Learning the Graph of Epidemic Cascades

Praneeth Netrapalli
Dept. of Electrical and Computer Eng.
The University of Texas at Austin
Austin, TX-78712
praneethn@utexas.edu

Sujay Sanghavi
Dept. of Electrical and Computer Eng.
The University of Texas at Austin
Austin, TX-78712
sanghavi@mail.utexas.edu

ABSTRACT

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

For the classic and popular "independent cascade" epidemics, we analytically establish sufficient conditions on the number of epidemics for both the global maximum-likelihood (ML) estimator, and a natural greedy algorithm to succeed with high probability. 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 epidemic samples required – in all cases – becomes independent of the network size n.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph algorithms; G.3 [**Probability and Statistics**]: Stochastic processes

General Terms

Epidemics, cascades, graph structure learning, theory

1. INTRODUCTION

Epidemic cascades: Initially developed as a way to study disease propagation, epidemic cascades have recently emerged as popular and useful models in a wide range of application areas. Examples include
(a) 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 [9, 1].
(b) social networks: epidemic cascades provide natural models for understanding both the consumption of online media (e.g. viral videos, news articles[8]) and spread of ideas and opinions (e.g. trending of topics and hashtags on Twitter/Facebook[16], keywords on blog networks[4])
(c) e-commerce: understanding epidemic cascades (and, in this case, finding influential nodes) is crucial to viral marketing [5], and predicting/optimizing uptake on social buying sites like Groupon, LivingSocial etc.
(d) security and reliability: epidemic cascades model both the spread of computer worms and malware [6], and cascading failures in infrastructure networks [7, 15] and complex organizations [12].

Structure Learning: The vast majority of work on epidemic cascades has focused on understanding how the graph structure of the network (e.g. power laws, small world, expansion etc.) affects the spread of epidemics. 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 sub-graphs 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 epidemic 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

SIGMETRICS'12, June 11–15, 2012, London, England, UK.

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. Ours is the first paper (that we know of) to do any performance analysis for graph learning for epidemics.

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 when we are given prior information of a super-graph. It is not possible to learn the graph from a single epidemic; we study the number of epidemics 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 epidemics we observe *decouples* into n convex problems – one for each node, and requiring as input only the infection times of that nodes' super-neighborhood (i.e., its neighborhood in the super-graph). Our main result here is to establish that for this efficient algorithm, the number of times a node i needs to be infected to find its size-d_i true neighborhood from size-D_i super-neighborhood with probability greater than $1-\delta$ is $O(d_i^2 \log \frac{D_i}{\delta})$, for a general graph.
(b) Greedy algorithm: We also consider a natural greedy algorithm which iteratively adds to the estimated neighborhood nodes that provide the best incremental explanation for the infections of a node. We show that if the graph is a tree, then this is able to find the true neighborhood with probability greater than $1-\delta$ with only $O(d \log \frac{D}{\delta})$ samples.
(c) Lower bounds: We establish an information-theoretic lower bound on the number of epidemics needed in a general setting, and specialize it to show that reliable recovery needs at least $\Omega(d \log \frac{D}{d})$ samples.

A nice feature of our results is that both the algorithms, and the lower bounds, 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. Similarly, the number of samples required to recover the neighborhood of a node depends only on the sizes of its own neighborhood and super-neighborhood. We can use union bound along with the node by node guarantees above to obtain sufficient conditions on the number of samples required for recovering the entire graph correctly. In the full version of this paper[11], we also establish the relationship between the graph of an epidemic and its Markov random field. Due to lack of space, we do not provide proofs of all the results in this paper. For complete proofs of all the results in this paper, please refer [11].

Directly related work: While structure learning from epidemics is an important primitive, there has been very little work investigating it:
(a) algorithms: A recent paper [14] 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 [10] shows max-likelihood estimation in the independent cascade model can be cast as a decoupled convex optimization problem (albeit a different one from ours). Another recent paper [13] shows that the likelihood function for the entire graph decouples into n likelihood functions, one for each node and that it is convex for some general models of epidemics. Though the paper shows that the maximum likelihood estimator is consistent, it does not analytically investigate the number of infections required for consistent estimation. *(b) performance:* On this, there has been no work we are aware of; indeed, this is the main focus of our paper.

2. SYSTEM MODEL

Most of the analytical results of this paper are for the classic and popular *independent cascade* model; in particular we will consider the simple one-step model first proposed in [3] and recently popularized by Kempe, Kleinberg and Tardos [5].

