Identification and Asymptotic Localization of Rumor Sources Using the Method of Types

Himaja Kesavareddigari[1], Sam Spencer[2], Atilla Eryilmaz[1], and R. Srikant[2]

Abstract—We are interested in identifying a rumor source on a tree network. We begin with extended star networks under the SI infection model with exponential waiting times. We present and analyze the *types center*, a highly tractable approximation of the ML source estimate, obtained using the method of types. We empirically show that this approximate ML estimator is exact for some small test cases. We prove that the approximation error is at most logarithmic in infection size on large networks, providing highly efficient source identification (especially compared to the accuracy in similar problems, such as the $\mathcal{O}(\sqrt{n})$ best possible accuracy estimate in a line network). We also show that the qualitative properties of the types and rumor centers are different on extended star networks. We further propose a heuristic-based generalization of this approach to trees: the *relative-leaf counting algorithm*. In simulations on regular and non-regular trees, types center's performance is competitive with rumor centrality (which is optimal for d-regular trees), while requiring less computation time. In addition to providing a faster (and sometimes more accurate) alternative on its own, our approach could potentially be used with rumor centrality to improve results with less than twice the total computation time.

Index Terms—Network problems, Probability and Statistics, Symbolic and algebraic manipulation, Performance evaluation of algorithms and systems, Heuristic design, Graph and tree search strategies, Trees

1 Introduction

The propagation of infections in contagion networks is an important problem that arises in many different contexts. Processes such as information dissemination via rumors, creation of cultural fads, spread of computer viruses, and similar phenomena can be studied by modeling one or more infections on a graph of the underlying connections.One longstanding class of problems (dating back at least as far as epidemiological studies of the London cholera epidemic of 1854) is the identification of infection sources and their effect on subsequent propagation.

Due to the large number of nodes and connections that are typically present in such networks, source identification must be computationally efficient. In current literature, the source identification problem has been studied under varying assumptions about the infection process, the class of graphs, the number of sources, and the information available regarding the infection state.

Literature discussing centrality measures such as the Jordan center and rumor center are relevant in the context of this work. In particular, [1] proposes the *rumor centrality* measure which is proven to be optimal for d-regular trees. Rumor centrality and its implementation through the message-passing algorithm proposed in [1] are, therefore, a natural point of comparison for the approaches that we propose in this paper.

The accuracy of rumor centrality as an ML estimator for various graph classes is discussed in [1], [2], [3], [4] for the SI (susceptible-infected) infection model. On the other hand, the universality of the Jordan center for estimating the location of a single source in a tree network for SI, SIS, SIR and SIRI models is presented in [5]. In [6], [7], assuming a single source and SI model on tree networks, when the set of the infected nodes is only partially revealed, the Jordan center is shown to be the source estimator that starts the infection along its most probable infection path among the revealed nodes.

The location of a single infection source, given the infection snapshot, is studied for SIS, SIR and SIRI infection models in [8], [9] and [10], respectively. However, the study of infection sources is not restricted to the assumption of a single source. Unlike the estimation of a single rumor source on a line graph with an SI infection model, the localization of *two* rumor sources for the same model is shown to be impossible in [11]. In [12], the number of sources in a tree network is estimated. The problem of identifying multiple rumor sources with different start times is studied in [13]. This work proposes a two-source joint estimation algorithm that utilizes any known single source estimation algorithm. The joint source estimation algorithm is shown to converge to a local optimum of the estimation function when the network is a quasi-regular tree with respect to the choice of single source estimator.

