades) required to learn the graph. Here we plot the probability of successful recovery for a 200-node random 4-regular graph, for the ML and Greedy algorithms, for two scenarios: when we are given a super-graph of regular degree 8 that contains the true graph, and when we are not given such information. We can see that the extent of reduction in sample complexity is moderate, reflecting the fact that the effect of super-graph information is logarithmic (i.e. $\log D$ vs $\log n$).

References

- [1] P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788, 2006.
- [2] A. Anandkumar and V. Y. F. Tan. High-Dimensional Structure Learning of Ising Models : Tractable Graph Families. *Preprint*, June 2011.
- [3] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg. Epidemic live streaming: optimal performance trade-offs. *SIGMETRICS Perform. Eval. Rev.*, 36:325–336, June 2008.
- [4] G. Bresler, E. Mossel, and A. Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *APPROX '08 / RANDOM '08*, pages 343–356, Berlin, Heidelberg, 2008. Springer-Verlag.
- [5] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.
- [6] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.
- [7] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. 13th International Conference on World Wide Web, WWW '04*, pages 491–501, New York, NY, USA, 2004. ACM.

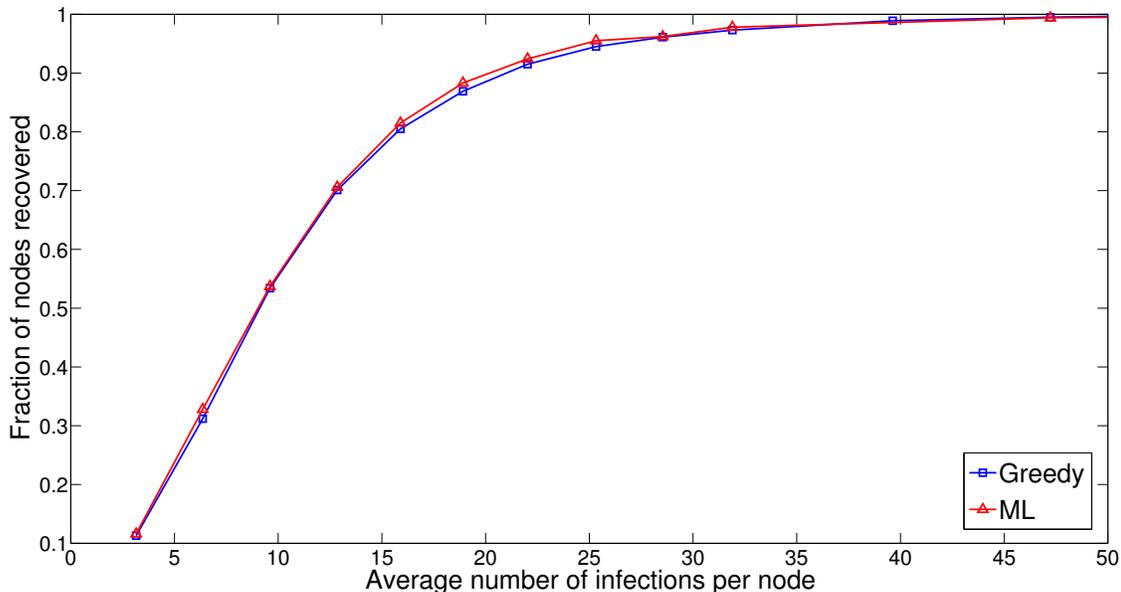


Figure 5: **Twitter graph as a super-graph:** We extracted a directed graph of 1000 nodes from Twitter as follows: if i follows j on Twitter (i.e. j 's tweets appear in i 's feed) then we put a directed edge $j \rightarrow i$ in the graph. We then treat this as a super-graph; from this we chose a 4-regular sub-graph of “real” neighbors – i.e. each person is assumed to have at most 4 significant influencers from among the people she/he follows. Edge parameters were chosen so as to give weight to only the chosen important parents. Infections were sampled by simulating the independent cascade model using the above parameters. Both ML and Greedy algorithms were given the infections as input along with the 1000 node graph as a super graph. For various number of total infections, the x-axis shows the average number of infections of a node and the y-axis shows the fraction of nodes whose neighborhoods are recovered exactly by the algorithm.

- [8] A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 378–387, 2011.
- [9] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.
- [10] J. O. Kephart and S. R. White. Directed-graph epidemiological models of computer viruses. *Security and Privacy, IEEE Symposium on*, 0:343, 1991.
- [11] D. Kosterev, C. Taylor, and W. Mittelstadt. Model validation for the august 10, 1996 wsc system outage. *Power Systems, IEEE Transactions on*, 14(3):967–979, aug 1999.
- [12] S. Lauritzen. *Graphical models*. Oxford University Press, 1996.
- [13] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on Digg and Twitter social networks. In *Proc. International AAAI Conference on Weblogs and Social Media*, 2010.

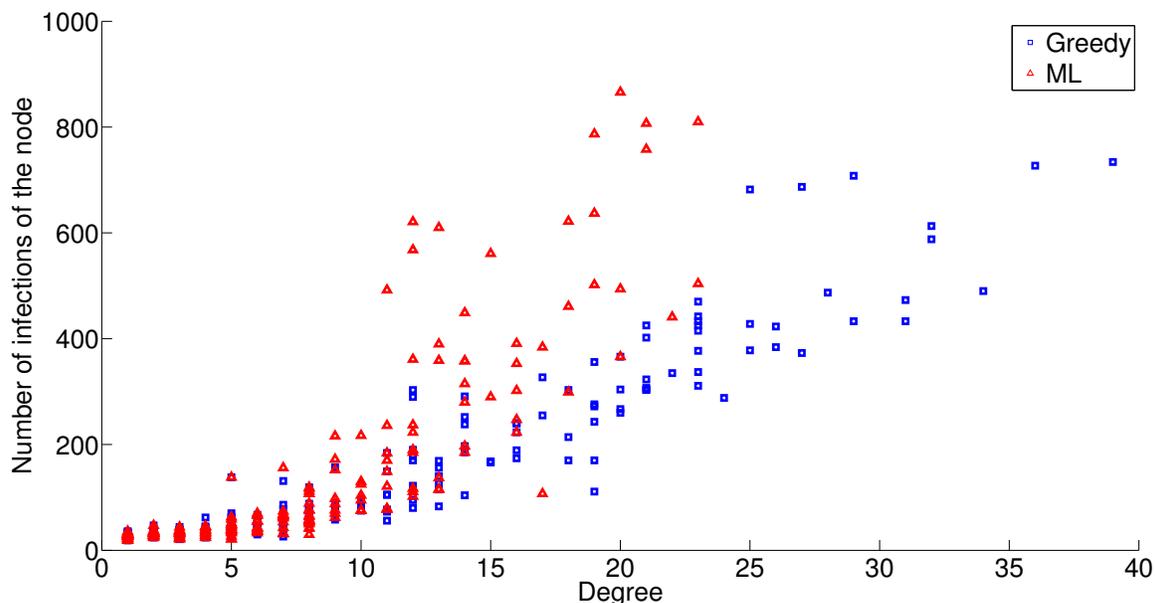


Figure 6: **Dependence on degree for Twitter graph:** This figure is a scatter plot where for each node we plot its degree, and the number of times it was infected before ML or Greedy succeeded in finding its neighborhood. The graph is a 300 node graph was extracted from Twitter (with edges made as explained in Figure 5); now however this is treated as the true graph to be learnt, and the algorithm is given no super-graph information. At least for this example, the sample complexity increases super-linearly with degree for the ML algorithm, where as the dependence of the sample complexity on the degree is almost linear for the Greedy algorithm. This is in spite of the fact that the graph is far from being a tree; it has several small cycles.