Standard independent cascade epidemic model [5]: 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. Let $\mathcal{V}_i := \{j : (j, i) \in E\}$ denote the set of parents of each node 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 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.

Observation model: For a 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 epidemic 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 epidemic u. We observe more than one epidemic 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*[1] 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 $j \in \mathcal{S}_i \backslash \mathcal{V}_i$. 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 neigh-

[1]For 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.

borhood $\mathcal{V}_i$, for some or all of the nodes i. Clearly, this is not possible when we only observe a single epidemic; we will thus be interested in learning the graph from *as few epidemics as possible.*

Note that multiple seeds begin each epidemic $u \in \mathcal{U}$; thus, for a single epidemic 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 epidemic 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 epidemic 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 an epidemic, then *that epidemic 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 epidemics 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.

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.*

3. MAXIMUM LIKELIHOOD

The graph learning problem can be interpreted as a parameter estimation problem: for each epidemic, 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

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

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

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 $\mathcal{L}_i(t_{\mathcal{S}_i};\theta_{*i})$ (which will hence forth be referred to as $\mathcal{L}_i(t;\theta_{*i})$ for ease of notation), 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 $|\mathcal{V}_i|$-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 epidemics 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.

Our main analytical result of this section is a characterization of the performance of this ML algorithm, in terms of the number of epidemics 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*

Algorithm 1 ML Algorithm for Node i

1: Find

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

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

2: Estimate the parental neighborhood to be

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

3: Output $\widehat{\mathcal{V}}_i$.

weakest parent. Assume $d_i p_{init} < \frac{1}{2}$. Then, for any $\delta > 0$, if the number of epidemics $m = |\mathcal{U}|$ satisfies

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

Then, with probability greater than $1-\delta$, the estimate $\widehat{\mathcal{V}}_i$ from the ML algorithm with threshold η will have
(a) *no false neighbors, i.e. $\hat{V}_i \subset \mathcal{V}_i$, and*
(b) *all strong enough neighbors: if $j \in \mathcal{V}_i$ and $p_{ji} > \frac{8}{\alpha}(e^{2\eta} - 1)$, then $j \in \widehat{\mathcal{V}}_i$ 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 $d_i, p_{init}, \alpha, p_{i,min}, \eta$ 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. $\widehat{\mathcal{V}}_i = \mathcal{V}_i$, by choosing the threshold $\eta \leq \frac{1}{2}\log(1 + \frac{\alpha p_{i,min}}{8})$ low enough, and the corresponding number of epidemic 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 epidemic 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.

4. GREEDY ALGORITHM

We now analyze the sample complexity of a simple iterative greedy algorithm – for the case when the graph is a tree[2]. The algorithm is of course defined for general graphs.

[2]We 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.

The idea is as follows: suppose we want to find the parents of node i from a given set of epidemics $\mathcal{U}$. In each epidemic 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 epidemics. It then *removes* those epidemics from further consideration (since they have been "accounted for") and proceeds as before on the remaining epidemics, stopping when all epidemics are exhausted.

Algorithm 2 Greedy Algorithm for Node i

1: Initialize unaccounted epidemics $U = \mathcal{U}$
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 epidemics : $U \leftarrow U \setminus \{u : t_k^u = t_i^u - 1\}$
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 $p_{init} < \frac{\alpha^2 p_{min}}{16 e d_i}$. If Algorithm 2 is given a super-neighbhorhood of size $D_i := |\mathcal{S}_i|$, then for any $\delta > 0$ if the number of samples satisfies*

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

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 epidemics 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 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 epidemics 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 epidemic 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 epidemics, 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 [2]. 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:

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

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 $H(G \mid \widehat{G})$.

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

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[2]. Combining the above two results, we obtain

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

□

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*

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

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*

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

PROOF. We have that

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

$$\log|\mathcal{B}_\gamma(G')| \leq \log\binom{\binom{n}{2}}{\gamma} \leq \gamma\log\frac{n^2}{\gamma}$$

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

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*

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

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,

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

where

$$\mathcal{B}_s(\widehat{G}) = \{\widetilde{G}\in\mathcal{G}_d : \widetilde{\mathcal{V}}_i \triangle \widehat{\mathcal{V}}_i \le s_i \;\forall\, i\in V\}$$

Note that in the second inequality we assume $s_i \le \frac{D}{2}$ because otherwise if $d < \frac{D}{2}$, we can choose $\widehat{\mathcal{V}}_i = \Phi$ and if $d \ge \frac{D}{2}$, 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. □

6. EXPERIMENTS

In this section, we will present experimental evaluations of both the ML and Greedy algorithms on synthetic and real world graphs.