In [14], [15] and [16], the problem of identifying multiple infection sources and their respective infection partitions is studied under the assumption that the order in which the nodes are infected is known. When the number of sources is known to be two, the algorithm is shown to identify

1 *H. Kesavareddigari and A. Eryilmaz are with the Electrical and Computer Engineering Department, The Ohio State University, Columbus, OH 43210, USA* {kesavareddigari.1, eryilmaz.2}@osu.edu

2 *S. Spencer and R. Srikant are with the Coordinated Science Laboratory and the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA* {samster, rsrikant}@illinois.edu

This work was supported by the DTRA grants: HDTRA1-15-1-0003 and HDTRA1-18-1-0050; and the NSF grants: CNS-NeTS-1514260, CNS-NeTS-1717045, CMMI-SMOR-1562065, CNS-ICN-WEN-1719371, CNS-SpecEES-1824337, NSF NeTS 17-18203, NSF CPS ECCS 17-39189, NSF CMMI 15-62276, NSF ECCS 16-09370, and ARO W911NF-16-1-0259.

the infection sources with probability approaching 1 as the network size increases. The problem of identifying multiple infection sources and infection regions is studied for the SIR infection model in [17], [18] and [19].

A time-varying network under the SIR infection model is studied in [20]. At the macroscopic level, partial observations of a node's geographic location, connectivity, and its infection state are aggregated to create static but partial estimates of the network in limited time windows. The nodes that can facilitate complete rumor propagation paths are called 'suspects'. Lastly, the likelihood of each 'suspect' being the source is computed, and the most likely 'suspect' node is identified as the rumor source. The recent work [21], also uses a macroscopic model to study the minimum number of messenger nodes needed for rumor source identification on directed and undirected graphs. Identifying groups of individuals as single nodes, with infection states defined by the fraction of local population infected, the number of messenger nodes required is calculated using observability theory. An undirected, scale-free graph is shown to node only one messenger node.

While [1], [2], [3] and [4] use a message passing algorithm for rumor source detection on trees and extend these results to general graphs by using their spanning trees, [22] and [23] solve belief propagation equations and dynamic message passing equations, respectively, on general graphs to detect the source. The equations for belief propagation and DMP algorithms are exact for trees.

The complementary problem of hiding the rumor source to impede reliable detection has also been studied. If a significant fraction of the nodes are equally likely to be sources, then probability of source detection might not be very high. Therefore, an intelligent source can design strategies to spread information while diminishing the probability of detection. The adaptive diffusion model in [24], [25] and [26] hides the source perfectly in an infinite, regular tree and for irregular trees limits the detection probability. In [27], a strategic game between the rumor source and its adversary, the rumor source locator, is designed.

Our work is based on the single source estimator for a stylized model (*extended star network*) proposed in [28] (where it is referred to as a "star network"). In [28], the ML estimate of a single rumor source is analyzed based on an infection snapshot of the extended star network under the SI infection model. Using the method of types, a tractable approximation to the ML source is identified as the "ML center" (which we call the *types center* in this work, to avoid confusion). The types center is argued to be asymptotically accurate. In addition, numerical results indicate that the types center might be accurate even for small infection sizes.

In this work, we rigorously prove that for large extended star networks, the distance between the ML source and the types center is at most logarithmic in infection size. Since the types center offers a computationally tractable, yet accurate approximation, we use a heuristic to extend the types center measure into a method for finding sources on general trees (the *heuristic types center*), and design an algorithm to carry out the method. The performance and accuracy of the resulting *relative-leaf counting algorithm* are compared to the message-passing algorithm for rumor centrality, on both regular and non-regular trees (rumor centrality is proven to be the optimal ML source estimator for infections on regular trees).

In Section 2.1, we outline the characteristics of an extended star network and the assumptions of our model. In Section 2.2, we formulate the "true" maximum likelihood (ML) estimator of the rumor source. In Section 2.3, we recount the derivation of the types center, a method of types-based approximation to the ML source estimate. In Section 2.4, we analyze the approximation error incurred by the types center, proving that its deviation from the ML source is at most logarithmic in the size of the infection. In Section 2.5, we discuss the qualitative properties of the types center, and show some contrasts with the properties of the rumor center for extended star networks.