- [14] L. Massoulié and A. Twigg. Rate-optimal schemes for peer-to-peer live streaming. *Performance Evaluation*, 65(11-12):804 – 822, 2008.
- [15] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.
- [16] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *Proc. Neural Information Processing Systems (NIPS)*, 2010.
- [17] P. Netrapalli, S. Banerjee, S. Sanghavi, and S. Shakkottai. Greedy learning of markov network structure. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1295 –1302, sept 29 - oct. 1 2010.
- [18] C. Perrow. *Normal Accidents: Living with High-Risk Technologies*. Princeton University Press, updated edition, Sept. 1999.
- [19] V. Poor. *An Introduction to Signal Detection and Estimation*. Springer, 1994.
- [20] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional graphical model selection using l_1 -regularized logistic regression. *Annals of Statistics*, 38(3):1287–1319, 2010.

- [21] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. Model selection in Gaussian graphical models: High-dimensional consistency of l1-regularized MLE. 2008.
- [22] M. G. Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proc. 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10*, pages 1019–1028, New York, NY, USA, 2010. ACM.
- [23] M. L. Sachtjen, B. A. Carreras, and V. E. Lynch. Disturbances in a power transmission system. *Phys. Rev. E*, 61:4877–4882, May 2000.
- [24] Z. Zhou, R. Bandari, J. Kong, H. Qian, and V. Roychowdhury. Information resonance on Twitter: watching Iran. In *Proc. 1st Workshop on Social Media Analytics, SOMA '10*, pages 123–131, New York, NY, USA, 2010. ACM.

A Correlation decay

Proof of Lemma 1. We establish this by an induction on the number of nodes n in the graph. If $n = 1$, the statement above is obvious. Suppose the statement above is true for all graphs which have upto $n - 1$ nodes. Consider now a graph G that has n nodes. Consider any node i . The statement of the proposition is clearly true for $t = 1$. For $t > 1$, consider the probability that i is infected by a parent $k \in \mathcal{S}_i$ at time step t . This can be upper bounded as follows:

$$\begin{aligned} \mathbb{P}_G [k \text{ infects } i \text{ at time } t] &\leq \mathbb{P}_{\tilde{G}} [T_k = t - 1] p_{ki} \\ &\leq (1 - \alpha)^{t-2} p_{\text{init}} p_{ki} \end{aligned}$$

where $\tilde{G} := G \setminus i$ is the graph without node i , \mathbb{P}_G denotes the probability when the graph is G , and similarly for $\mathbb{P}_{\tilde{G}}$. The second inequality follows from the induction assumption, and the fact that if α is the decay coefficient for G , it is also for \tilde{G} . Taking a union bound over $k \in \mathcal{S}_i$ now gives us the statement of the theorem for G :

$$\begin{aligned} \mathbb{P}_G [T_i = t] &= \sum_{k \in \mathcal{S}_i} \mathbb{P}_G [k \text{ infects } i \text{ at time } t] \\ &\leq (1 - \alpha)^{t-2} p_{\text{init}} \sum_{k \in \mathcal{S}_i} p_{ki} \\ &\leq (1 - \alpha)^{t-1} p_{\text{init}} \end{aligned}$$

The bounds on $\mathbb{P}[T_i < \infty]$ follow simply from summing this geometric series. □

B Maximum Likelihood

B.1 Proof of Prop. 1

Let $X_i(\tau) = 0$ if i is susceptible at time τ , 1 if i is active at time τ and 2 if i is inactive at time τ . Let $X(\tau)$, $\tau = 0, \dots, n$ be the corresponding vector process. Note that $X(\tau)$ is a Markov process, and there

is a one to one correspondence between the set of infection times t and sample path $x(\tau)$ of the process $X(\tau)$.

Given t , let $x^0(\tau)$ be the corresponding vector process. In particular,

$$x_i^0(\tau) = \begin{cases} 0 & \text{if } \tau < t_i \\ 1 & \text{if } \tau = t_i \\ 2 & \text{if } \tau > t_i \end{cases}$$

Then,

$$\begin{aligned} \mathbb{P}_\theta [T = t] &= \mathbb{P}_\theta [X(\tau) = x^0(\tau) \text{ for } \tau = 0, \dots, n] \\ &= \mathbb{P}_\theta [X(0) = x^0(0)] \times \\ &\quad \prod_{\tau=1}^n \mathbb{P}_\theta [X(\tau) = x^0(\tau) | X(\tau-1) = x^0(\tau-1)] \end{aligned}$$

Now, $\mathbb{P}_\theta [X(0) = x^0(0)] = p_{\text{init}}^s (1 - p_{\text{init}})^{n-s}$. Also,

$$\begin{aligned} &\mathbb{P}_\theta [X(\tau) = x^0(\tau) | X(\tau-1) = x^0(\tau-1)] \\ &= \prod_{i \in V} \mathbb{P}_\theta [X_i(\tau) = x_i^0(\tau) | X(\tau-1) = x^0(\tau-1)] \end{aligned}$$

because each node gets infected independently from each of its currently active neighbors. Thus we have that

$$\mathbb{P} [T = t] = p_{\text{init}}^s (1 - p_{\text{init}})^{n-s} \prod_{i \in V} \left(\prod_{\tau=1}^n a_i(\tau) \right) \quad (5)$$

where $a_i(\tau) = \mathbb{P}_\theta [X_i(\tau) = x_i^0(\tau) | X(\tau-1) = x^0(\tau-1)]$. It is clear that for $\tau > t_i$, $a_i(\tau) = 1$. For $\tau = t_i$, $a_i(\tau)$ is the probability that at least one of its active nodes at time $t_i - 1$ infected node i . Thus,

$$a_i(t_i) = 1 - \prod_{j: t_j = t_i - 1} \exp(-\theta_{ji}) \quad (6)$$

Finally, for each $\tau < t_i$, $a_i(\tau)$ is the probability that active nodes at time $\tau - 1$ failed to infect node i . The set of all nodes that were active but failed to infect susceptible node i is $\{j : t_j \leq t_i - 2\}$. So we have

$$\prod_{\tau < t_i} a_i(\tau) = \prod_{j: t_j \leq t_i - 2} \exp(-\theta_{ji}) \quad (7)$$

Putting (5), (6) and (7) together and taking log gives the result.

