Rumors in a Network: Who's the Culprit?

Devavrat Shah
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
devavrat@mit.edu

Tauhid R . Zaman
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
zlisto@mit.edu

Abstract

Motivated by applications such as the detection of sources of worms or viruses in computer networks, identification of the origin of infectious diseases, or determining the causes of cascading failures in large systems such as financial markets, we study the question of inferring the source of a *rumor* in a network.

We start by proposing a natural, effective model for the spread of the rumor in a network based on the classical SIR model. We obtain an estimator for the rumor source based on the infected nodes and the underlying network structure – it assigns each node a likelihood, which we call the *rumor centrality*. We show that the node with the maximal rumor centrality is indeed the maximum likelihood estimator for regular trees. Rumor centrality is a complex combinatoric quantity, but we provide a simple linear time message-passing algorithm for evaluating it, allowing for fast estimation of the rumor source in large networks.

For general trees, we find the following surprising phase transition: asymptotically in the size of the network, the estimator finds the rumor source with probability 0 if the tree *grows* like a line and it finds the rumor source with probability strictly greater than 0 if the tree *grows* at a rate quicker than a line.

Our notion of rumor centrality naturally extends to arbitrary graphs. With extensive simulations, we establish the effectiveness of our rumor source estimator in different network topologies, such as the popular small-world and scale-free networks.

1 Introduction

In the modern world the ubiquity of networks has made us vulnerable to new types of network risks. These network risks arise in many different contexts, but share a common structure: an isolated risk is amplified because it is spread by the network. For example, as we have witnessed in the recent financial crisis, the strong dependencies or 'network' between institutions have led to the situation where the failure of one (or few) institution(s) have led to global instabilities. Other network risks include computer viruses or worms spreading in the Internet or contagious diseases spreading in populations. Another situation where network risks arise is when social networks allow information and instructions to be disseminated. In this case finding the leader of these networks is of great interest for various purposes – identification of the 'hidden voice' in a spy network or determining the 'latent leader' in a political network.

In essence, all of these situations can be modeled as a rumor spreading through a network. The goal is to find the source of the rumor in these networks in order to control and prevent these network risks based on limited information about the network structure and the 'rumor infected' nodes. In this paper, we will provide a systematic study of the question of identifying the rumor source in a network, as well as understand the fundamental limitations on this estimation problem.

1.1 Related Work

Prior work on rumor spreading has primarily focused on viral epidemics in populations. The natural (and somewhat standard) model for viral epidemics is known as the *susceptible-infected-recovered* or SIR model [1]. In this model, there are three types of nodes: (i) susceptible nodes, capable of being infected; (ii) infected nodes that can spread the virus further; and (iii) recovered nodes that are cured and can no longer become infected. Research in the SIR model has focused on understanding how the structure of the network and rates of infection/cure lead to large epidemics [2],[3]. This motivated various researchers to propose network inference techniques to learn the relevant network parameters [4],[5],[6],[7],[8]. However, there has been little (or no) work done on inferring the source of an epidemic.

The primary reason for the lack of such work in finding epidemic sources is that the problem is quite challenging. To substantiate this, we briefly describe a closely related (and relatively simpler) problem of reconstruction on trees [9],[10], or more generally, on graphs [11]. In this problem, a known source transmits a binary value (-1 or 1) to other nodes in a network with a noisy channel. The problem then is to use the noisy value at the other nodes to infer the value of the source. Currently this problem is only understood for trees and tree-like graphs. Now the rumor source identification problem is, in a sense harder, as we wish to identify the location of the source among many nodes based on the infected nodes – clearly a much noisier situation than the reconstruction problem.

1.2 Our Contributions

In this paper, we provide a systematic study of the question of finding the rumor source in a network. We construct the rumor source estimator in Section 2. We first propose a simple rumor spreading model based upon the SIR model and then cast finding the rumor source as a maximum likelihood (ML) estimation problem. Following the approach of researchers working on the reconstruction problem and efficient inference algorithm design (i.e. Belief Propagation), we begin by addressing the rumor source estimation problem for tree networks. For regular trees, we are able to reduce the ML estimator to a novel combinatoric quantity we call *rumor centrality*. Despite being a complex combinatoric object, we are able to provide a simple linear time message-passing algorithm for evaluating rumor centrality. We note that this message-passing algorithm has no relation to standard Belief Propagation or its variants, other than that it is an iterative algorithm.