6.1 Synthetic graphs

Grids: First, we present the results of both ML and Greedy algorithms on grids. For each grid size, edge parameters are chosen so as to satisfy the assumptions of Theorem 1. Using these parameters, independent infections are generated by simulating the independent cascade model on the graph. Both ML and Greedy algorithms are given these infections as input to obtain an estimate of the grid. A graph is said to be recovered if the output parent set of *each* node is the same as the parent set of that node in the original graph. Given m infections, a trial executes the algorithm with those infections. The probability of recovery for m infections is calculated empirically as the fraction of trials which recover the parent set of every node of the graph exactly out of a total of 20 trials. Figure 1 shows the probability of recovery of a graph versus the total number of infections. We can see that as the number of infections increases, both the algorithms recover the graph exactly with higher probability. We also note that the number of infections needed to obtain a given probability of exact recovery increases as the graph size increases. On the other hand, Figure 2 shows the probability of recovery versus the *average number of times a node is infected*. One remarkable fact to note is that the plots for recovery almost line up on top of each other for various problem sizes. This indicates that in a grid, the probability of recovery is dependent only on the average number of infections experienced by a node irrespective of the problem size.

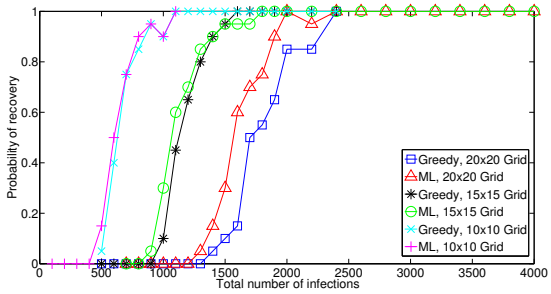

Figure 1: Recovery of grids with total number of infections: This plot shows the probability of recovery versus the total number of infections for various grid sizes.

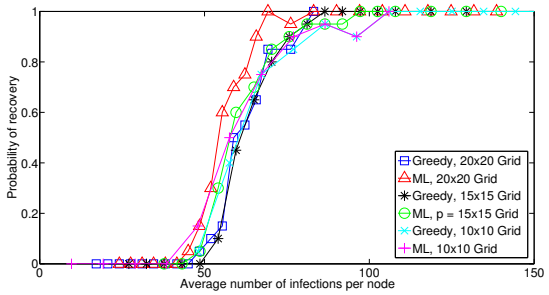

Figure 2: Recovery of grids with average infections per node: This plot is from the same experiment as Figure 1. The only (and crucial) difference is that the x-axis in this plot denotes the average number of times a node is infected.

Random Regular Graphs: We now present the results of ML and Greedy algorithms on random regular graphs of degree 4. A random regular graph is first sampled and appropriate values are chosen for the edge parameters. Independent infections are then sampled by simulating the independent cascade model. Figure 3 shows the plot of probability of recovery versus the average number of infections per node for different sizes of graphs for both the ML and Greedy algorithms. Figure 4 shows the comparison between the ML and Greedy algorithms when there is a super graph as opposed to when there is no super graph. The underlying graph is a random 4-regular 200 node graph. Both the ML and Greedy algorithms are run on varying number of infections. For each number of infections, two versions of

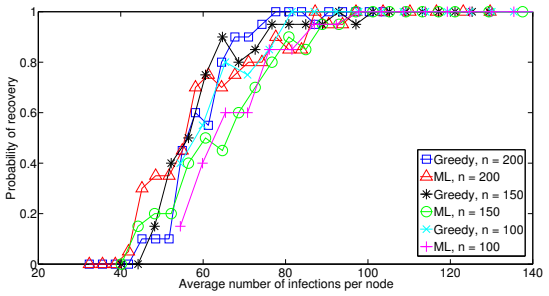

Figure 3: Random 4-regular graph: This figure plots the probability of recovery versus the average number of infections per node for both the ML and Greedy algorithms for different sizes of random regular graphs.

each algorithm were executed - one that has a super graph and the other that does not have a super graph. The y-axis denotes the fraction of those graphs that the algorithm recovered exactly. The x-axis denotes the average number of times a node is infected. Our analytical results suggest that the non availability of a super graph does not affect the sample complexity drastically. We can see that the above plot corroborates this claim.