Concavity follows from the fact that $\log(1 - \exp(-x))$ is a concave function of x , and the fact that if any function $f(x)$ is a concave function of x then $f(\sum_i \theta_i)$ is jointly concave in θ . ■

B.2 Proof of Theorem 1

We focus on the recovery of the neighborhood of node i . For brevity, we will drop i from sub-scripts; thus we denote θ_{*i} by θ , \mathcal{V}_i by \mathcal{V} and \mathcal{S}_i by \mathcal{S} , and d_i, D_i by d, D . Let θ^* be the true parameter values. Define the empirical log-likelihood function by

$$\widehat{L}(\theta) := \frac{1}{m} \sum_u \mathcal{L}_i(t^u; \theta)$$

Note that the ML algorithm finds $\widehat{\theta} = \operatorname{argmax}_{\theta} \widehat{L}(\theta)$. Also let $L(\theta) := \mathbb{E}_{\theta^*} [\mathcal{L}_i(T, \theta)]$.

Idea: Note that as the number of samples m increases, $\widehat{L} \rightarrow L$. Also, we know that $\theta^* = \operatorname{arg min}_{\theta} L(\theta)$; this is just stating that the expected value of the likelihood function is maximized by the true parameter values, a simple classical result from ML estimation [19]. Thus when $\widehat{L} \simeq L$, their minimizers will also be close; i.e. $\theta^* \simeq \widehat{\theta}$. However, they will not be exactly equal; hence hope then is to have subsequent thresholding find the significant edges. The challenge is in establishing non-asymptotic bounds that show that m scales much slower than n (the network size) or D (the size of the super-neighborhood).

Roadmap to the proof:

(a) In Proposition 3 we provide an expression for the gradient $\nabla_j L(\theta^*)$ of the expected log-likelihood evaluated at the true parameters θ^* . This can be used to show that $\nabla_j L(\theta^*) = 0$ for the true neighbors $j \in \mathcal{V}$, and for the others we can show that $\nabla_j L(\theta^*) < 0$ for $j \notin \mathcal{V}$.

(b) Note that if we had similar relationships hold for the empirical likelihood, i.e. if $\nabla_j \widehat{L}(\widehat{\theta}) = 0$ for $j \in \mathcal{V}$ and $\nabla_j \widehat{L}(\widehat{\theta}) < 0$ for $j \notin \mathcal{V}$, then we would be done; this is because by complementary slackness conditions we would have that $\widehat{\theta}_j > 0$ for $j \in \mathcal{V}$ and $\widehat{\theta}_j = 0$ otherwise: the non-zero $\widehat{\theta}_j$ would then correspond to the true neighborhood. Of course, these relationships do not hold exactly; the rest of the proof is showing they hold approximately, and the neighborhood can be found by thresholding.

(c) As a first step to analyzing $\nabla_j \widehat{L}(\widehat{\theta})$, in Lemma 3 we establish concentration results showing that an intermediate quantity $\nabla_j \widehat{L}(\theta^*)$ is close to $\nabla_j L(\theta^*)$, and hence we can show that $|\nabla_j \widehat{L}(\theta^*)| < a$ for $j \in \mathcal{V}$ (i.e. the gradient is small for the true neighbors), and $\nabla_j \widehat{L}(\theta^*) < -b$ for $j \notin \mathcal{V}$ (i.e. the gradient is negative for the others). Here a and b depend on the system parameters, and a depends on the threshold η as well, with $a \rightarrow 0$ as $\eta \rightarrow 0$. This latter dependence is important as it shows that once the number of samples m becomes large, we can choose η small and get exact recovery.

(d) In Lemma 4, we provide an upper bound on the value of $\widehat{\theta}_j$ for $j \in \mathcal{V}$. We need this to not be too large for the next step.

(e) In Lemma 5 we derive an upper bound on the total value $\sum_{j \notin \mathcal{V}} \widehat{\theta}_j$ of the non-neighbor parameters in $\widehat{\theta}$. This upper bound implies that no non-neighbors will be selected after thresholding at η , completing the proof of the first claim of the theorem.

(f) Finally, in Lemma 6 we show that, for true neighbors $j \in \mathcal{V}$, if the true $p_{ji}^* > \frac{8}{\alpha}(e^{2\eta} - 1)$ then $\widehat{\theta}_j > \eta$, and will thus be estimated to be in the true neighborhood. This completes the proof of the second claim of the theorem.

Proposition 3.

$$\nabla_j L(\theta^*) = -\mathbb{P}[T_i > T_j ; T_k \neq T_j \quad \forall k \in \mathcal{V}] \quad (8)$$

Proof. Taking the derivative of $L(\cdot)$ with respect to θ_j , we obtain

$$\nabla_j L(\theta) = \mathbb{E} \left[-\mathbb{1}_{\{T_j \leq T_{i-2}\}} + \frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_k\right) - 1} \right]$$

Let \mathcal{F}_{T_j} be the sigma algebra with information up to the (random) time T_j . By iterated conditioning, we obtain

$$\nabla_j L(\theta^*) = -\mathbb{E} \left[\mathbb{E} \left[\mathbb{1}_{\{T_j \leq T_{i-2}\}} - \frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_{ki}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] \right] \quad (9)$$

Since the event $\{T_i \leq T_j\}$ is measurable in \mathcal{F}_{T_j} , we have

$$\mathbb{E} \left[\mathbb{1}_{\{T_j \leq T_{i-2}\}} - \frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_{ki}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] = 0 \text{ if } T_i \leq T_j \quad (10)$$

On the other hand, if $\{T_i > T_j\}$, we have

$$\begin{aligned} & \mathbb{E} \left[\mathbb{1}_{\{T_j \leq T_{i-2}\}} - \frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_{si}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] \\ &= \mathbb{P}[T_i \geq T_j + 2 \mid \mathcal{F}_{T_j}] - \mathbb{E} \left[\frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_{ki}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] \end{aligned}$$