We extend this notion of rumor centrality to construct estimators for general trees. Building on the tree estimator, we design an estimator for any graph. We study the performance of this general graph estimator through extensive simulations. As representative results, we test its performance on the popular small-world and scale-free networks. This is detailed in Section 4, along with simulation results for the tree estimator.

Section 3 presents our theoretical results on the rumor source estimator performance. For arbitrary trees, we find the following surprising threshold phenomenon about the estimator's effectiveness. If a tree grows like a line, then the detection probability of the ML rumor source estimator will go to 0 as the network grows in size; but for trees growing faster than a line, the detection probability of our estimator will always be strictly greater than 0 (uniformly bounded away from 0) irrespective of the network size. In the latter case, we find that when the estimator makes an error, the wrong prediction is within a few hops of the actual source. Thus, our estimator is essentially the optimal for any tree network.

We conclude in Section 5 with implications of these results and future directions for this work.

2 Rumor Source Estimator

2.1 Rumor Spreading Model

We consider a network of nodes to be modeled by an undirected graph $G(V, E)$, where V is a countably infinite set of nodes and E is the set of edges of the form (i, j) for some i and j in V. We assume the set of nodes is countably infinite in order to avoid boundary effects. We consider the case where initially only one node v^* is the rumor source.

For the rumor spreading model, we use a variant of the popular SIR model known as the *susceptible-infected* or SI model which does not allow for any nodes to recover, i.e. once a node has the rumor, it keeps it forever. Once a node i has the rumor, it is able to spread it to another node j if and only if there is an edge between them, i.e. if $(i, j) \in E$. The time for a node i to spread the rumor to node j is modeled by an exponential random variable τ_{ij} with rate λ. We assume without loss of generality that $\lambda = 1$. All τ_{ij}'s are independent and identically distributed.

2.2 Rumor Source Maximum Likelihood Estimator

We now assume that the rumor has spread in $G(V, E)$ according to our model and that N nodes have the rumor. These nodes are represented by a rumor graph $G_N(V, E)$ which is a subgraph of $G(V, E)$. We will refer to this rumor graph as G_N from here on. The actual rumor source is denoted as v^* and our estimator will be $\widehat{v}$. We assume that each node is equally likely to be the source a priori, so the best possible estimator will be the ML estimator. The only data we have available is the final rumor graph G_N, so the estimator is

$$\widehat{v} = \arg\max_{v \in G_N} \mathbf{P}(G_N | v^* = v) \tag{1}$$

In general, $\mathbf{P}(G_N | v^* = v)$ will be difficult to evaluate. However, we will show that in regular tree graphs, ML estimation is equivalent to a combinatorial problem.

2.3 Rumor Source Estimator for Regular Trees

To simplify our rumor source estimator, we consider the case where the underlying graph is a regular tree where every node has the same degree. In this case, $\mathbf{P}(G_N | v^* = v)$ can be exactly evaluated when we observe G_N at the instant when the N^{th} node is infected.

First, because of the tree structure of the network, there is a unique sequence of nodes for the rumor to spread to each node in G_N. Therefore, to obtain the rumor graph G_N, we simply need to construct a permutation of the N nodes subject to the ordering constraints set by the structure of the rumor graph, which we refer to as a *permitted permutation*. For example, for the network in Figure 1, if node 1 is the source, then $\{1, 2, 4\}$ is a permitted permutation, whereas $\{1, 4, 2\}$ is not because node 2 must have the rumor before node 4.

