A Comparative Analysis of Source Identification Algorithms

Pablo A. Curiel, Richard C. Tillquist

California State University, Chico, USA

Abstract

Identifying the source of a spread in a network, often referred to as the patient-zero problem, is a difficult task when only given the subgraph of infected nodes. Since 2011, several algorithms have been created to try to solve this problem. Their success depends on many factors, such as the graph's size and structure, the infection rate, and time since the start of the spread. This paper empirically compares four prominent source identification algorithms when applied to randomly-generated graphs and to a real-world airport network.

1. Introduction

The importance of modeling spread processes on networks can not be overstated. Applications related to the spread of diseases in a physical-contact network, rumors, knowledge, or memes in a social network, electricity in a power grid network, and malware attacks in a computer-systems network [1] all require a deep understanding of the relationship between network structure and spread dynamics.

In recent years, determining the source of a spread (often referred to as **patient-zero**) has become a popular area of research in network science. This is largely due to how networks capture important features necessary for spread, in contrast to some traditional models of spread. Many traditional models rely on the assumption of homogeneous mixing in the population, which states that all individuals have the same probability of coming into contact with any other individual. This assumption implies that all individuals are connected and disregards the true contact network. Determining the source of a spread can have many critical real-world applications, such as contact tracing, optimizing the allocation of immunization resources, mitigating the spread of rumors or misinformation, and defending against malware attacks.

This paper focuses on comparing four state-of-the-art methods of identifying the source of a spread when applied to random graphs of varying type and size, as well as a real-world U.S. airport network [2].

2. Background

Traditional spread models often utilize continuous-time differential equations to simulate a spread. Perhaps the most popular of these models is the SI-model, where individuals in a population are divided into two distinct classes or states [3]: **S**usceptible or **I**nfected. There are many variations of the SI-model, which include different numbers or orderings of states. Some of these variations include the SIR-model that adds a **R**ecovered (or removed) state, the SIS-model that allows an infected individual to return to a **S**usceptible state, and the SEIR-model that includes a pre-infectious **E**xposed state.

These traditional models often rely on two fundamental assumptions: compartmentalization and homogeneous mixing [1]. Compartmentalization refers to the classifying of individuals in a population based on their disease state. Homogeneous mixing, also known as well-mixed, is unrealistic in most real-world scenarios. Such models help capture the broader picture of a spread, but their assumption of a well-mixed population reduces their realism. This is where modeling spread using networks becomes important.

In contrast to traditional models, modeling spread on networks takes into account each individual's contacts. This helps provide a more realistic representation of a spread, as individuals can only interact with their direct connections. In addition, spread models on networks use discrete time-steps instead of continuous time. For the SI-model on a network, a single source node is chosen at time $t = 0$ and classified as infected. At each following time-step, an infected node can infect any of its susceptible one-hop neighbors with probability β, the infection rate. Given a connected network G_N and $\beta > 0$, $I_N \to N$ as $t \to \infty$, where I_N is the number of infected nodes in the network and N is the total number of nodes in the network. An important quantity when discussing spread in networks is the transmission rate $\beta\langle k\rangle$, where $\langle k\rangle$ is the average degree of the network. This can be thought of as the speed of the spread through a network and is analogous to the basic reproductive rate R_0 in the context of epidemic spreads. More precisely, $R_0 = \frac{\beta\langle k\rangle}{\mu}$, where μ represents a recovery rate. When $R_0 > 1$ we can expect an epidemic will spread,

and when $R_0 < 1$ we can expect an epidemic will die out [1, 3].

3. Related Work

In 2011, Shah and Zaman published the first paper focused on identifying the source of a spread in a network [4]. Their primary contribution was a maximum likelihood estimator for the source node called "rumor centrality". They define rumor centrality as the number of permitted permutations of an infected subgraph centered at some node. A permitted permutation describes all of the possible orderings of nodes that led to a given infected subgraph starting with some node. Rumor centrality is calculated for all nodes in the infected subgraph, and the source, or rumor center, is estimated to be the node with the maximum rumor centrality. Their method is designed for tree-structured graphs. For general graphs, the method finds an infected subgraph using a BFS-tree rooted at each node, then calculates the rumor centrality for each BFS-tree.

Jordan centrality, another measure of node importance with respect to a spread, refers to a node's eccentricity, or maximum distance to all other nodes in the network [5, 6]. A "Jordan infection center" describes an infected node with the smallest Jordan centrality out of all other nodes in an infected subgraph. This node is estimated to be the source of the spread.

The NETSLEUTH algorithm [7] focuses on the minimum description length principle. When applied to source inference, it is referred to as the minimal infection description problem. The goal of this algorithm is to minimize the total description length of an infected subgraph for a given set of seed nodes, or possible sources, and a valid propagation ripple, which can be thought of as a permitted permutation from the Shah and Zaman paper [4]. The NETSLEUTH algorithm iteratively reduces the description length of an infected subgraph by considering potential sources given the current set of infected nodes until the most succinct encoding is found. Notably, this algorithm is linear in the size of the graph.