Furthermore, in Section 3 we extend the principle of the types center to design the heuristic types center source estimator for general trees. In Section 3.1, we lay out a specific procedure for computing this center. The computational efficiency and the accuracy of the heuristic types center are compared to those of the rumor center in Sections 3.2 and 3.3, respectively.

2 Source Estimation in Extended Star Networks

2.1 Model

In this work, we define an extended star network as a *hub node*, O, with m "arms" of nodes proceeding outward from O. The nodes of each arm will be numbered starting with 1 (for the node adjacent to O) and increasing from there.

We use the SI infection model with edge-based propagation in continuous time to describe the spreading of the rumor. That is, nodes are either "susceptible" (have not yet heard the rumor) or "infected" (have already heard it). If a susceptible node shares an edge with an infected neighbor, then the infection will "traverse" that edge and infect the susceptible node with a waiting time that is exponentially distributed with mean T. Once infected, a node remains that way indefinitely. An important consequence of this model is that we can invoke the memoryless property of the system to state that at any given time, the next infection is equally likely to occur along any outgoing edge from the current infected set. For a given observed infection pattern (the subgraph of infected nodes at some point in time), we wish to find the maximum likelihood estimate of the source giving rise to that infection pattern.

Our infection pattern consists of O, along with the closest k_i nodes along each arm i. If the infection were confined to $m = 1$ or 2 arms, we could simply consider the problem on a line graph, and the ML solution is well-known to be the midpoint of the infection (in fact, for a uniform prior, the likelihood function follows a binomial distribution on the infected nodes [1], [11]). Since the infection arises from a single source, it must be contiguous, so any infection which spans multiple arms must also include O.

2.2 ML Source Estimation in Extended Star Networks

For a given infection pattern (as described in Section 2.1), we compute the likelihood of the observed pattern occurring at some point in time, given that the rumor originated at O.

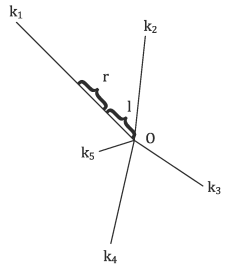


Fig. 1: Illustrating the calculation in (3). Note that r can range from 0 up to $k_1 - l$.

Each of the m arms is set to acquire an additional infected node after its own IID waiting time. Since the waiting time RVs are memoryless (because they are exponentially distributed) and independent, with identical infection rates, $\frac{1}{T}$, each of the m arms has an equal probability ($\frac{1}{m}$) of acquiring the next infected node in the sequence.

Let $\kappa := (k_1, k_2, \cdots, k_m)$ and $K = \sum_{i=1}^{m} k_i$. Then the probability of observing k_i infections along arm i is given by a multinomial distribution.

$$P(O;\kappa) = \frac{K!}{k_1!k_2!\ldots k_m!m^K} \tag{1}$$

If instead, the rumor source is located along one of the arms (let us assume, without loss of generality, that the source is located on arm 1) at node l, then the propagation of the rumor occurs in two phases: At first, the infection spreads along arm 1 in either direction, until the inward propagation reaches O. At that point, it can spread outward along any of the m arms. Accordingly, we decompose the set of possibilities according to the extent that the infection proceeds outward along arm 1 before the inward propagation reaches O. Suppose the infection reaches an additional r nodes beyond l before reaching O, as shown in Fig. 1. Then the probability of r outward infections and $l-1$ inward infections (in any order) followed the last inward infection reaching O is $\frac{\binom{r+l-1}{l-1}}{2^{r+l}}$. Afterwards, the probability of fulfilling the remaining infections exactly can be computed using (1), replacing k_1 with $k_1 - (r+l)$. Multiplying these two probabilities, we obtain

$$\begin{aligned} P(l,r;\kappa) &= \frac{\binom{r+l-1}{l-1}}{2^{r+l}} \frac{(K-(r+l))!}{(k_1-(r+l))!k_2!\ldots k_m!m^{K-(r+l)}} \\ &= \frac{(r+l-1)!}{r!(l-1)!2^{r+l}} \frac{(K-(r+l))!}{(k_1-(r+l))!k_2!\ldots k_m!m^{K-(r+l)}}. \end{aligned} \tag{2}$$