Second, because of the memoryless property of the rumor spreading time between nodes and the constant degree of all nodes, each permitted permutation resulting in G_N is equally likely. To see this, imagine every node has degree k and we wish to find the probability of a permitted permutation σ conditioned on $v^* = v$. A new node can connect to any node with a free edge with equal probability. When it joins, it contributes $k - 2$ new free edges. Therefore, the probability of any N node permitted permutation σ for any node v in G_N is

$$\mathbf{P}(\sigma | v^* = v) = \left(\frac{1}{k}\right)\left(\frac{1}{k + (k-2)}\right)\left(\frac{1}{k + 2(k-2)}\right) \cdots \left(\frac{1}{k + (N-2)(k-2)}\right) \tag{2}$$

The probability of obtaining G_N given that $v^* = v$ is obtained by summing the probability of all permitted permutations which result in G_N. Because all of the permutations are equally likely, $\mathbf{P}(G_N | v^* = v)$ will be proportional to the number of permitted permutations which start with v and result in G_N. Because we will find it necessary to count the number of these permutations, we introduce the following definition:

Definition 1. *R(v,T) is the number of permitted permutations of nodes that result in a tree T and begin with node $v \in T$.*

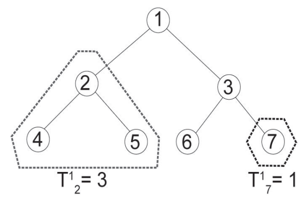

Figure 1: Illustration of variables T_2^1 and T_7^1.

With this definition, the likelihood is proportional to $R(v, G_N)$, so we can then rewrite our estimator as

$$\widehat{v} = \arg\max_{v \in G_N} \mathbf{P}(G_N | v^* = v) = \arg\max_{v \in G_N} R(v, G_N) \tag{3}$$

Because the ML estimator for the rumor source is also the node which maximizes $R(v, G_N)$, we call this term the *rumor centrality* of the node v, and the node which maximizes it the *rumor center* of the graph. Now, there can be an exponential number of permitted permutations, so it is not clear if this result has made the problem any easier. However, we will see next that there is a simple closed for expression for the rumor centrality.

2.4 Evaluating Rumor Centrality

We now show how to evaluate the rumor centrality $R(v, G_N)$. To begin, we first define a term which will be of use in our calculations.

Definition 2. *$T_{v_j}^v$ is the number of nodes in the subtree rooted at node v_j, with node v as the source.*

To illustrate this definition, a simple example is shown in Figure 1. In this graph, $T_2^1 = 3$ because there are 3 nodes in the subtree with node 2 as the root and node 1 as the source. Similarly, $T_7^1 = 1$ because there is only 1 node in the subtree with node 7 as the root and node 1 as the source.

We now can count the permutations of G_N with v as the source. In the following analysis, we will abuse notation and use $T_{v_j}^v$ to refer to the subtrees and the number of nodes in the subtrees. To begin, we assume v has k neighbors, $(v_1, v_2, ...v_k)$. Each of these nodes is the root of a subtree with $T_{v_1}^v, T_{v_2}^v, ..., T_{v_k}^v$ nodes, respectively. Each node in the subtrees can receive the rumor after its respective root has the rumor. We will have N slots in a given permitted permutation, the first of which must be the source node v. Then, from the remaining $N-1$ nodes, we must choose $T_{v_1}^v$ slots for the nodes in the subtree rooted at v_1. These nodes can be ordered in $R(v_1, T_{v_1}^v)$ different ways. With the remaining $N - 1 - T_{v_1}^v$ nodes, we must choose $T_{v_2}^v$ nodes for the tree rooted at node v_2, and these can be ordered $R(v_2, T_{v_2}^v)$ ways. We continue this way recursively to obtain

$$R(v, G_N) = \binom{N-1}{T_{v_1}^v}\binom{N-1-T_{v_1}^v}{T_{v_2}^v}...\binom{N-1-\sum_{i=1}^{k-1} T_{v_i}^v}{T_{v_k}^v}\prod_{i=1}^{k} R(v_i, T_{v_i}^v) \tag{4}$$