Considering the two terms above separately, we see that

$$\mathbb{P}[T_i \geq T_j + 2 \mid \mathcal{F}_{T_j}] = \exp\left(-\sum_{k: T_k = T_j} \theta_{ki}^*\right)$$

which follows from the fact that the probability that (active) j failed to infect (susceptible) i is equal to the probability that all the nodes that were active at T_j failed to infect i . For the second term, we have

$$\begin{aligned} \mathbb{E} \left[\frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_{i-1}} \theta_{ki}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] &= \mathbb{E} \left[\frac{\mathbb{1}_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k: T_k = T_j} \theta_{ki}^*\right) - 1} \middle| \mathcal{F}_{T_j} \right] \\ &\stackrel{(\S_1)}{=} \frac{1}{\exp\left(\sum_{k: T_k = T_j} \theta_{ki}^*\right) - 1} \mathbb{E} \left[\mathbb{1}_{\{T_j = T_{i-1}\}} \middle| \mathcal{F}_{T_j} \right] \\ &\stackrel{(\S_2)}{=} \frac{\left(1 - \exp\left(-\sum_{k: T_k = T_j} \theta_{ki}^*\right)\right) \mathbb{1}_{\{\exists k \in \mathcal{V} \text{ s.t. } T_k = T_j\}}}{\exp\left(\sum_{k: T_k = T_j} \theta_{ki}^*\right) - 1} \\ &= \exp\left(-\sum_{k: T_k = T_j} \theta_{ki}^*\right) \mathbb{1}_{\{\exists k \in \mathcal{V} \text{ s.t. } T_k = T_j\}} \end{aligned}$$

where (ς_1) follows from the fact that $\{k : T_k = T_j\}$ is measurable in \mathcal{F}_{T_j} and (ς_2) follows from the fact that $T_i = T_j + 1$ if and only if at least one of the parents of i were active at T_j and succeeded in infecting i . Combining the above two equations, we obtain

$$\mathbb{E} \left[\mathbb{1}_{\{T_j \leq T_i - 2\}} - \frac{\mathbb{1}_{\{T_j = T_i - 1\}}}{\exp \left(\sum_{k: T_k = T_i - 1} \theta_{si}^* \right) - 1} \middle| \mathcal{F}_{T_j} \right] = \mathbb{1}_{\{T_k \neq T_j \forall k \in \mathcal{V}\}} \text{ if } T_i > T_j \quad (11)$$

Combining (9), (10) and (11)

$$\nabla_j L(\theta^*) = -\mathbb{P}[T_i > T_j; T_k \neq T_k \forall k \in \mathcal{V}] \quad (12)$$

□

An easy corollary of Proposition 3 is that if j is a parent of i , then the gradient with respect to θ_j is zero since the probability above needs none of the parents of i to be infected at the same time as j . On the other hand, if j is not a parent of i , the gradient is strictly negative since the probability on the right hand side is strictly positive.

$$\nabla_j L(\theta^*) = 0 \text{ if } j \in \mathcal{V} \quad (13)$$

$$\nabla_j L(\theta^*) < 0 \text{ if } j \notin \mathcal{V} \quad (14)$$

We now state our concentration results. For any j , let $\nabla_j \widehat{L}(\theta)$ be the partial derivative of $\widehat{L}(\theta)$ with respect to θ_j . For $j \in \mathcal{V}$, let

$$m_{1,j} := |\{u : t_j^u = t_i^u - 1 \ \& \ t_k^u \neq t_i^u - 1 \ \forall k \in \mathcal{V} \setminus j\}|$$

be the number of cascades where j is the sole infector of node i and

$$m_{2,j} := |\{u : t_j^u \leq t_i^u - 2\}|$$

be the number of cascades where j is infected at least two time units before i .

Lemma 3. For $m > \frac{c}{p_{init}} \left(\frac{1}{\alpha^7 \eta^2 p_{i,min}^2} \right) d_i^2 \log \left(\frac{D_i}{\delta} \right)$, we have that

$$(a) \quad \left| \nabla_j \widehat{L}(\theta^*) \right| < a \text{ for } j \in \mathcal{V} \text{ where } a := \frac{\alpha^3 \eta p_{init}}{144d}$$

$$(b) \quad \nabla_j \widehat{L}(\theta^*) < -b \text{ for } j \notin \mathcal{V} \text{ where } b := \frac{\alpha p_{init}}{16}$$

$$(c) \quad \xi_1 p_j^* < m_{1,j} < \bar{\xi}_1 \text{ for } j \in \mathcal{V} \text{ where } \xi_1 := \frac{c}{4} \log \frac{D}{\delta}, \bar{\xi}_1 := \frac{2c}{\alpha} \log \frac{D}{\delta} \text{ and } p_j^* := 1 - \exp(-\theta_j^*)$$

$$(d) \quad \xi_2 < m_{2,j} < \bar{\xi}_2 \text{ for } j \in \mathcal{V} \text{ where } \xi_2 := \frac{c\alpha}{4} \log \frac{D}{\delta} \text{ and } \bar{\xi}_2 := \frac{2c}{\alpha} \log \frac{D}{\delta}$$

with probability greater than $1 - \delta$.

Proof. For simplicity of notation we denote the number of samples as $m = \frac{C \log \frac{D}{\delta}}{p_{\text{init}}}$ where $C = \frac{cd_i^2}{\alpha^7 \eta^2 p_{i,\text{min}}^2}$ and $D = D_i$. We will first prove (c). First, we note the following bounds for independent Bernoulli random variables X_l where μ is the mean of the sum of X_l .

$$\mathbb{P} \left[\sum_l X_l < (1 - \kappa)\mu \right] < \left(\frac{\exp(-\kappa)}{(1 - \kappa)^{(1 - \kappa)}} \right)^\mu \quad (15)$$

$$\mathbb{P} \left[\sum_l X_l > (1 + \kappa)\mu \right] < \left(\frac{e^{\frac{\kappa}{1 + \kappa}}}{1 + \kappa} \right)^{(1 + \kappa)\mu} \quad (16)$$

So as to be able to use the above inequalities, we first establish bounds on the expected value of $m_{1,j}$.

$$\mathbb{E}[m_{1,j}] \geq mp_{\text{init}}(1 - p_{\text{init}})^d p_j^* \geq 2\xi_1 p_j^*$$

where the bound uses the probability that j is infected at time 0 and neither i nor any of its other neighbors are infected at time 0 and j infects i at time 1. Similarly, we have

$$\mathbb{E}[m_{1,j}] \leq m\mathbb{P}[T_j < \infty] \leq \frac{\bar{\xi}_1}{2}$$

where we use Lemma 1. Now applying (15) to $m_{1,j}$ we obtain

$$\mathbb{P} \left[m_{1,j} < (1 - \frac{1}{2})2\xi_1 p_j^* \right] < \left(\frac{\exp(-\frac{1}{2})}{(\frac{1}{2})^{\frac{1}{2}}} \right)^{2\xi_1 p_j^*} < \frac{\delta}{8D}$$

Similarly applying (16) to $m_{1,j}$ gives us

$$\mathbb{P} \left[m_{1,j} > (1 + 1)\frac{\bar{\xi}_1}{2} \right] < \left(\frac{\sqrt{e}}{2} \right)^{\bar{\xi}_1} < \frac{\delta}{8D}$$

This proves (c). The proof of (d) is similar.

