Learning Graphs from Noisy Epidemic Cascades

Jessica Hoffmann

The University of Texas at Austin
Austin, TX
hoffmann@cs.utexas.edu

Constantine Caramanis

The University of Texas at Austin
Austin, TX
constantine@utexas.edu

CCS CONCEPTS

• **Mathematics of computing** → **Graph theory; Graph algorithms; Probability and statistics; Computing most probable explanation; Combinatorics**; • **Networks** → **Network algorithms; Theory of computation** → **Graph algorithms analysis; Sample complexity and generalization bounds; Social networks.**

KEYWORDS

Epidemics; SIR model; Contact process on graph; Noisy inverse problem; Graph reconstruction; Robust graph algorithms.

ACM Reference Format:

Jessica Hoffmann and Constantine Caramanis. 2019. Learning Graphs from Noisy Epidemic Cascades. In *ACM SIGMETRICS Int'l Conference on Measurement & Modeling of Computer Systems (SIGMETRICS'19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3326155>

1 INTRODUCTION

Epidemic models accurately represent (among other processes) the spread of diseases, information (rumors, viral videos, news stories, etc.), the spread of malevolent agents in a network (computer viruses, malicious apps, etc.), or even biological processes (pathways in cell signaling networks, chains of activation in the gene regulatory network, etc.). We focus on epidemics that spread on an underlying graph [5].

Most settings assume we know the underlying graph and aim to study properties of the spread. Much work has been done in detection, where the goal is to decide whether or not there is indeed an infection. This problem is of importance in deciding whether or not a computer network is under attack, for instance, or whether a product gets sold through word-of-mouth or thanks to the advertisement campaign (or both). More specifically, the problem of source detection or obfuscation has been extensively studied. On the other side of the spectrum, both experimental and theoretical work has tackled the problem of modeling, predicting the growth, and controlling the spread of epidemics.

In this work, we tackle the complementary problem: assuming we know some properties of the spread, can we recover the underlying graph? The early works on this subject proposed a few heuristics and experimentally proved their effectiveness. Netrapalli et al. [4] established the first theoretical guarantees for this problem

for discrete-time infections. They proved one can recover the edges of any graph with correlation decay, with access to the times of infection for multiple cascades spreading on the graph. They introduced a likelihood, proved it decouples into convex subproblems, and demonstrated that the edges of the graph can therefore be obtained efficiently. They also proved a sample complexity lower bound and showed their method is within a log factor of it. Abraham et al. [1] also introduced a method of solving this problem, this time for a more realistic, continuous-time infection model, through learning only the first edge of each cascade. Zarezade et al. [6] proposed a first experimental attempt to tackle the case of correlated cascades using Hawkes processes. Khim et al. [3] extended the theoretical results to the case where the cascades spreading on the graph are not independent, which required completely new machinery involving martingales and weighted Pólya urns.

All the results above assume we have perfect knowledge of the properties of the spread we use to reconstruct the graph. For most of the literature, those are the times of infection for all nodes for each cascade. This assumption may hold for online epidemics, as information is usually dated (for instance, posts or retweets on social networks have time stamps). For human networks, or any setting where diagnosis is required, this assumption is often unrealistic: official diagnosis (and hence recording by any tracking entity such as the CDC) may come days, weeks, or in important examples such as HIV, *years* after the actual moment of infection. Moreover, this can be highly variable from person to person, hence the infector is often diagnosed after the infectee. Similar issues arise with biological networks: we only know the expression of a gene when a measure is taken, which can happen after a typically arbitrary delay.

We therefore develop a method for learning the graph of epidemics with noisy times of infection. We demonstrate that past approaches are unusable, due to the fact that even small levels of noise are typically enough to cause order-of-diagnosis to differ from order-of-infection. We develop new techniques to deal with this setting, and prove our algorithm can reliably learn the edges of a tree in the limited-noise setting, for *any* noise distribution. We also show we can learn the structure of any bounded degree graph from a very weak observation model, in a sample-optimal fashion. Finally, we provide an algorithm which learns the weights of any bounded-degree graph in the limited-noise setting.

1.1 Model

Propagation model: We consider a particular variant of the *independent cascade model* [2]. The epidemic spreads in *discrete time* on a *weighted directed graph* $G = (V, E)$, where parents can infect their children with probability equal to the weight of the edge between them, but children cannot infect their parents.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS'19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3326155>

This process is an instance of a *Susceptible* $\rightarrow$ *Infected* $\rightarrow$ *Removed* (SIR) process. Each node starts out in the Susceptible state. As in [1], each cascade m starts on a unique Infected source node, picked uniformly among the nodes of the graph. Once a node becomes infected, it is removed from the graph, and if it has susceptible children, each is infected at the next time step independently with probability equal to the weight of the edge from infected parent node to its susceptible child. The process ends when there are no newly infected nodes (either because no infection happened during the previous time step, in which case some nodes may never be infected, or because all the nodes of the graph are removed). One realization of this process from start to finish is called a **cascade**.