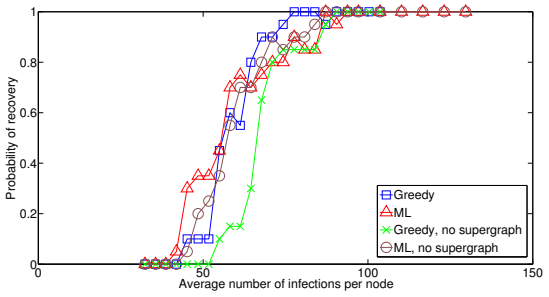

Figure 4: Super graph/No super graph: comparison between the ML and Greedy algorithms when there is a super graph as opposed to when there is no super graph.

6.2 Twitter Graph

We present the results of two experiments on Twitter graph. In the first experiment, a connected 1000 node subgraph of the Twitter follower-following graph was extracted. Even on this small subgraph, some nodes have high in-degree and/or out-degree. To emulate people who "officially" follow many other people but are really influenced by a few of them, for each node, only a few of the in-edges of that node were assigned positive edge parameters (corresponding to those who actually influence the current node) and the rest are assigned an edge parameter of zero. Infections were then sampled by simulating the independent cascade model using the above assignment of edge parameters. Both the ML and Greedy algorithms were given infections and the entire 1000 node graph as a super graph. Figure 5 shows the plot of fraction of nodes recovered versus the average number of infections of a node.

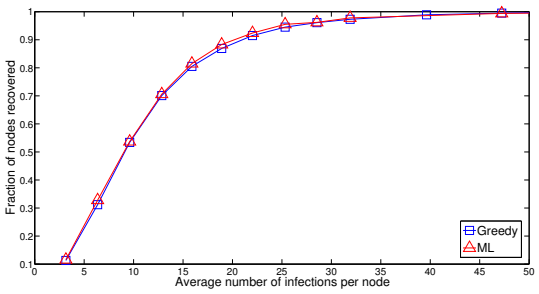

Figure 5: Twitter sub-graph with a few important parents.

In the second experiment, a 300 node subgraph of the Twitter graph was extracted. As was the case in the 1000 node sub graph, even in the 300 node subgraph, there were nodes of high in-degree and/or out-degree. For each node, all parents were assigned equal edge parameters. Infections were sampled according to the independent cascade model using these edge parameters. Both the ML and Greedy algorithms were given these infections as input without any super graph information. Figure 6 shows a scatter plot of the number of infections taken by a node for its neighborhood to be estimated correctly versus the degree of that node. Each scatter point corresponds to a node in the graph - the value of the x-axis is the degree of that node and the value of the y-axis is the number of infections of that node so that the algorithm estimated its neighborhood correctly. Note that 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.

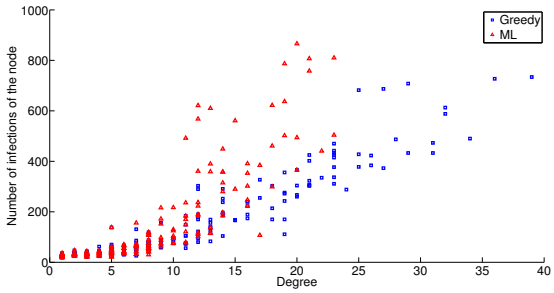

Figure 6: Twitter sub-graph with high degree: Complexity of learning the neighborhood vs degree of the node

7. ACKNOWLEDGMENTS

This research was supported by NSF grants 0954059 (CAREER) and 1017525.

8. REFERENCES

[1] 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.

[2] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[3] 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.

[4] 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.

[5] 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.

[6] J. O. Kephart and S. R. White. Directed-graph epidemiological models of computer viruses. *Security and Privacy, IEEE Symposium on*, 0:343, 1991.

[7] D. Kosterev, C. Taylor, and W. Mittelstadt. Model validation for the august 10, 1996 wscc system outage. *Power Systems, IEEE Transactions on*, 14(3):967 –979, aug 1999.

[8] 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.

[9] L. Massoulié and A. Twigg. Rate-optimal schemes for peer-to-peer live streaming. *Performance Evaluation*, 65(11-12):804 – 822, 2008.

[10] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *Proc. Neural Information Processing Systems (NIPS)*, 2010.

[11] P. Netrapalli and S. Sanghavi. Finding the graph of epidemic cascades, 2012. http://arxiv.org/abs/1202.1779.

[12] C. Perrow. *Normal Accidents: Living with High-Risk Technologies*. Princeton University Press, updated edition, Sept. 1999.