Nie and Quinn [8] provide an algorithm that calculates a maximum likelihood estimator of each node in the infected subgraph being the source. Their algorithm assumes that the time it takes for the spread to propagate across edges in the graph is exponentially distributed. In their method, the probability of observing the set of infected nodes is calculated using each node as the source. The node maximizing this probability is chosen as the estimated source. The algorithm does not have a name and is referred to as "LISN" following the title of the paper in which it was introduced [8].

4. Experiments

4.1. Experimental Setup

Using the cosasi Python package [9], we tested the four source inference methods described in the previous section under different conditions. This package allows users to simulate a spread on a NetworkX [10] graph, apply several source inference methods, and obtain results related to the spread and inference methods. For all simulations, a single source node was chosen at random, and the spread was simulated for up to 100 time-steps. Each source inference method was applied at time-steps $10, 20$, and 30.

Experiments were conducted on four random network types and one real-world network. Barabási-Albert random graphs [11] with $m = 1$ and 3, Erdős-Rényi random graphs [12] with $p = \frac{\ln(N)+1}{N}$, and Watts-Strogatz random graphs [13] with parameters $p = 0.01$ and $k = 4$ were considered for networks of size $100, 250$, and 500.

Spread simulations were conducted using five different infection rates: $0.1, 0.2, 0.3, 0.4$, and 0.5. A total of $6,000$ spread simulations were conducted on the random networks with $1,500$ for each random network type. For each set of parameters (network type, network size, β-value), 100 spread simulations were run.

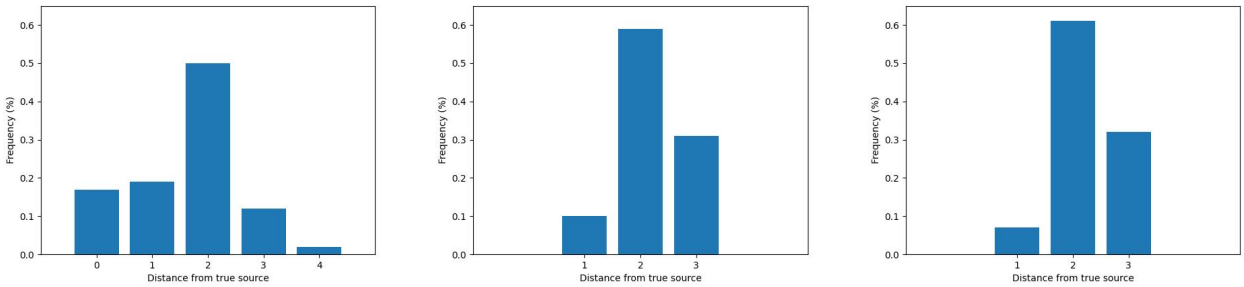
Spreads across a real-world network of U.S. airports [2] were also simulated with the same set of infection rates and observation times. Due to the size of this network ($1,572$ nodes, $17,214$ edges) and to computational limitations, twenty spreads were simulated for each infection rate.

All code is available on GitHub: `https://github.com/pacuriel/cscsu-2023`

4.2. Experimental Results

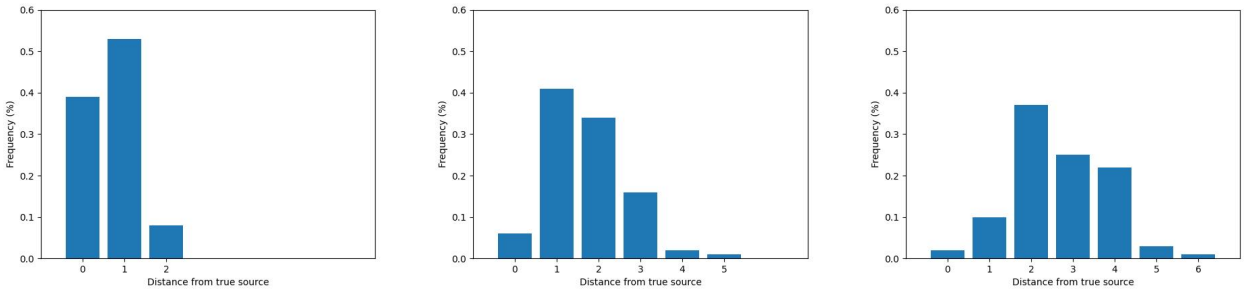
From the experiments conducted, several metrics were collected to analyze the success of each source inference method. These metrics include the estimated source's distance from the true source (in number of hops), the average distance from the estimated source to the true source, the true source's rank, and the average of the true source's rank. Rank is a measure of a node's likelihood to be the source node. For example, an inference method's estimated source will have a rank of one. Ideally, the higher a node's rank is, the more likely it is to be the source node.