Summing over all possible values of r, we obtain

$$\begin{aligned} P(l;\kappa) &= \sum_{r=0}^{k_1-l} \frac{\binom{r+l-1}{l-1}}{2^{r+l}} \frac{(K-(r+l))!}{(k_1-(r+l))!k_2!\ldots k_m!m^{K-(r+l)}} \\ &= \sum_{r=0}^{k_1-l} \frac{(r+l-1)!}{r!(l-1)!2^{r+l}} \frac{(K-(r+l))!}{(k_1-(r+l))!k_2!\ldots k_m!m^{K-(r+l)}}. \end{aligned} \tag{3}$$

2.3 Derivation of the Method of Types Approximation, Types Center

In order to further analyze the situation, we will use the method of types. For a source with a uniform distribution on X, the probability of observing a type T of $\mathbf{n_s}$ samples with empirical distribution Q satisfies

$$\frac{1}{(n_s+1)^{|\chi|}} 2^{-n_s D(Q||U_X)} \leq P(T^{n_s}(Q)) \leq 2^{-n_s D(Q||U_X)}, \tag{4}$$

where $|\chi|$ is the size of the set of choices [29]. Note that the lower and upper bounds are the same except for the leading term in the lower bound. Therefore, we will work with the upper bound for now, and consider the effect of the leading term in the lower bound afterwards.

$$\begin{aligned} P(T^{n_s}(Q)) \leq & 2^{-n_s D(Q||U_X)} = 2^{-n_s(\log|X| - H(Q))} \\ = & 2^{-n_s(\log|X| + \sum_X Q(x)\log Q(x))}. \end{aligned} \tag{5}$$

Applying this to each of the phases of the infection yields

$$P(l;\kappa) = \sum_{r=0}^{k_1-l} 2^{-h(l,r;\kappa)} \tag{6}$$

$$\text{where, } h(l,r;\kappa) = h_1(l,r;\kappa) + h_2(l,r;\kappa) \text{ with,} \tag{7}$$

$$h_1(l,r;\kappa) = (r+l)\left(1 - H\left(\frac{r}{r+l}, \frac{l}{r+l}\right)\right) \tag{8}$$

$$\begin{aligned} h_2(l,r;\kappa) = (K-(r+l))\Bigg[& \log m \\ & - H\left(\frac{k_1-(r+l)}{K-(r+l)}, \frac{k_2}{K-(r+l)}, \ldots, \frac{k_m}{K-(r+l)}\right)\Bigg]. \end{aligned} \tag{9}$$

Remember that we are interested in finding the value of l that maximizes this expression, and observe that the value of the sum is asymptotically dominated by the term with the largest (or least negative) exponent. Furthermore, notice that the terms of the sum depend only on $r+l$ rather than r or l individually, with the exception of the $H(\frac{r}{r+l}, \frac{l}{r+l})$ in the first exponent. This value is maximized when $r = l$. Therefore, we can conclude that the dominant term of the sum for the maximizing value of l occurs when $r = l$. If this were not the case, we could replace r with r' and l with l', where $r' = l' = \frac{r+l}{2}$ and obtain a more dominant term with a different value of l. Accordingly, we will replace r with l going forward, and in the process, we eliminate the first part of the dominant term.