[13] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 561–568, New York, NY, USA, June 2011. ACM.

[14] 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.

[15] M. L. Sachtjen, B. A. Carreras, and V. E. Lynch. Disturbances in a power transmission system. *Phys. Rev. E*, 61:4877–4882, May 2000.

[16] 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.

APPENDIX

A. CORRELATION DECAY

PROOF OF LEMMA 1. We establish this by induction on n. If $n = 1$, the lemma is clearly true. Suppose that the lemma 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:

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

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

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

Bounds on $\mathbb{P}[T_i < \infty]$ follow from summing this 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 $X(\tau)$, $\tau = 0, \cdots, n$ 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,

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

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

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

because each node gets infected independently from each of

its currently active neighbors. Thus we have that

$$\mathbb{P}\left[T=t\right]=p_{\text{init}}^{s}\left(1-p_{\text{init}}\right)^{n-s}\prod_{i\in V}\left(\prod_{\tau=1}^{n}a_i(\tau)\right) \tag{4}$$

where $a_i(\tau)=\mathbb{P}_\theta\left[X_i(\tau)=x_i^0(\tau)\middle|X(\tau-1)=x^0(\tau-1)\right]$. 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,

$$a_i(t_i)=1-\prod_{j:t_j=t_i-1}\exp\left(-\theta_{ji}\right) \tag{5}$$

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

$$\prod_{\tau<t_i}a_i(\tau)=\prod_{j:t_j\leq t_i-2}\exp\left(-\theta_{ji}\right) \tag{6}$$

Putting (4), (5) and (6) 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

For brevity, we denote θ_{*i} by θ, $\mathcal{V}_i$ by $\mathcal{V}$ and $\mathcal{S}_i$ by $\mathcal{S}$. Let θ^* be the true parameter values. Define

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

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

Idea : Note that as m increases, $\widehat{L}\to L$. Also, we know that $\theta^*=\arg\min_\theta L(\theta)$; 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. Thus when $\widehat{L}\simeq L$, θ^* will approximately minimize $\widehat{L}$ as well. This means it will be close to $\widehat{\theta}$.

Implementing the above roadmap involves showing concentration results and some tricks from convex analysis. Since our analysis is based on first-order methods (i.e., using the gradient), we prove a characterization of the gradient of the log likelihood function. For any j, let $\bigtriangledown_j\widehat{L}(\theta)$ be the partial derivative of $\widehat{L}(\theta)$ with respect to θ_j.

Proposition 2.

$$\bigtriangledown_j L(\theta^*)=-\mathbb{P}\left[T_i>T_j;T_k\neq T_j\forall k\in\mathcal{V}\right] \tag{7}$$

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

$$\bigtriangledown_j\,L(\theta)=\mathbb{E}\left[-\mathbb{1}_{\{T_j\leq T_i-2\}}+\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k\right)-1}\right]$$

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

$$\begin{aligned}&\bigtriangledown_j\,L(\theta^*)\\&=-\mathbb{E}\left[\mathbb{E}\left[\mathbb{1}_{\{T_j\leq T_i-2\}}-\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\right]\end{aligned} \tag{8}$$

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

$$\begin{aligned}&\mathbb{E}\left[\mathbb{1}_{\{T_j\leq T_i-2\}}-\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\\&=0\text{ if }T_i\leq T_j\end{aligned} \tag{9}$$

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

$$\begin{aligned}&\mathbb{E}\left[\mathbb{1}_{\{T_j\leq T_i-2\}}-\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\\&=\mathbb{P}\left[T_i\geq T_j+2\,\middle|\,\mathcal{F}_{T_j}\right]-\\&\quad\mathbb{E}\left[\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\end{aligned}$$

Considering the two terms above separately, we see that

$$\mathbb{P}\left[T_i\geq T_j+2\,\middle|\,\mathcal{F}_{T_j}\right]=\exp\left(-\sum_{k\colon T_k=T_j}\theta_k^*\right)$$

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

$$\begin{aligned}&\mathbb{E}\left[\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\\&=\mathbb{E}\left[\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_j}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\\&\overset{(\varsigma_1)}{=}\frac{1}{\exp\left(\sum_{k\colon T_k=T_j}\theta_k^*\right)-1}\mathbb{E}\left[\mathbb{1}_{\{T_j=T_i-1\}}\,\middle|\,\mathcal{F}_{T_j}\right]\\&\overset{(\varsigma_2)}{=}\frac{\left(1-\exp\left(-\sum_{k\colon T_k=T_j}\theta_k^*\right)\right)\mathbb{1}_{\{\exists k\in\mathcal{V}\text{ s.t. }T_k=T_j\}}}{\exp\left(\sum_{k\colon T_k=T_j}\theta_k^*\right)-1}\\&=\exp\left(-\sum_{k\colon T_k=T_j}\theta_k^*\right)\mathbb{1}_{\{\exists k\in\mathcal{V}\text{ s.t. }T_k=T_j\}}\end{aligned}$$