Figures 1 and 2 display results after inferring the source using the LISN algorithm [8] and rumor centrality [4]. While the primary purpose of the figures is to emphasize the roles of the infection rate and observation time, these methods were chosen because of how they differ in terms of time. Specifically, rumor centrality is the oldest method (2011) and LISN is the newest method (2019) out of the four methods described in this paper.



(a) (b) (c)

Figure 1: LISN algorithm; BA ($m = 3$), $N = 500, \beta = 0.1$; (a): $t = 10$, (b): $t = 20$, (c): $t = 30$.



(a) (b) (c)

Figure 2: Rumor centrality; BA ($m = 1$), $N = 500, t = 10$; (a): $\beta = 0.1$, (b): $\beta = 0.3$, (c): $\beta = 0.5$.

The importance of the infection rate β and the observation times is highlighted in figures 1 and 2. Figure 1 shows the frequency of the estimated source's distance from the true source for the LISN method when applied to Barabási-Albert random graphs with $m = 3$ where $N = 500, \beta = 0.1$, and the observation time t varies. When $t = 10$, the method is able to correctly identify the source node approximately 20% of the time. As t increases to 20 and 30, the method is no longer able to correctly identify the source node, which is shown by the absence of a distance $d = 0$. Similarly, figure 2 shows the frequency of the estimated source's distance from the true source for the rumor centrality method applied to Barabási-Albert random graphs with $m = 1$ where $N = 500, t = 10$, and the infection rate β varies. Rumor centrality is known to excel when applied to tree-structured graphs, as shown by its approximately 40% accuracy of identifying the source when $\beta = 0.1$. However, once $\beta = 0.3$, its top-1 accuracy drops below 10%, and it drops below 5% when $\beta = 0.5$.

Table 1 reinforces the significance of the infection rates and observation times, while showing the importance of graph structure. The rows of these tables are source inference methods, and the columns are the ratios of average rank $\langle R \rangle$ to graph size N and average distance between the true and estimated source $\langle d \rangle$ to average diameter $\langle diam(G) \rangle$. All table values were obtained from simulations on graphs with $N = 500$ nodes and varying β and time values.

Table 1a's values result from simulations with $\beta = 0.5$ and time $t = 30$. Aside from the Watts-Strogatz random graphs, most graphs had poor average ranks for the true source node, as well as average distance between the estimated and true sources. In table 1b, the simulations were conducted with a weaker infection rate of $\beta = 0.1$. This resulted in better results for some graph types. Notably, most graph types produced results similar to table 1a. Lastly, table 1c resulted from experiments with a low infection rate $\beta = 0.1$ and time of observation $t = 10$. These low parameter values resulted noticeable differences across all graph types.

5. Discussion

The experiments conducted have shown how the success of each source inference method depends on several factors, including the graph structure, the spread's infection rate, and the time since the spread began. Some methods known to perform well with certain graph types were

Type/method	**a.)** $\beta = 0.5, t = 30$		**b.)** $\beta = 0.1, t = 30$		**c.)** $\beta = 0.1, t = 10$	
BA $(m = 1)$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$	$\frac{\langle R \rangle}{N}$	$\frac{\langle d \rangle}{\langle diam(G) \rangle}$
Rumor	0.50	0.25	0.09	0.13	0.02	0.06
Jordan	0.51	0.25	0.08	0.14	0.02	0.10
NETSLEUTH	0.48	0.29	0.10	0.16	0.02	0.13
LISN	0.49	0.25	0.07	0.16	0.02	0.10
BA $(m = 3)$						
Rumor	0.50	0.36	0.52	0.36	0.16	0.36
Jordan	0.51	0.36	0.54	0.36	0.19	0.36
NETSLEUTH	0.51	0.36	0.44	0.36	0.15	0.36
LISN	0.48	0.36	0.54	0.36	0.17	0.36
ER						
Rumor	0.55	0.48	0.50	0.37	0.06	0.32
Jordan	0.50	0.40	0.50	0.32	0.08	0.322
NETSLEUTH	0.45	0.48	0.48	0.40	0.14	0.40
LISN	0.53	0.40	0.48	0.40	0.09	0.32
WS						
Rumor	0.12	0.15	0.02	0.05	0.01	0.02
Jordan	0.04	0.12	0.02	0.05	0.01	0.02
NETSLEUTH	0.26	0.19	0.04	0.07	0.02	0.02
LISN	0.11	0.15	0.02	0.03	0.01	0.02
AP						
Rumor	0.43	0.25	0.40	0.25	0.32	0.25
Jordan	0.44	0.25	0.39	0.25	0.36	0.25
NETSLEUTH	0.47	0.25	0.47	0.25	0.47	0.25
LISN	0.46	0.25	0.39	0.25	0.28	0.13

Table 1: Ratio of average rank of true source $\langle R \rangle$ to graph size N and the ratio of average distance between the estimated and true sources $\langle d \rangle$ to average diameter $\langle diam(G) \rangle$ for varying graph types where $N = 500$. Values for β and t vary and are listed above each column.

shown to decline as spread parameters were intensified. Interestingly, the Watts-Strogatz random graphs displayed the most promising results across all inference methods and spread parameters. Their average distance between estimated and true sources were often relatively low, which is likely because their diameter and distances scale logarithmically with the size of the graphs.