$$P(l;\kappa) \leq 2^{-(K-2l)(\log m - H(\frac{k_1-2l}{K-2l}, \frac{k_2}{K-2l}, \ldots, \frac{k_m}{K-2l}))} \tag{10}$$

$$\leq 2^{-(K-2l)\log m + (K-2l)H(\frac{k_1-2l}{K-2l}, \frac{k_2}{K-2l}, \ldots, \frac{k_m}{K-2l})}$$

$$\leq 2^{-(K-2l)\log m - (K-2l)(\frac{k_1-2l}{K-2l}\log\frac{k_1-2l}{K-2l} + \sum_{i=2}^{m}\frac{k_i}{K-2l}\log\frac{k_i}{K-2l})}$$

$$= 2^{-(K-2l)\log m - (k_1-2l)\log\frac{k_1-2l}{K-2l} - \sum_{i=2}^{m} k_i \log\frac{k_i}{K-2l}} \tag{11}$$

Taking the exponent, and setting the derivative with respect to l to zero, we obtain

$$0 = 2\log m + 2\log(k_1 - 2l) + 2 - 2\log(K-2l) - 2,$$

which leads to,

$$l = \frac{k_1 - \mu_{-1}}{2}, \text{ where } K_{-1} = K - k_1,\ \mu_{-1} = \frac{K_{-1}}{m-1}. \quad (12)$$

In other words, l is chosen so that the remaining length of arm 1 once O is reached is equal to the arithmetic mean of the other arms. Since we can apply this reasoning to any arm, we have a local maximum for any arm whose length is above average. However, the form of the exact solution in (3) makes it clear that the global maximum is attained when the longest arm is chosen to be arm 1. Let us denote this choice of l as the *types center*, $\hat{l}$.

Having chosen $\hat{l}$ to optimize the upper bound, let us consider the effect of the leading coefficient in the lower bound. While it is different for each term of the sum in (6), it can be bounded from below by $\frac{1}{(K+1)^m}$. We can then use our earlier reasoning with (10) to show that

$$P(l;\kappa) \geq \frac{1}{(K+1)^m} 2^{-(K-2l)(\log m - H(\frac{k_1-2l}{K-2l}, \frac{k_2}{K-2l}, \ldots, \frac{k_m}{K-2l}))}. \quad (13)$$

Consider what happens if we allow the pattern to grow larger, but maintain the relative sizes of the k's (in other words, replace each k by nk, and let n go to infinity). Then

$$P(l;n\kappa) \leq 2^{-(nK-2l)(\log m - H(\frac{nk_1-2l}{nK-2l}, \frac{nk_2}{nK-2l}, \ldots, \frac{nk_m}{nK-2l}))}, \quad (14)$$

and

$$P(l;n\kappa) \geq \frac{2^{-(nK-2l)(\log m - H(\frac{nk_1-2l}{nK-2l}, \frac{nk_2}{nK-2l}, \ldots, \frac{nk_m}{nK-2l}))}}{(nK+1)^m}. \quad (15)$$

Remember that the types center, $\hat{l}$, was chosen (proportional to the k_i's) in such a way as to minimize the (negated) exponent in (11). Therefore, letting $l = n\hat{l}$ will yield the minimum exponent in (14) and (15) (n times the old optimal exponent). If, instead, we were to choose $l = l' \neq n\hat{l}$ (relative to the k's), then the higher (negated) exponent will eventually cause the upper bound in (14) evaluated at l' to drop below the lower bound in (15) evaluated at $\hat{l}$. Therefore, for sufficiently large instances, our choice of $l = \hat{l}$ must be optimal. In fact, our empirical results suggest that this is the case even for smaller instances.

2.4 Error Analysis for the Types Center Approximation to the ML Source Estimate

Continuing from the previous section, we are now interested in the asymptotic performance of our source estimator. For some $n \in \mathbb{N}$, we define a new system with $\kappa \mapsto \kappa_n := n\kappa$, $l \in \{0, 1, \cdots, nk_1\}$, and $r \in \{0, 1, \cdots, nk_1 - l\}$, for a fixed l. In this system, the ML source (l^*) and the types center ($\hat{l}$) are as follows.

$$l_n^* := \underset{l \in \{0,1,\cdots,nk_1\}}{\arg\max}\ P(l;\kappa_n).$$
$$\hat{l}_n := \underset{l \in \{0,1,\cdots,nk_1\}}{\arg\max}\ 2^{-h(l,l;\kappa_n)} = n\hat{l}.$$

We want to find an upper bound on the deviation of the types center from the ML source, $\epsilon(\kappa_n) := |l_n^* - \hat{l}_n|$, especially for large n, when arm 1 is the longest arm.