$$= (N-1)! \prod_{i=1}^{k} \frac{R(v_i, T_{v_i}^v)}{T_{v_i}^v!} \tag{5}$$

Now, to complete the recursion, we expand each of the $R(v_i, T_{v_i}^v)$ in terms of the subtrees rooted at the nearest neighbor children of these nodes. To simplify notion, we label the nearest neighbor children of node v_i with a second subscript, i.e. v_{ij}. We continue this recursion until we reach the leaves of the tree. The leaf subtrees have 1 node and 1 permitted permutation. Therefore, the number of permitted permutations for a given tree G_N rooted at v is

$$R(v, G_N) = (N-1)! \prod_{i=1}^{k} \frac{R(v_i, T_{v_i}^v)}{T_{v_i}^v!} \tag{6}$$

$$= (N-1)! \prod_{i=1}^{k} \frac{(T_{v_i}^v - 1)!}{T_{v_i}^v!} \prod_{v_{ij} \in T_{v_i}^v} \frac{R(v_{ij}, T_{v_{ij}}^v)}{T_{v_{ij}}^v!} \tag{7}$$

$$= (N-1)! \prod_{i=1}^{k} \frac{1}{T_{v_i}^v} \prod_{v_{ij} \in T_{v_i}^v} \frac{R(v_{ij}, T_{v_{ij}}^v)}{T_{v_{ij}}^v!} \tag{8}$$

$$= N! \prod_{u \in G_N} \frac{1}{T_u^v} \tag{9}$$

In the last line, we have used the fact that $T_v^v = N$. We thus end up with a simple expression for $R(v, G_N)$ in terms of the size of the subtrees of all nodes in G_N.

2.5 Calculating Rumor Centrality: A Message Passing Algorithm

In order to find the rumor center of a tree graph of N nodes G_N, we need to first find the rumor centrality of every node in G_N. To do this we need the size of the subtrees T_u^v for all v and u in G_N. There are N^2 of these subtrees, but we can utilize a local condition of the rumor centrality in order to calculate all the rumor centralities with only $O(N)$ computation. Consider two neighboring nodes u and v in G_N. All of their subtrees will be the same size except for those rooted at u and v. In fact, there is a special relation between these two subtrees.

$$T_u^v = N - T_v^u \tag{10}$$

For example, in Figure 1, for node 1, T_2^1 has 3 nodes, while for node 2, T_1^2 has $N - T_2^1$ or 4 nodes. Because of this relation, we can relate the rumor centralities of any two neighboring nodes.

$$R(u, G_N) = R(v, G_N) \frac{T_u^v}{N - T_u^v} \tag{11}$$

This result is the key to our algorithm for calculating the rumor centrality for all nodes in G_N. We first select any node v as the source node and calculate the size of all of its subtrees T_u^v and its rumor centrality $R(v, G_N)$. This can be done by having each node u pass two messages up to its parent. The first message is the number of nodes in u's subtree, which we call $t_{u \to parent(u)}^{up}$. The second message is the cumulative product of the size of the subtrees of all nodes in u's subtree, which we call $p_{u \to parent(u)}^{up}$. The parent node then adds the $t_{u \to parent(u)}^{up}$ messages together to obtain the size of its own subtree, and multiplies the $p_{u \to parent(u)}^{up}$ messages together to obtain its cumulative subtree product. These messages are then passed upward until the source node receives the messages. By multiplying the cumulative subtree products of its children, the source node will obtain its rumor centrality, $R(v, G_N)$.

With the rumor centrality of node v, we then evaluate the rumor centrality for the children of v using equation (11). Each node passes its rumor centrality to its children in a message we define as $r_{u \to child(u)}^{down}$. Each node u can calculate its rumor centrality using its parent's rumor centrality and its own subtree size T_u^v. We recall that the the rumor centrality of a node is the number of permitted permutations that result in G_N. Thus, this message passing algorithm is able to count the (exponential) number of permitted permutations for every node in G_N using only $O(N)$ computations.