We will now prove (a). Fix any $j \in \mathcal{V}$. Let $\mathcal{U}_j = \{u \in \mathcal{U} : T_j^u < \infty\}$. Since $\mathbb{E}[|\mathcal{U}_j|] \geq p_{\text{init}}m = C \log \frac{D}{\delta}$, using (15), we obtain

$$\mathbb{P} \left[|\mathcal{U}_j| < \frac{C \log \frac{D}{\delta}}{2} \right] < \frac{\delta}{16D} \quad (17)$$

Similarly since $\mathbb{E}[|\mathcal{U}_j|] \leq \frac{p_{\text{init}}}{\alpha}m = \frac{C}{\alpha} \log \frac{D}{\delta}$, using (16), we obtain

$$\mathbb{P} \left[|\mathcal{U}_j| > \frac{2C \log \frac{D}{\delta}}{\alpha} \right] < \frac{\delta}{16D} \quad (18)$$

Define the random variable

$$Z_j = -1_{\{T_j \leq T_{i-2}\}} + \frac{1_{\{T_j = T_{i-1}\}}}{\exp\left(\sum_{k:T_k = T_{i-1}} \theta_k^*\right) - 1}$$

Note that we have the following absolute bound on Z_j

$$|Z_j| < 1 + \frac{1}{\exp(\theta_j^*) - 1} = \frac{1}{p_j^*} \quad (19)$$

where $p_j^* = 1 - \exp(-\theta_j)$ and also

$$\nabla_j \widehat{L}(\theta^*) = \frac{1}{m} \sum_{u \in \mathcal{U}} Z_j^u = \frac{1}{m} \sum_{u \in \mathcal{U}_j} Z_j^u$$

where Z_j^u is the realization of Z_j on infection u .

$$\mathbb{P} \left[\left| \nabla_j \widehat{L}(\theta^*) \right| \geq a \right] = \mathbb{P} \left[\frac{1}{m} \left| \sum_{u \in \mathcal{U}} Z_j^u \right| \geq a \right] = \mathbb{P} \left[\left| \sum_{u \in \mathcal{U}} Z_j^u \right| \geq ma \right]$$

At this point we could apply Azuma-Hoeffding inequality to bound the above probability. However, the scaling factor in the exponent will be ma^2 which gives us an extra p_{init} . To avoid this, we bound the above quantity as follows:

$$\begin{aligned} & \mathbb{P} \left[\left| \sum_{u \in \mathcal{U}} Z_j^u \right| \geq ma \right] \\ & \leq \mathbb{P} \left[|\mathcal{U}_j| > \frac{2C \log \frac{D}{\delta}}{\alpha} \text{ or } |\mathcal{U}_j| < \frac{C \log \frac{D}{\delta}}{2} \right] + \sum_{s=\frac{C \log \frac{D}{\delta}}{2}}^{\frac{2C \log \frac{D}{\delta}}{\alpha}} \mathbb{P} \left[|\mathcal{U}_j| = s; \left| \sum_{u \in \mathcal{U}} Z_j^u \right| \geq ma \right] \\ & \stackrel{(\varsigma_1)}{\leq} \frac{\delta}{8D} + \sum_{s=\frac{C \log \frac{D}{\delta}}{2}}^{\frac{2C \log \frac{D}{\delta}}{\alpha}} \sum_{U_j: |U_j|=s} \mathbb{P} [U_j = U_j] \mathbb{P} \left[\left| \sum_{u \in U_j} Z_j^u \right| \geq ma \mid U_j = U_j \right] \end{aligned} \quad (20)$$

where U_j varies over all the subsets of \mathcal{U} and (ς_1) follows from (17) and (18). Focusing on the last term, we first note that Z_j^u are still independent random variables for $u \in U_j$. Since $\mathbb{E}[Z_j] = 0$ from (13), we can apply Azuma-Hoeffding inequality and using (19) we obtain

$$\mathbb{P} \left[\left| \sum_{u \in U_j} Z_j^u \right| \geq ma \mid U_j = U_j, |U_j| = s \right] \leq 2 \exp \left(\frac{-(ma)^2}{2s \left(\frac{1}{p_j^*} \right)^2} \right) < \frac{\delta}{16D} \quad (21)$$

where (ς_1) follows from the fact that $s \leq \frac{2C \log \frac{D}{\delta}}{\alpha}$. The proof of (b) is on the same lines after noting that for any $j \notin \mathcal{V}$,

$$\begin{aligned} \mathbb{E}[Z_j] = \nabla_j L(\theta^*) & \stackrel{(\varsigma_1)}{=} -\mathbb{P}[T_i > T_j; T_j \neq T_k \forall k \in \mathcal{V}] \\ & \stackrel{(\varsigma_2)}{<} -p_{\text{init}} (1 - p_{\text{init}})^{d+1} \stackrel{(\varsigma_3)}{<} -\frac{p_{\text{init}}}{2} \end{aligned} \quad (22)$$

where (ς_1) follows from Proposition 3, (ς_2) follows from the fact that the probability when j is infected before i and none of the parents of i are infected at the same time can be lower bounded by the case where

j is infected at time 0 and neither i nor any of its parents are infected at time 0. (ς_3) follows from the assumption that $p_{\text{init}} < \frac{1}{2d}$ and hence $(1 - p_{\text{init}})^{d+1} > \frac{1}{2}$. Using (22) and Lemma 1, we obtain

$$\mathbb{E}[Z_j \mid T_j < \infty] = \frac{\mathbb{E}[Z_j] - \mathbb{E}[Z_j \mathbf{1}_{\{T_j = \infty\}}]}{\mathbb{P}[T_j < \infty]} \leq \frac{-\alpha}{2} \quad (23)$$

Using (20) it suffices to show that

$$\mathbb{P}\left[\sum_{u \in U_j} Z_j^u \geq -mb \mid \mathcal{U}_j = U_j, |U_j| = s\right] < \frac{\delta}{16D}$$

for $\frac{C \log \frac{D}{\delta}}{2} \leq s \leq \frac{2C \log \frac{D}{\delta}}{\alpha}$. An application of Azuma-Hoeffding inequality gives us the required bound as follows.

$$\begin{aligned} & \mathbb{P}\left[\sum_{u \in U_j} Z_j^u \geq -mb \mid \mathcal{U}_j = U_j, |U_j| = s\right] \\ & \stackrel{(\varsigma_1)}{=} \mathbb{P}\left[\sum_{u \in U_j} Z_j^u - s\mathbb{E}[Z_j] \geq \frac{-C\alpha \log \frac{D}{\delta}}{16} - s\mathbb{E}[Z_j] \mid \mathcal{U}_j = U_j, |U_j| = s\right] \\ & \stackrel{(\varsigma_2)}{\leq} \mathbb{P}\left[\sum_{u \in U_j} Z_j^u - s\mathbb{E}[Z_j] \geq \frac{C\alpha \log \frac{D}{\delta}}{8} \mid \mathcal{U}_j = U_j, |U_j| = s\right] \\ & \stackrel{(\varsigma_3)}{\leq} \exp\left(\frac{\left(\frac{C\alpha \log \frac{D}{\delta}}{8}\right)^2}{2\left(\frac{2C \log \frac{D}{\delta}}{\alpha}\right)\left(\frac{1}{p_j^*}\right)^2}\right) \leq \frac{\delta}{16D} \end{aligned}$$

where (ς_1) follows by subtracting $s\mathbb{E}[Z_j]$ from both sides of the inequality for which we are bounding the probability, (ς_2) follows from the fact that $s \geq \frac{C \log \frac{D}{\delta}}{2}$ and (23) and (ς_3) is an application of the Azuma-Hoeffding inequality using (19) and the fact that $s \leq \frac{2C \log \frac{D}{\delta}}{\alpha}$. \square