Theorem 1. *Given n and κ for an extended star network, $\exists N_0 \in \mathbb{N}$ such that $\frac{\log_e(N_0K+1)}{N_0} \leq \frac{1}{m^2}\left(\mu_{-1} - \frac{1}{m}\right)$,* [1]

$$\Pr\{\epsilon(\kappa_n) \leq m^4 \mu_{-1} \log_e(nK+1)\} = 1,\ \forall n \geq N_0. \quad (16)$$

That is, $|l_n^* - \hat{l}_n| \leq m^4 \mu_{-1} \log_e(nK+1)$ (the distance between the types center and the ML source grows only logarithmically in network size).

Proof of Theorem 1. We extend the bounds on $P(l;\kappa_n)$ to the functions $U(l;\kappa_n)$ and $L(l;\kappa_n)$ that have a similar form.

Lemma 1. *$\forall\ \kappa, n$ and $l \in \{0, 1, \cdots, nk_1\}$,*

$$\begin{aligned} P(l;\kappa_n) &\leq (nK+1)^m 2^{-h(r_l^*, l;\kappa_n)} \\ &\leq (nK+1)^m 2^{-h_2(r_l^*, l;\kappa_n)} =: U(l;\kappa_n), \\ P(l;\kappa_n) &\geq (nK+1)^{-m} 2^{-h(l,l;\kappa_n)} \\ &= (nK+1)^{-m} 2^{-h_2(l,l;\kappa_n)} =: L(l;\kappa_n), \end{aligned}$$

where $r_l^* := \arg\min_r h(r, l;\kappa_n)$.

The remainder of the proof builds on these relaxed bounds on the probability $P(l;\kappa_n)$. Each of the Lemmas 2 and 3, and Results 1 – 8 presents either a behavioral property, or an alternate representation of the quantities related to $P(l;\kappa_n)$. (The proofs of Lemmas 1, 3 and the derivations of Results 1, 3 – 8 can be found in the Appendix.)

The properties of l_n^*, $\hat{l}_n$ and the bounds from Lemma 1 impose the constraint in Lemma 2 and Figure 2.

Lemma 2. *For fixed n, κ and $l^c \in \{0, 1, \cdots, nk_1\}$:*

$$U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n) \Rightarrow l_n^* \neq l^c \text{ w.p. } 1.$$

Proof of Lemma 2. By definition, $P(l_n^*;\kappa_n) \geq P(l;\kappa_n)$, $\forall l, n, \kappa$.

$$\begin{aligned} &\Longrightarrow U(l_n^*;\kappa_n) \geq P(l_n^*;\kappa_n) \geq P(l;\kappa_n) \geq L(l;\kappa_n)\ \forall\ l, \kappa_n. \\ &\Longrightarrow U(l_n^*;\kappa_n) \geq L(l;\kappa_n)\ \forall\ l, \kappa_n. \\ &\Longrightarrow U(l_n^*;\kappa_n) \geq L(\hat{l}_n;\kappa_n)\ \forall\ \kappa_n. \end{aligned}$$

So, for any valid in l^c in κ_n,

$$U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n) \Rightarrow P\{l_n^* \neq l^c;\kappa_n\} = 1.$$

□

$\{l^c : U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n)\}$ contains some values of $l^c \neq l_n^*$. Therefore, its complement contains l_n^*, which guides our derivation of an upper bound on $|l_n^* - \hat{l}_n|$.