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

$$\begin{aligned}&\mathbb{E}\left[\mathbb{1}_{\{T_j\leq T_i-2\}}-\frac{\mathbb{1}_{\{T_j=T_i-1\}}}{\exp\left(\sum_{k\colon T_k=T_i-1}\theta_k^*\right)-1}\,\middle|\,\mathcal{F}_{T_j}\right]\\&=\mathbb{1}_{\{T_k\neq T_j\ \forall\ k\in\mathcal{V}\}}\text{ if }T_i>T_j\end{aligned} \tag{10}$$

Combining (8), (9) and (10)

$$\bigtriangledown_j L(\theta^*) = -\mathbb{P}\left[T_i > T_j; T_k \neq T_k \forall k \in \mathcal{V}\right] \tag{11}$$

□

An easy corollary of Proposition 2 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.

$$\bigtriangledown_j L(\theta^*) = 0 \text{ if } j \in \mathcal{V} \tag{12}$$

$$\bigtriangledown_j L(\theta^*) < 0 \text{ if } j \notin \mathcal{V} \tag{13}$$

We now state our concentration results. For $j \in \mathcal{V}$, let

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

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

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

be the number of epidemics 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) $\left|\bigtriangledown_j \widehat{L}(\theta^*)\right| < a$ *for* $j \in \mathcal{V}$ *where* $a \colon = \frac{\alpha^3 \eta p_{init}}{144 d}$

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

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

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

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 $C = \frac{c d_i^2}{\alpha^7 \eta^2 p_{i,min}^2}$ 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.

$$\mathbb{P}\left[\sum_l X_l < (1-\kappa)\mu\right] < \left(\frac{\exp(-\kappa)}{(1-\kappa)^{(1-\kappa)}}\right)^{\mu} \tag{14}$$

$$\mathbb{P}\left[\sum_l X_l > (1+\kappa)\mu\right] < \left(\frac{e^{\frac{\kappa}{1+\kappa}}}{1+\kappa}\right)^{(1+\kappa)\mu} \tag{15}$$

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

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

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

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

where we use Lemma 1. Now applying (14) to $m_{1,j}$ we obtain

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

Similarly applying (15) to $m_{1,j}$ gives us

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

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 $\mathbb{E}\left[|\mathcal{U}_j|\right] \geq p_{\text{init}} m = C \log\frac{D}{\delta}$, using (14), we obtain

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

Similarly since $\mathbb{E}\left[|\mathcal{U}_j|\right] \leq \frac{p_{\text{init}}}{\alpha} m = \frac{C}{\alpha}\log\frac{D}{\delta}$, using (15), we obtain

$$\mathbb{P}\left[|\mathcal{U}_j| > \frac{2C\log\frac{D}{\delta}}{\alpha}\right] < \frac{\delta}{16D} \tag{17}$$

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

$$|Z_j| < 1 + \frac{1}{\exp(\theta_j^*) - 1} = \frac{1}{p_j^*} \tag{18}$$

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

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

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

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

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:

$$\begin{aligned}
&\mathbb{P}\left[\left|\sum_{u\in\mathcal{U}} Z_j^u\right| \geq ma\right] \\
&\leq \mathbb{P}\left[|\mathcal{U}_j| > \frac{2C\log\frac{D}{\delta}}{\alpha} \text{ or } |\mathcal{U}_j| < \frac{C\log\frac{D}{\delta}}{2}\right] \\
&\quad + \sum_{s=\frac{C\log\frac{D}{\delta}}{2}}^{\frac{2C\log\frac{D}{\delta}}{\alpha}} \mathbb{P}\left[|\mathcal{U}_j| = s; \left|\sum_{u\in\mathcal{U}} Z_j^u\right| \geq ma\right] \\
&\overset{(\varsigma_1)}{\leq} \frac{\delta}{8D} + \\
&\sum_{s=\frac{C\log\frac{D}{\delta}}{2}}^{\frac{2C\log\frac{D}{\delta}}{\alpha}} \sum_{U_j:|U_j|=s} \mathbb{P}\left[\mathcal{U}_j = U_j\right] \mathbb{P}\left[\left|\sum_{u\in U_j} Z_j^u\right| \geq ma \,\middle|\, \mathcal{U}_j = U_j\right]
\end{aligned} \tag{19}$$