2.6 Rumor Source Estimator for General Trees

Rumor centrality is an exact ML rumor source estimator for regular trees. In general trees where node degrees may not all be the same, this is no longer the case, as all permitted permutations may not be equally likely. This considerably complicates the construction of the ML estimator. To avoid this complication, we define the following randomized estimator for general trees. Consider a rumor that has spread on a tree and reached all nodes in the subgraph G_N. Then, let the estimate for the rumor source be a random variable $\widehat{v}$ with the following distribution.

$$\mathbf{P}(\widehat{v} = v | G_N) \propto R(v, G_N) \tag{12}$$

This estimator weighs each node by its rumor centrality. It is not the ML estimator as we had for regular trees. However, we will show that this estimator is qualitatively as good as the best possible estimator for general trees.

2.7 Rumor Source Estimator for General Graphs

When a rumor spreads in a network, each node receives the rumor from one other node. Therefore, there is a spanning tree corresponding to a rumor graph. If we knew this spanning tree, we could apply the previously developed tree estimators. However, the knowledge of the spanning tree will be unknown in a general graph, complicating the rumor source inference.

We circumvent the issue of not knowing the underlying rumor spanning tree with the following heuristic. We assume that if node $v \in G_N$ was the source, then it spread the rumor along a breadth first search (BFS) tree rooted at v, $T_{bfs}(v)$. The intuition is that if v was the source, then the BFS tree would correspond to all the closest neighbors of v being infected as soon as possible. With this heuristic, we define the following rumor source estimator for a general rumor graph G_N.

$$\widehat{v} = \arg\max_{v \in G_N} R(v, T_{bfs}(v)) \tag{13}$$

We will show with simulations that this estimator performs well on different network topologies.

3 Detection Probability: A Threshold Phenomenon

This section examines the behavior of the detection probability of the rumor source estimators for different graph structures. We establish that the asymptotic detection probability has a phase-transition effect: for line graphs it is 0, while for trees with finite growth it is strictly greater than 0. We exclude all proofs for this section due to space constraints.

3.1 Line Graphs: No Detection

We first consider the detection probability for a line graph. This is a regular tree with degree 2, so we use the ML estimator for regular trees. We will establish the following result for the performance of the rumor source estimator in a line graph.

Theorem 1. *Define the event of correct rumor source detection after time t on a line graph as $\mathcal{C}_t$. Then the probability of correct detection of the maximum likelihood rumor source estimator, $\mathbf{P}(\mathcal{C}_t)$, scales as*

$$\mathbf{P}(\mathcal{C}_t) = O\left(\frac{1}{\sqrt{t}}\right) \tag{14}$$

The line graph detection probability scales as $t^{-1/2}$, which goes to 0 as t goes to infinity. The intuition for this result is that the rumor source estimator provides very little information because of the line graph's trivial structure.

3.2 Geometric Trees: Non-Trivial Detection

We now consider the detection probability of our estimator in a geometric tree, which is a non-regular tree parameterized by a number α. If we let $n(d)$ denote the maximum number of nodes a distance d from any node, then there exist constants b and c such that $b \leq c$ and

$$bd^{\alpha} \leq n(d) \leq cd^{\alpha} \tag{15}$$

We use the randomized estimator for geometric trees. For this estimator, we obtain the following result.

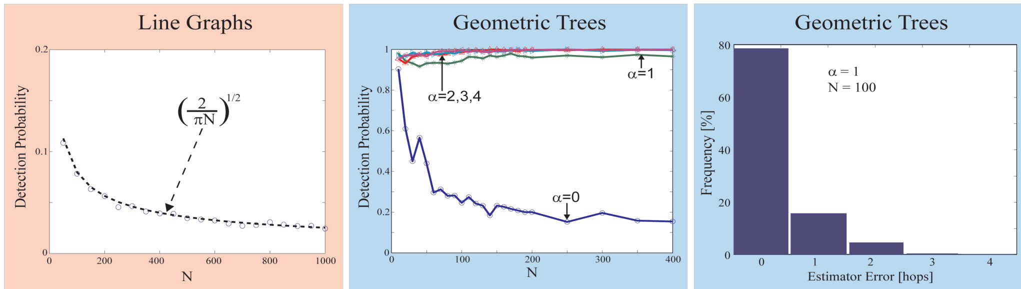

Figure 2: Plots of the rumor source estimator detection probability for line graphs (left) and geometric trees (middle) vs. number of nodes N, and a histogram of the error for a 100 node geometric tree with $\alpha = 1$ (right). In the left plot, the dotted line is a plot of $\sqrt{2/\pi}N^{-1/2}$ and the circles are the empirical error probability. 1000 rumor graphs were generated for each rumor graph size N.