Fix $l^c \in \{0, 1, \cdots, nk_1\}$. Let $r_{l^c}^* = \arg\min_r h(r, l^c;\kappa_n)$. From Lemma 1,

$$\begin{aligned} U(l^c;\kappa_n) &= (nK+1)^m 2^{-h_2(r_{l^c}^*, l^c;\kappa_n)} \\ L(l;\kappa_n) &= (nK+1)^{-m} 2^{-h_2(l,l;\kappa_n)}. \end{aligned}$$

If $U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n)$, then

$$\begin{aligned} &(nK+1)^m 2^{-h_2(r_{l^c}^*, l^c;\kappa_n)} < (nK+1)^{-m} 2^{-h_2(\hat{l}_n, \hat{l}_n;\kappa_n)} \\ &\Longrightarrow\ h_2(r_{l^c}^*, l^c;\kappa_n) - h_2(\hat{l}_n, \hat{l}_n;\kappa_n) > 2m \log(nK+1). \end{aligned} \quad (17)$$

1. In this paper, $log(n)$ is used to denote $log_2(n)$ and $log_e(n)$ is used to denote the natural logarithm.

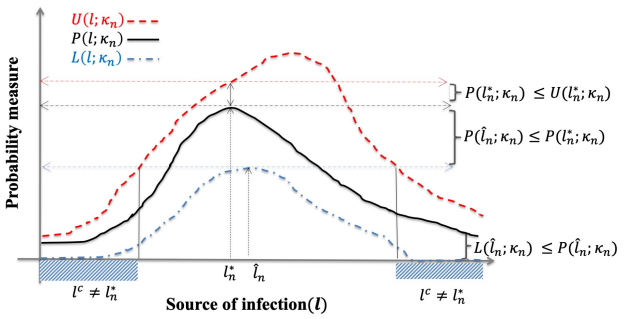


Fig. 2: $U(l_n^*;\kappa_n) \ge L(\hat{l}_n;\kappa_n),\ \forall\, n,\kappa$, for the ML source l_n^*.

Let $l^c \mapsto \rho_n^{l^c} := \frac{1}{n}(r_{l^c}^* + l^c - 2\hat{l}_n) \in [-(k_1-\mu_{-1}),\ \mu_{-1}]$. We can define the following:

$$\begin{aligned} nK - r_{l^c}^* - l^c &= n(K - 2\hat{l} - \rho_n^{l^c}) \\ &= n(m\mu_{-1} - \rho_n^{l^c}) =: n\mathbf{K}(\rho_n^{l^c}) \text{ and} \qquad (18) \\ nk_1 - r_{l^c}^* - l^c &= n(k_1 - 2\hat{l} - \rho_n^{l^c}) \\ &= n(\mu_{-1} - \rho_n^{l^c}) =: n\mathbf{k}_1(\rho_n^{l^c}). \qquad (19) \end{aligned}$$

With the definitions of $\mathbf{K}(\rho_n^{l^c})$ and $\mathbf{k}_1(\rho_n^{l^c})$, we can rewrite:

$$\frac{nk_1 - r_{l^c}^* - l^c}{nK - r_{l^c}^* - l^c} = \frac{\mathbf{k}_1(\rho_n^{l^c})}{\mathbf{K}(\rho_n^{l^c})} \text{ and } \frac{nk_i}{nK - r_{l^c}^* - l^c} = \frac{k_i}{\mathbf{K}(\rho_n^{l^c})}.$$

Since $K_{-1} > 0$ and $r_{l^c}^* + l^c \le nk_1$, we have

$$\mathbf{K}(\rho_n^{l^c}) > \mathbf{k}_1(\rho_n^{l^c}) > 0 \implies 0 < \frac{K_{-1}}{\mathbf{K}(\rho_n^{l^c})} < 1. \qquad (20)$$

We derive Results 1 and 2 to compare algebraically $h_2(r_{l^c}^*, l^c;\kappa_n) - h_2(\hat{l}_n, \hat{l}_n;\kappa_n)$.