Lemma 4. *When (a)-(d) in Lemma 3 hold, $\max_{j \in \mathcal{V}} \hat{\theta}_j < \frac{\bar{\xi}_1}{\bar{\xi}_2}$*

Proof. Let $k = \operatorname{argmax}_{j \in \mathcal{V}} \hat{\theta}_j$. If $\hat{\theta}_k = 0$, we are done. So assume $\hat{\theta}_k > 0$. By the optimality of $\hat{\theta}$, we see that

$$\nabla_k \hat{L}(\hat{\theta}) = 0 \quad (24)$$

On the other hand, we have

$$\begin{aligned}
\nabla_k \widehat{L}(\widehat{\theta}) &= \frac{1}{m} \left(-m_{2,k} + \sum_u 1_{\{t_i^u < \infty\}} \left(\exp \left(\sum_{j:t_j^u = t_i^u - 1} \widehat{\theta}_j \right) - 1 \right)^{-1} \right) \\
&\stackrel{(\varsigma_1)}{\leq} \frac{1}{m} \left(-m_{2,k} + \frac{1}{\exp(\widehat{\theta}_k) - 1} m_{1,k} \right) \\
&\stackrel{(\varsigma_2)}{\leq} \frac{1}{m} \left(-\xi_2 + \frac{1}{\exp(\widehat{\theta}_k) - 1} \bar{\xi}_1 \right) \leq \frac{1}{m} \left(-\xi_2 + \frac{1}{\widehat{\theta}_k} \bar{\xi}_1 \right)
\end{aligned} \tag{25}$$

where (ς_1) follows from the definition of $m_{1,k}$ and the fact that on the infections corresponding to $m_{1,k}$, we have

$$\sum_{j:t_j^u = t_i^u - 1} \widehat{\theta}_j \geq \widehat{\theta}_k$$

and (ς_2) follows from Lemma 3. Putting (24) and (25) together, we obtain the result. \square

Lemma 5. *When (a)-(d) in Lemma 3 hold, $\sum_{j \notin \mathcal{V}} \widehat{\theta}_j \leq \frac{ad}{b} \left(\frac{\bar{\xi}_1}{\xi_2} + \log \frac{1}{\alpha} \right) < \eta$*

Proof. Since $\widehat{L}(\theta)$ is concave, the subgradient condition at θ^* gives us the following

$$\begin{aligned}
\widehat{L}(\widehat{\theta}) - \widehat{L}(\theta^*) &\leq \left\langle \nabla \widehat{L}(\theta^*), \widehat{\theta} - \theta^* \right\rangle \\
&\stackrel{(\varsigma_1)}{=} \left\langle \nabla_{\mathcal{V}^c} \widehat{L}(\theta^*), \widehat{\theta}_{\mathcal{V}^c} \right\rangle + \left\langle \nabla_{\mathcal{V}} \widehat{L}(\theta^*), \widehat{\theta}_{\mathcal{V}} - \theta_{\mathcal{V}}^* \right\rangle \\
&\stackrel{(\varsigma_2)}{\leq} -b \|\widehat{\theta}_{\mathcal{V}^c}\|_1 + a \|\widehat{\theta}_{\mathcal{V}} - \theta_{\mathcal{V}}^*\|_1 \\
&\leq -b \|\widehat{\theta}_{\mathcal{V}^c}\|_1 + ad \left(\|\widehat{\theta}_{\mathcal{V}}\|_{\infty} + \|\theta_{\mathcal{V}}^*\|_{\infty} \right)
\end{aligned} \tag{26}$$

where (ς_1) follows from the fact that $\theta_{\mathcal{V}^c}^* = 0$ and (ς_2) follows from the fact that $\widehat{\theta} > 0$ and Lemma 3. The optimality of $\widehat{\theta}$ gives us

$$\widehat{L}(\widehat{\theta}) - \widehat{L}(\theta^*) \geq 0 \tag{27}$$

Finally we have the following bound on $\|\theta_{\mathcal{V}}^*\|_{\infty}$:

$$\theta_j^* = -\log(1 - p_j^*) \leq \log \frac{1}{\alpha} \tag{28}$$

Using (26), (27), (28) and Lemma 4 proves the first inequality, that $\sum_{j \notin \mathcal{V}} \widehat{\theta}_j \leq \frac{ad}{b} \left(\frac{\bar{\xi}_1}{\xi_2} + \log \frac{1}{\alpha} \right)$. The second inequality, that $\frac{ad}{b} \left(\frac{\bar{\xi}_1}{\xi_2} + \log \frac{1}{\alpha} \right) < \eta$, is easy to see. \square

Lemma 6. When (a)-(d) in Lemma 3 hold, for every $j \in \mathcal{V}$ we have that $\hat{\theta}_j > \log\left(1 + \frac{p_j^* \xi_1}{\xi_2}\right) - \eta$ where $p_j^* = 1 - \exp(\theta_j^*)$.

Proof. Since $\hat{\theta}_j \geq 0$, by the optimality of $\hat{\theta}$ we have

$$\nabla_j \hat{L}(\hat{\theta}) \leq 0 \quad (29)$$

On the other hand, we have the following bound on the gradient

$$\begin{aligned} \nabla_j \hat{L}(\hat{\theta}) &= \frac{1}{m} \left(-m_{2,j} - \sum_u 1_{\{t_i^u < \infty\}} \left(\exp \left(\sum_{k:t_k^u = t_i^u - 1} \hat{\theta}_k \right) - 1 \right)^{-1} \right) \\ &\stackrel{(\varsigma_1)}{\geq} \frac{1}{m} \left(-m_{2,j} + \frac{1}{\exp(\hat{\theta}_j + \|\hat{\theta}_{\mathcal{V}^c}\|_1) - 1} m_{1,k} \right) \\ &\stackrel{(\varsigma_2)}{\geq} \frac{1}{m} \left(-\bar{\xi}_2 + \frac{1}{\exp(\hat{\theta}_j + \|\hat{\theta}_{\mathcal{V}^c}\|_1) - 1} p_j^* \xi_1 \right) \end{aligned} \quad (30)$$

where (ς_1) follows from the fact that on the infections corresponding to $m_{1,k}$, we have

$$\sum_{k:t_k^u = t_i^u - 1} \hat{\theta}_k \leq \hat{\theta}_j + \|\hat{\theta}_{\mathcal{V}^c}\|_1$$

and (ς_2) follows from Lemma 3. Combining (29), (30) and Lemma 5 gives us the result. \square

Thus we see that if the true parameter $p_{j_i}^* > \frac{8}{\alpha}(e^{2\eta} - 1)$, then $\hat{\theta}_j > \eta$ and thus will be in the estimated neighborhood $\hat{\mathcal{N}}_i$. This completes the proof of Theorem 1.