Theorem 2. *Define the event of correct rumor source detection after time t on a geometric tree with parameter $\alpha > 0$ as $\mathcal{C}_t$. Then the probability of correct detection of the randomized rumor source estimator, $\mathbf{P}(\mathcal{C}_t)$, is strictly greater than 0. That is,*

$$\liminf_t \mathbf{P}(\mathcal{C}_t) > 0 \tag{16}$$

This theorem says that $\alpha = 0$ and $\alpha > 0$ serve as a threshold for non-trivial detection: For $\alpha = 0$, the graph is essentially a line graph, so we would expect the detection probability to go to 0 based on Theorem 1, but for any strictly positive α, we always have a strictly positive detection probability.

While Theorem 2 only deals with correct detection, one would also be interested in the size of the rumor source estimator error. The following corollary addresses the estimator error.

Corollary 1. *Consider a rumor that has spread on a geometric tree with parameter $\alpha > 0$. Define $d(\widehat{v}, v^*)$ as the distance between the randomized rumor source estimator $\widehat{v}$ and the actual source v^*. Then, for any $\epsilon > 0$ there exists a finite d_ϵ such that*

$$\liminf_t \mathbf{P}(d(v^*, \widehat{v}) < d_\epsilon) > 1 - \epsilon \tag{17}$$

This corollary says that the error will remain finite with high probability no matter how large the rumor graph is. We will see how small this error is for certain trees in Section 4.

4 Simulation Results

4.1 Tree Networks

We generated 1000 rumor graphs per rumor graph size on an underlying line graph. The detection probability of the rumor source estimator versus the graph size is show in Figure 2. As can be seen, the detection probability decays as $N^{-1/2}$ as predicted in Theorem 1.

We generated 1000 instances of rumor graphs per rumor graph size on underlying geometric trees. The α parameters ranged from 0 to 4. As can be seen in Figure 2, the detection probability of the rumor source estimator remains close to 1 as the tree size grows for $\alpha > 0$ and decays to 0 for $\alpha = 0$, as predicted by Theorem 2. A histogram for a 100 node rumor graph on a geometric tree with $\alpha = 1$ shows that the rumor source estimator error is no larger than 4 hops. This indicates that the estimator error remains bounded, as predicted by Corollary 1.

4.2 General Networks

We performed simulations on synthetic small-world [12] and scale-free [13] networks. These are two very popular models for networks and so we would like our rumor source estimator to perform well on these topologies. We generated 1000 instances of 400 node rumor graphs for each topology. For both topologies, the underlying graph contained 5000 nodes. Figure 3 shows an example of

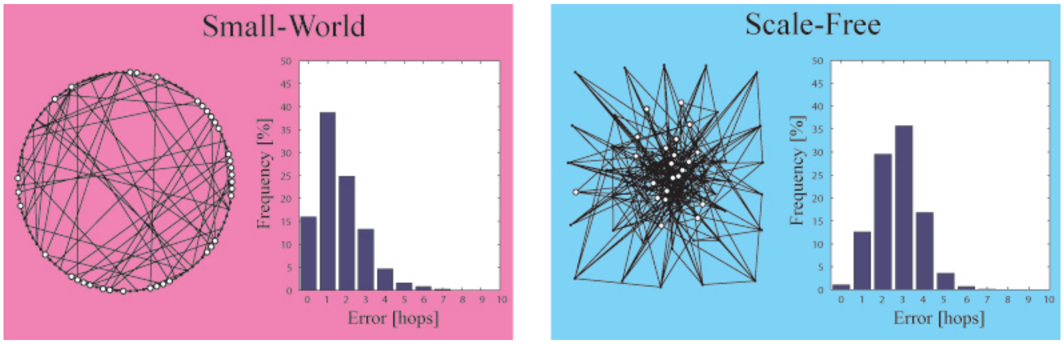

Figure 3: An example of a rumor graph (infected nodes in white) and a histogram of the rumor source estimator error for a 400 node rumor network on small-world (left) and scale-free networks (right). 1000 rumor graphs were generated for each histogram.