where U_j varies over all the subsets of $\mathcal{U}$ and (ς_1) follows from (16) and (17). 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 (12), we can apply Azuma-Hoeffding inequality and using (18) we obtain

$$\begin{aligned}
&\mathbb{P}\left[\left|\sum_{u\in U_j} Z_j^u\right| \geq ma \,\middle|\, \mathcal{U}_j = U_j, |U_j| = s\right] \\
&\leq 2\exp\left(\frac{-(ma)^2}{2s\left(\frac{1}{p_j^*}\right)^2}\right) < \frac{\delta}{16D}
\end{aligned} \tag{20}$$

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 $j \notin \mathcal{V}$,

$$\begin{aligned}
\mathbb{E}[Z_j] = \bigtriangledown_j L(\theta^*) &\overset{(\varsigma_1)}{=} -\mathbb{P}\left[T_i > T_j; T_j \neq T_k \;\forall\, k \in \mathcal{V}\right] \\
&\overset{(\varsigma_2)}{<} -p_{\text{init}}\left(1-p_{\text{init}}\right)^{d+1} \overset{(\varsigma_3)}{<} -\frac{p_{\text{init}}}{2}
\end{aligned} \tag{21}$$

where (ς_1) follows from Proposition 2, (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 (21) and Lemma 1, we obtain

$$\mathbb{E}[Z_j \mid T_j < \infty] = \frac{\mathbb{E}[Z_j] - \mathbb{E}\left[Z_j 1_{\{T_j=\infty\}}\right]}{\mathbb{P}[T_j < \infty]} \leq \frac{-\alpha}{2} \tag{22}$$

Using (19) it suffices to show that

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

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.

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

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 (22) and (ς_3) is an application of the Azuma-Hoeffding inequality using (18) and the fact that $s \leq \frac{2C\log\frac{D}{\delta}}{\alpha}$. □

We will use the quantities defined in Lemma 3 in what follows. Recall we need to show $\widehat{\theta}_j < \eta$ for $j \notin \mathcal{V}$ and $\widehat{\theta}_j > \eta$ for $j \in \mathcal{V}$ and $\theta_j^* > -\log\left(1 - \frac{8}{\alpha}(e^{2\eta}-1)\right)$.

LEMMA 4. $\max_{j\in\mathcal{V}} \widehat{\theta}_j < \frac{\overline{\xi}_1}{\xi_2}$

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

$$\bigtriangledown_k \widehat{L}(\widehat{\theta}) = 0 \tag{23}$$

On the other hand, we have

$$\begin{aligned}
&\bigtriangledown_k \widehat{L}(\widehat{\theta}) \\
&= \frac{1}{m}\left(-m_{2,k} + \sum_u 1_{\{t_i^u<\infty\}}\left(\exp\left(\sum_{j:t_j^u=t_i^u-1}\widehat{\theta}_j\right) - 1\right)^{-1}\right) \\
&\overset{(\varsigma_1)}{\leq} \frac{1}{m}\left(-m_{2,k} + \frac{1}{\exp(\widehat{\theta}_k)-1} m_{1,k}\right) \\
&\overset{(\varsigma_2)}{\leq} \frac{1}{m}\left(-\xi_2 + \frac{1}{\exp(\widehat{\theta}_k)-1}\overline{\xi}_1\right) \leq \frac{1}{m}\left(-\xi_2 + \frac{1}{\widehat{\theta}_k}\overline{\xi}_1\right)
\end{aligned} \tag{24}$$

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 (23) and (24) together, we obtain the result. □

LEMMA 5. $\sum_{j\notin\mathcal{V}} \widehat{\theta}_j \leq \frac{ad}{b}\left(\frac{\overline{\xi}_1}{\xi_2} + \log\frac{1}{\alpha}\right)$