Observation model: We introduce three observation models. In the **no-noise setting**, we have access to the exact times of infection. In the **limited-noise setting**, we never get to observe the exact times of infection, but only a noisy version. We assume the noise is i.i.d., comes from a known distribution, and cannot have infinite value.

Learning tasks: We focus on two different learning tasks. When we learn the **structure** of a graph, it means that for any two nodes i and j , we can decide whether or not there exists an edge between these two nodes (whatever its direction). When we learn the **weights** of the graph, it means that for every two nodes i and j , we learn the exact value of both p_{ij} and p_{ji} up to any given precision ϵ .

1.2 Why is it a hard problem?

1.2.1 Counting approaches. Most approaches in the no-noise setting relate to counting. In our setting, for instance, a natural (and consistent) estimator for p_{ij} is:

$$\hat{p}_{ij} = \frac{\text{Number of times } j \text{ becomes infected one time step after } i}{\text{Number of times } i \text{ was infected before } j}.$$

In the no-noise setting, j could only have been infected by a node signaling exactly one time step before j . However, in the limited-noise setting, j signaling its infection one time step after i could stem from a variety of scenarios:

- i could have indeed infected j .
- j could have infected i , but the noise flipped the order of signaling.
- No infection happen between i and j , and the probability of infection depends mainly on other nodes of the graph. For instance, i could have infected k , which in turn infected j .

The natural estimator introduced earlier is therefore not consistent anymore; instead, it tends to a quantity which depends on p_{ij} , but also p_{ji} , and $p_{ik}, p_{ki}, p_{jk}, p_{kj}$ as well, for all the other nodes k in the graph. Using this estimator, or any obvious (to us) extension of it, would not only imply learning the wrong weights for the edges, but also learning edges when there are no edges.

1.2.2 Max-likelihood approaches. Another common approach is to use likelihood-based methods. In our setting, the log-likelihood is not convex, and depends on many latent variables. This implies that simply computing the log-likelihood take exponential time, which render max-likelihood unusable in practice.

1.3 Proofs intuition

For our results on bidirectional trees, we rely on the fact that there is unicity of paths between two nodes. This allows us to compute a quantity for each pair of nodes, which is locally maximal when the nodes share an edge. This allows us to learn the structure of the tree, which can in turn be leveraged to learn the weights of the tree.

For graphs of maximal degree d , we extend the previous quantity to all pairs of (node, sets of nodes of size at most d). Similarly, this allows us to learn the structure of the graphs. We finally use cascades of size up to two to learn all the weights.

1.4 Contributions

The contributions of this article are multiple:

- This is the first work (as far as we know) to tackle the problem of learning the edges of a graph from noisy times of infection, a simple but natural extension of a well-known problem.
- We provide the first efficient algorithm for learning the structure and weights of a bidirectional tree in this setting. We also establish a tree-specific lower bound which shows that our algorithm is sample-optimal for learning the structure of the tree.
- We prove it is possible to learn the structure of any bounded-degree graph in the extreme setting for which we only have access to the infection status (*i.e.*, whether or not a node was infected). Moreover, we can do so with optimal sample complexity, according to the bound established in [4].
- We provide polynomial algorithms for learning the weights of bounded degree graphs.
- Finally, we extend the results from bounded-degree graphs to general graphs. This proves the problem is solvable under any noise distribution, though exponential sample complexity and running time are prohibitive in practice.

REFERENCES

- [1] Bruno Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. 2013. Trace complexity of network inference. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13* (2013), 491. <https://doi.org/10.1145/2487575.2487664> arXiv:arXiv:1308.2954v1
- [2] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval* 4, 2 (2001), 133–151. <https://doi.org/10.1023/A:1011419012209> arXiv:arXiv:astro-ph/0005074v1
- [3] Justin Khim and Po-Ling Loh. 2018. A theory of maximum likelihood for weighted infection graphs. (2018), 1–47. arXiv:arXiv:1806.05273v1 <https://arxiv.org/pdf/1806.05273.pdf>
- [4] Praneeth Netrapalli and Sujay Sanghavi. 2012. Learning the Graph of Epidemic Cascades. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems (SIGMETRICS'12)*. 211–222. <https://doi.org/10.1145/2318857.2254783> arXiv:1202.1779
- [5] M. E. J. Newman. 2014. *Networks: An Introduction*. Vol. 23. 73–75 pages. <https://doi.org/10.1017/S0963180113000479>
- [6] Ali Zareade, Ali Khodadadi, Mehrdad Farajtabar, Hamid R Rabiee, and Hongyuan Zha. 2017. Correlated Cascades : Compete or Cooperate. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. 238–244.

ACKNOWLEDGMENTS

The work is partially supported by the National Science Foundation under Grants No.: 1302435, No.: 1609279 and No.: 1704778.

Received February 2019; revised March 2019; accepted April 2019