rumor spreading in a small-world and a scale-free network. The graphs show the rumor infected nodes in white. Also shown is the histogram of the rumor source estimator error for 400 node rumor graphs in each network. As can be seen, we have correct detection (0 hop error) 15 percent of the time for the small-world network and at most 1 hop error at least 15 percent of the time for the scale-free network. Also, the error in these simulations did not exceed 7 hops, while the average diameters of the rumor graphs were 22 hops for the small-world network and 12 hops for the scale-free network. Thus, we are seeing good performance of the general graph estimator for both small-world and scale-free networks.

5 Conclusion and Future Work

We constructed estimators for the rumor source in regular trees, general trees, and general graphs. We defined the ML estimator for a regular tree to be a new notion of network centrality which we called rumor centrality. Rumor centrality was used as the basis for estimators for general trees and general graphs. We provided a fast, linear time message-passing algorithm to evaluate rumor centrality.

We analyzed the asymptotic behavior of the rumor source estimator for line graphs and geometric trees. For line graphs, it was shown that the detection probability goes to 0 as the network grows in size. However, for geometric trees, it was shown that the estimator detection probability is bounded away from 0 as the graph grows in size and that the estimator error was bounded. Simulations performed on synthetic graphs agreed with these tree results and also demonstrated that the general graph estimator performed well in different network topologies such as small-world and scale-free networks.

There are several future steps for this work. First, we would like to develop estimators for more general rumor spreading models. Second, we would like to have theoretical bounds for the general graph estimator, similar to the results for trees. Third, we would like to test our estimators on real networks to accurately assess their performance.

References

[1] N. T. J. Bailey, *The Mathematical Theory of Infectious Diseases and its Applications*, London, Griffin. (1975).

[2] M. E. J. Newman, "The spread of epidemic disease on networks", *Phys. Rev. E*, 66:016128, (2002).

[3] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),* vol. 2, pp. 1455-1466, (2005).

[4] N. Demiris and P. D. O'Neill, "Bayesian inference for epidemics with two levels of mixing", *Scandinavian J. of Statistics*, vol. 32, pp. 265 - 280 (2005).

[5] G. Streftaris and G. J. Gibson, "Statistical inference for stochastic epidemic models." *Proc. 17th international Workshop on Statistical Modeling*, pp. 609-616. (2002).

[6] P. D. O'Neill, "A tutorial introduction to Bayesian inference for stochastic epidemic models using Markov chain Monte Carlo methods," *Mathematical Biosciences* vol. 180, pp. 103-114. (2002).

[7] N. Demiris and P. D. O'Neill, "Bayesian inference for stochastic multitype epidemics in structured populations via random graphs," *J. Roy. Statist. Soc. B*, vol. 67, pp. 731-745. (2005).

[8] H. Okamura, K. Tateishi, and T. Doshi, "Statistical inference of computer virus propagation using non-homogeneous Poisson processes," *Proc. 18th IEEE International Symposium on Software Reliability*, vol. 5-9, pp. 149 - 158. (2007).

[9] W. Evans, C. Kenyon, Y. Peres, and L. Schulman, "Broadcasting on trees and the Ising model", *Ann. Appl. Prob.*, vol. 10, pp. 410-433. (2000).

[10] E. Mossel, "Reconstruction on trees: beating the second eigenvalue", *Ann. Appl. Prob.*, vol. 11, pp. 285-300. (2001).

[11] A. Gerschenfeld and A. Montanari, "Reconstruction for models on random graphs," *Proc. 48'th IEEE Symp. Found. Comp. Sci.* pp. 194-204 (2007).

[12] D. J. Watts and S. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature* pp. 440-442 (1998).

[13] A. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science* pp. 509-512 (1999).