C Greedy algorithm

C.1 Proof of Theorem 2

To simplify notation, we again denote \mathcal{V}_i by \mathcal{V} , \mathcal{S}_i by \mathcal{S} and so on. From Lemma 1, we have that for every node j ,

$$\mathbb{P}[T_j < \infty] < \frac{p_{\text{init}}}{\alpha}$$

Since the graph is a tree, for every node j there exists a unique (undirected) path between i and j . All the nodes on this path are said to be ancestors of j . Consider a node $j \in \mathcal{S} \setminus \mathcal{V}$. Let $k \in \mathcal{V}$ be the ancestor of j on this path. Then we have that

$$\mathbb{P}[T_j \neq T_k; T_k = T_i - 1] \geq p_{\text{init}} (1 - p_{\text{init}})^2 p_{\text{min}}$$

If $l \in \mathcal{V}$ but is not an ancestor of j then

$$\mathbb{P}[T_j = T_l = T_i - 1] < \mathbb{P}[T_j < \infty]\mathbb{P}[T_l < \infty] < \left(\frac{p_{\text{init}}}{\alpha}\right)^2$$

since T_j and T_l are independent conditioned on $T_j, T_l < T_i$. For any event A that depends on the infection times, let $N(A)$ denote the number of cascades in \mathcal{U} in which event A has occurred. Using (15) and (16), we have the following bounds on probabilities of error events:

$$\begin{aligned} \mathbb{P}\left[N(T_k = T_i - 1) \leq \left(1 - \frac{1}{2}\right) mp_{\text{min}}p_{\text{init}}(1 - p_{\text{init}})\right] &< \left(\frac{2}{e}\right)^{\frac{mp_{\text{min}}p_{\text{init}}(1-p_{\text{init}})}{2}} \\ \mathbb{P}\left[N(T_k = T_l = T_i - 1) \geq \frac{mp_{\text{init}}p_{\text{min}}}{8d}\right] &< \left(\frac{em\left(\frac{p_{\text{init}}}{\alpha}\right)^2}{\left(\frac{mp_{\text{init}}p_{\text{min}}}{8d}\right)}\right)^{\frac{mp_{\text{init}}p_{\text{min}}}{8d}} \\ \mathbb{P}\left[N(T_j = T_l = T_i - 1) \geq \frac{mp_{\text{init}}p_{\text{min}}}{8d}\right] &< \left(\frac{em\left(\frac{p_{\text{init}}}{\alpha}\right)^2}{\left(\frac{mp_{\text{init}}p_{\text{min}}}{8d}\right)}\right)^{\frac{mp_{\text{init}}p_{\text{min}}}{8d}} \\ \mathbb{P}\left[N(T_j \neq T_k; T_k = T_i - 1) \leq \left(1 - \frac{1}{2}\right) mp_{\text{init}}p_{\text{min}}(1 - p_{\text{init}})^2\right] &< \left(\frac{2}{e}\right)^{mp_{\text{init}}p_{\text{min}}(1-p_{\text{init}})^2} \end{aligned} \quad (31)$$

where $k, l \in \mathcal{V}$ and $j \notin \mathcal{V}$ such that k is an ancestor of j . Substituting the value of m from the statement of Theorem 2 and recalling the assumption on p_{init} , we see that with probability greater than $1 - \delta$, we have

$$N(T_k = T_i - 1) > \frac{cd(1 - p_{\text{init}})\log\frac{D}{\delta}}{2} \quad (32)$$

$$N(T_k = T_l = T_i - 1) < \frac{c\log\frac{D}{\delta}}{8} \quad (33)$$

$$N(T_j = T_l = T_i - 1) < \frac{c\log\frac{D}{\delta}}{8} \quad (34)$$

$$N(T_j \neq T_k; T_k = T_i - 1) > \frac{cd(1 - p_{\text{init}})^2\log\frac{D}{\delta}}{2} \quad (35)$$

Note that the assumption on p_{init} also yields an upper bound of $\frac{1}{16}$ on p_{init} . Now we will show that under the above conditions, Algorithm 2 recovers the original graph exactly. Suppose in iteration s , the neighborhood is $s - 1$ of the correct parents and there is atleast one $k \in \mathcal{V}$, not in the current neighborhood. Let the current set of infections be U . Then from (32) and (33), we see that

$$\begin{aligned} N_U(T_k = T_i - 1) &> \frac{cd(1 - p_{\text{init}})\log\frac{D}{\delta}}{2} - d\frac{c\log\frac{D}{\delta}}{8} \\ &= \frac{cd\log\frac{D}{\delta}}{8}(4(1 - p_{\text{init}}) - 1) > 0 \end{aligned}$$

So there is atleast one node that will be added to the neighborhood. Now consider any $j \notin \mathcal{V}$. If the ancestor of j that is a parent of i has already been added to the neighborhood list, then from (34)

$$\begin{aligned} N_U(T_j = T_i - 1) &< d\frac{c\log\frac{D}{\delta}}{8} \\ &< (4(1 - p_{\text{init}}) - 1)\frac{cd\log\frac{D}{\delta}}{8} \\ &< N_U(T_k = T_i - 1) \end{aligned}$$

Suppose the ancestor of j that is a parent of i has not yet been added to the neighborhood of i . Without loss of generality, let k be the ancestor of j . Then,

$$\begin{aligned}
& N_U(T_k = T_i - 1) - N_U(T_j = T_i - 1) \\
&= N_U(T_j \neq T_k; T_k = T_i - 1) \\
&\quad - N_U(T_j = T_i = T_i - 1: l \neq k, l \in S) \\
&> \frac{cd(1 - p_{\text{init}})^2 \log \frac{D}{\delta}}{2} - 2d \frac{c \log \frac{D}{\delta}}{8} \\
&= \frac{cd \log \frac{D}{\delta}}{4} \left(2(1 - p_{\text{init}})^2 - 1 \right) > 0
\end{aligned}$$

Applying union bound over all nodes in the superneighborhood, we can conclude that all nodes in the superneighborhood satisfy (32), (33), (34) and (35) with probability greater than $1 - \delta$. This proves Theorem 2.