Notably, table 1b displayed little-to-no change in the results for Barabási-Albert random graphs with $m = 3$, Erdős-Rényi random graphs, and the real-world U.S. airport network after lowering the infection rate. This likely has to do with the structure and overall density of these graphs. Also, the scale-free nature of Barabási-Albert random graphs with $m = 3$ and the U.S. airport network likely had a large influence on the spread. The hubs associated to these graph structures can be thought of as super-spreaders in an epidemic context. Their many connections allow them to quickly become infected and propagate the spread to the rest of their uninfected connections. This highlights the importance of taking action early when a spread occurs before it gets into contact with the hubs.

It is important to mention that each of the methods studied perform best under different conditions. For example, rumor centrality is known to perform well on tree-structured graphs when there is a single-source. In contrast, NETSLEUTH is capable of inferring multiple sources. Depending on the context, one method may be the best option for inferring the source of a spread.

6. Future Work

Future work in this area involves conducting more experiments on other source inference methods. Applying these methods to more real-world networks could also prove useful. Taking into account more metrics and extracting results in a more statistical manner would likely provide more insight on the performance of these inference methods. Determining what statistical methods are best for this analysis is a problem alone.

We focused our attention on the parameters closely related to the spread simulations, such as the infection rate and time since spread began. It would be helpful to take into account other relevant parameters or metrics, includ-

ing the graph size and each graph type's parameters.

Also, as the field of deep learning continues to grow, especially the area of graph neural networks, learning how to best apply these methods to the source identification problem could help improve overall accuracy and efficiency. Research in this area has already begun [14], and it will be interesting to see how it performs once perfected.

7. Conclusion

In this paper, we conducted extensive experiments on four state-of-the-art source inference methods. These experiments varied greatly in both graph and spread parameters. Simulations were conducted on several random graph types and a real-world U.S. airport network. The success of these inference methods was shown to depend on several factors, such as graph structure, graph size, infection rates, and time since the start of the spread.

References

[1] Barabási A, Pósfai M. Network Science. Cambridge University Press, 2016. ISBN 9781107076266. URL https://books.google.com/books?id=iLtGDQAAQBAJ.

[2] Kunegis J. Konect: The Koblenz Network Collection. In Proceedings of the 22nd International Conference on World Wide Web. 2013; 1343–1350. URL http://konect.cc/networks/opsahl-usairport/.

[3] Murray JD. Mathematical Biology: I. An Introduction. Springer, 2002.

[4] Shah D, Zaman T. Rumors in a network: Who's the culprit? IEEE Transactions on Information Theory 2011; 57(8):5163–5181.

[5] Ying L, Zhu K. Diffusion source localization in large networks. Synthesis Lectures on Communication Networks 2018;11(1):1–95.

[6] Luo W, Tay WP, Leng M. On the universality of jordan centers for estimating infection sources in tree networks. IEEE Transactions on Information Theory 2017; 63(7):4634–4657.

[7] Prakash BA, Vreeken J, Faloutsos C. Efficiently spotting the starting points of an epidemic in a large graph. Knowledge and Information Systems 2014;38(1):35–59.

[8] Nie G, Quinn C. Localizing the information source in a network. In TrueFact 2019: KDD 2019 Workshop on Truth Discovery and Fact Checking: Theory and Practice. 2019; .

[9] McCabe LH. cosasi: Graph diffusion source inference in python. Journal of Open Source Software 2022;7(80):4894.

[10] Hagberg A, Swart P, S Chult D. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab (LANL), Los Alamos, NM (United States), 2008.

[11] Barabási AL, Albert R. Emergence of scaling in random networks. science 1999;286(5439):509–512.

[12] Erdős P, Rényi A. On random graphs i. Publicationes mathematicae 1959;6(1):290–297.

[13] Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature 1998;393(6684):440–442.

[14] Shah C, Dehmamy N, Perra N, Chinazzi M, Barabási A, Vespignani A, Yu R. Finding patient zero: Learning contagion source with graph neural networks. CoRR 2020; abs/2006.11913. URL https://arxiv.org/abs/2006.11913.

Address for correspondence:

Pablo A. Curiel
Computer Science Department
400 W. First St, Chico, CA, 95929
pacuriel@csuchico.edu