PROOF. Since $\widehat{L}(\theta)$ is concave, the subgradient condition

at θ^* gives us the following

$$\begin{aligned}\widehat{L}(\widehat{\theta}) - \widehat{L}(\theta^*) &\leq \left\langle \bigtriangledown \widehat{L}(\theta^*), \widehat{\theta} - \theta^* \right\rangle \\ &\overset{(\varsigma_1)}{=} \left\langle \bigtriangledown_{\mathcal{V}^c} \widehat{L}(\theta^*), \widehat{\theta}_{\mathcal{V}^c} \right\rangle + \left\langle \bigtriangledown_{\mathcal{V}} \widehat{L}(\theta^*), \widehat{\theta}_{\mathcal{V}} - \theta^*_{\mathcal{V}} \right\rangle \\ &\overset{(\varsigma_2)}{\leq} -b||\widehat{\theta}_{\mathcal{V}^c}||_1 + a||\widehat{\theta}_{\mathcal{V}} - \theta^*_{\mathcal{V}}||_1 \\ &\leq -b||\widehat{\theta}_{\mathcal{V}^c}||_1 + ad\left(||\widehat{\theta}_{\mathcal{V}}||_\infty + ||\theta^*_{\mathcal{V}}||_\infty\right) \quad (25)\end{aligned}$$

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

$$\widehat{L}(\widehat{\theta}) - \widehat{L}(\theta^*) \geq 0 \quad (26)$$

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

$$\theta^*_j = -\log\left(1 - p^*_j\right) \leq \log\frac{1}{\alpha} \quad (27)$$

Using (25), (26), (27) and Lemma 4 gives us the result. □

Note that $\eta > \frac{ad}{b}\left(\frac{\bar{\xi}_1}{\xi_2} + \log\frac{1}{\alpha}\right)$ and hence we have that $\widehat{\theta}_j < \eta$ for all $j \notin \mathcal{V}$. Turning now to $j \in \mathcal{V}$, we have the following:

LEMMA 6. *$\widehat{\theta}_j > \log\left(1 + \frac{p^*_j \xi_1}{\bar{\xi}_2}\right) - \eta$ where $p^*_j = 1 - \exp(\theta^*_j)$ for every $j \notin \mathcal{V}$*

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

$$\bigtriangledown_j \widehat{L}(\widehat{\theta}) \leq 0 \quad (28)$$

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

$$\begin{aligned}&\bigtriangledown_j \widehat{L}(\widehat{\theta}) \\ &= \frac{1}{m}\left(-m_{2,j} - \sum_u 1_{\{t^u_i < \infty\}}\left(\exp\left(\sum_{k: t^u_k = t^u_i - 1} \widehat{\theta}_k\right) - 1\right)^{-1}\right) \\ &\overset{(\varsigma_1)}{\geq} \frac{1}{m}\left(-m_{2,j} + \frac{1}{\exp\left(\widehat{\theta}_j + ||\widehat{\theta}_{\mathcal{V}^c}||_1\right) - 1} m_{1,k}\right) \\ &\overset{(\varsigma_2)}{\geq} \frac{1}{m}\left(-\bar{\xi}_2 + \frac{1}{\exp\left(\widehat{\theta}_j + ||\widehat{\theta}_{\mathcal{V}^c}||_1\right) - 1} p^*_j \xi_1\right) \quad (29)\end{aligned}$$

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

$$\sum_{k: t^u_k = t^u_i - 1} \widehat{\theta}_k \leq \widehat{\theta}_j + ||\widehat{\theta}_{\mathcal{V}^c}||_1$$

and (ς_2) follows from Lemma 3. Combining (28), (29) and Lemma 5 gives us the result. □

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

C. LOWER BOUNDS

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

$$\begin{aligned}H(T_i) &= -\sum_{t=1}^{n} \mathbb{P}[T_i = t] \log \mathbb{P}[T_i = t] \\ &\quad -\mathbb{P}[T_i = \infty] \log \mathbb{P}[T_i = \infty] \\ &\leq -\sum_{t=1}^{n} (1-\alpha)^{t-1} p_{\text{init}} \log (1-\alpha)^{t-1} p_{\text{init}} \\ &\quad -\left(1 - \frac{p_{\text{init}}}{\alpha}\right) \log\left(1 - \frac{p_{\text{init}}}{\alpha}\right) \\ &\overset{(\varsigma_1)}{\leq} \frac{p_{\text{init}}}{1-\alpha}\left(\log\frac{1}{p_{\text{init}}} + \left(\frac{1-\alpha}{\alpha}\right)^2 \log\frac{1}{1-\alpha}\right) \\ &\quad -\left(1 - \frac{p_{\text{init}}}{\alpha}\right) \log\left(1 - \frac{p_{\text{init}}}{\alpha}\right)\end{aligned}$$

where (ς_1) follows from some algebraic manipulations. □