D Lower Bounds

D.1 Proof of Lemma 2

Recall from Lemma 1 that $\mathbb{P}[T_i = t] \leq (1 - \alpha)^{t-1} p_{\text{init}}$. The proof just involves using this to bound $H(T_i)$. Since $p_{\text{init}} < \frac{1}{e}$, we have the following

$$\begin{aligned}
H(T_i) &= - \sum_{t=1}^n \mathbb{P}[T_i = t] \log \mathbb{P}[T_i = t] \\
&\quad - \mathbb{P}[T_i = \infty] \log \mathbb{P}[T_i = \infty] \\
&\leq - \sum_{t=1}^n (1 - \alpha)^{t-1} p_{\text{init}} \log (1 - \alpha)^{t-1} p_{\text{init}} \\
&\quad - \left(1 - \frac{p_{\text{init}}}{\alpha} \right) \log \left(1 - \frac{p_{\text{init}}}{\alpha} \right) \\
&\stackrel{(\zeta_1)}{\leq} \frac{p_{\text{init}}}{1 - \alpha} \left(\log \frac{1}{p_{\text{init}}} + \left(\frac{1 - \alpha}{\alpha} \right)^2 \log \frac{1}{1 - \alpha} \right) \\
&\quad - \left(1 - \frac{p_{\text{init}}}{\alpha} \right) \log \left(1 - \frac{p_{\text{init}}}{\alpha} \right)
\end{aligned}$$

where (ζ_1) follows from some algebraic manipulations.

E Generalized Independent Cascade Model

E.1 Proof of Prop. 2

Defining

$$x_i^0(\tau) = \begin{cases} 0 & \text{if } \tau < t_i \\ 1 & \text{if } \tau \geq t_i \end{cases}$$

and proceeding as in the proof of Proposition 1, we obtain

$$\mathbb{P}_\theta [T = t] = \mathbb{P}_\theta [X(0) = x^0(0)] \times \prod_{\tau=1}^n \mathbb{P}_\theta [X(\tau) = x^0(\tau) | X(0 : \tau - 1) = x^0(0 : \tau - 1)]$$

where $X(0 : \tau)$ denotes the (joint) values of the vectors $X(0), \dots, X(\tau)$. Now, $\mathbb{P}_\theta [X(0) = x^0(0)] = p_{\text{init}}^s (1 - p_{\text{init}})^{n-s}$. Also,

$$\mathbb{P}_\theta [X(\tau) = x^0(\tau) | X(0 : \tau - 1) = x^0(0 : \tau - 1)] = \prod_{i \in V} \mathbb{P}_\theta [X_i(\tau) = x_i^0(\tau) | X(0 : \tau - 1) = x^0(0 : \tau - 1)]$$

because each node gets infected independently from each of its currently active neighbors. Thus we have that

$$\mathbb{P}[T = t] = p_{\text{init}}^s (1 - p_{\text{init}})^{n-s} \prod_{i \in V} \left(\prod_{\tau=1}^n b_i(\tau) \right) \quad (36)$$

where $b_i(\tau) = \mathbb{P}_\theta [X_i(\tau) = x_i^0(\tau) | X(0 : \tau - 1) = x^0(0 : \tau - 1)]$. It is clear that for $\tau > t_i$, $b_i(\tau) = 1$. For $\tau = t_i$, $b_i(\tau)$ is the probability that at least one of the parents j of i infected before t_i infected node i at time t_i given that j did not infect i before t_i . Thus,

$$\begin{aligned} b_i(t_i) &= 1 - \prod_{j:t_j < t_i} \frac{1 - \sum_{r \in [t_i]} p_{ji}^r}{1 - \sum_{r \in [t_i - 1]} p_{ji}^r} \\ &= 1 - \prod_{j:t_j < t_i} \exp\left(-\theta_{ji}^{t_i - t_j}\right) \end{aligned} \quad (37)$$

Finally, for each $\tau < t_i$, $b_i(\tau)$ is the probability that active nodes at time $\tau - 1$ failed to infect node i . The set of all nodes that were active but failed to infect susceptible node i is $\{j : t_j \leq t_i - 2\}$. Each such node j failed to infect i for $t_i - t_j - 1$ time slots. So we have

$$\begin{aligned} \prod_{\tau < t_i} b_i(\tau) &= \prod_{j:t_j \leq t_i - 2} \left(1 - \sum_{r \in [t_i - t_j - 1]} p_{ji}^r \right) \\ &= \prod_{j:t_j \leq t_i - 2} \prod_{r \in [t_i - t_j - 1]} \exp\left(-\theta_{ji}^r\right) \end{aligned} \quad (38)$$

Putting (36), (37) and (38) together and taking log gives the result.

Concavity again follows from the fact that $\log(1 - \exp(-x))$ is a concave function of x , and the fact that if any function $f(x)$ is a concave function of x then $f(\sum_i \theta_i)$ is jointly concave in θ . \square

F Markov Graphs and Causality

F.1 Proof of Theorem 4

We will show that $\mathbb{P}[T = t]$ can be written as a product of various factors where each factor depends only on $t_{\mathcal{V}_i}$ for some $i \in V$. Given any vector t , for every $i \in V$ define the infection vectors

$$x_i(\tau) = \begin{cases} 0 & \text{if } 0 \leq \tau < t_i^{(1)} \\ 1 & \text{if } t_i^{(1)} \leq \tau < t_i^{(2)} \\ 2 & \text{if } \tau \geq t_i^{(2)} \end{cases}$$

We can see that there is a one to one correspondence between valid time vectors t and valid infection vectors x . Using the above transformation, we can calculate the probability of a given time vector t as follows:

$$\begin{aligned} \mathbb{P}[T = t] &= \mathbb{P}[X = x] \\ &= \mathbb{P}[X(0) = x(0)] \times \prod_{s=1}^{\infty} \mathbb{P}[X(s) = x(s) \mid x(0 : s-1)] \\ &= \left(\prod_{i \in V} \mathbb{P}[X_i(0) = x_i(0)] \right) \times \prod_{s=1}^{\infty} \prod_{i \in V} \mathbb{P}[X_i(s) = x_i(s) \mid x_{\mathcal{V}_i}(0 : s-1)] \\ &= \left(\prod_{i \in V} \mathbb{P}[X_i(0) = x_i(0)] \right) \times \prod_{i \in V} \prod_{s=1}^{\infty} \mathbb{P}[X_i(s) = x_i(s) \mid x_{\mathcal{V}_i}(0 : s-1)] \\ &= \prod_{i \in V} f_i(t_{\mathcal{V}_i}) \end{aligned}$$

where $f_i(t_{\mathcal{V}_i}) = \mathbb{P}[X_i(0) = x_i(0)] \times \prod_{s=1}^{\infty} \mathbb{P}[X_i(s) = x_i(s) \mid x_{\mathcal{V}_i}(0 : s-1)]$.

□