Result 1. *Rewrite* $G(\rho_n^{l^c}) := \frac{h_2(r_{l^c}^*, l^c;\kappa_n)}{n}$ *as:*

$$\begin{aligned} G(\rho_n^{l^c}) =& \mathbf{K}(\rho_n^{l^c}) \log m + \mathbf{k}_1(\rho_n^{l^c}) \log \mathbf{k}_1(\rho_n^{l^c}) \\ &- \mathbf{K}(\rho_n^{l^c}) \log \mathbf{K}(\rho_n^{l^c}) + \sum_{i=2}^{m} k_i \log k_i. \qquad (21) \end{aligned}$$

Result 2. *Rewrite* $\frac{h_2(\hat{l}_n, \hat{l}_n;\kappa_n)}{n} = G(0)$ *with* $\rho_n^{l^c} = 0$ *as:*

$$\begin{aligned} G(0) =& \mathbf{K}(0) \log m + \mathbf{k}_1(0) \log \mathbf{k}_1(0) \\ &+ \sum_{i=2}^{m} k_i \log k_i - \mathbf{K}(0) \log \mathbf{K}(0). \qquad (22) \end{aligned}$$

Using Results 1 and 2, we obtain a constraint on $\rho_n^{l^c}$.

Result 3. *Define* γ_n *with* $\frac{\gamma_n}{\log_e(2)} = \frac{2m}{n}\log(nK+1)$. *Then*

$$\begin{aligned} \frac{\gamma_n}{\log_e(2)} <& -\rho_n^{l^c} \log m + \mu_{-1} \log\left(1 - \frac{\rho_n^{l^c}}{\mu_{-1}}\right) \\ &- m\mu_{-1} \log\left(1 - \frac{\rho_n^{l^c}}{m\mu_{-1}}\right) - \rho_n^{l^c} \log\left(1 - \frac{K_{-1}}{\mathbf{K}(\rho_n^{l^c})}\right). \qquad (23) \end{aligned}$$

We have a constraint on $\rho_n^{l^c}$, but we are interested in the range of $l^c - \hat{l}_n$. Since $\rho_n^{l^c} = \frac{1}{n}(r_{l^c}^* + l^c - 2\hat{l}_n)$, we obtain and utilize suitable bounds on $\rho_n^{l^c}$ in terms of $l^c - \hat{l}_n$.

Lemma 3.

a) If $\rho_n^{l^c} \ge 0$, $l^c \ge \hat{l}_n$, then $\frac{2(l^c - \hat{l}_n)}{n} \ge \rho_n^{l^c} \ge 0$.

b) If $\rho_n^{l^c} \le 0$, $l^c \le \hat{l}_n$, then $\frac{2(l^c - \hat{l}_n)}{n} \le \rho_n^{l^c} \le 0$.

Combining the constraint on $\rho_n^{l^c}$ from Result 3 and Lemma 3a), we obtain a bound when $l^c - \hat{l}_n \ge 0$. (Claim 1)

Claim 1. $0 \le l_n^* - \hat{l}_n \le m^4\mu_{-1}\log_e(nK+1)$.

Proof of Claim 1.

Examine $U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n)$ for $l^c - \hat{l}_n \ge 0$ and $\rho_n^{l^c} \ge 0$. Choose $0 \le \rho_n^{l^c} < \mu_{-1}$.

Result 4. *Applying (20),* $\frac{-x}{1-x} \le \log_e(1-x) \le -x$ *for* $0 \le x < 1$ *on* $\frac{\gamma_n}{\log_e(2)} < G(\rho_n^{l^c}) - G(0)$, *we get:*

$$\rho_n^{l^c\,2}[(m-1)\log_e m - 1] + \rho_n^{l^c} f(\kappa) - K_{-1}\gamma_n > 0. \qquad (24)$$

where $f(\kappa) := K_{-1}(m - 1 - \log_e m) + \mu_{-1} + (m-1)\gamma_n \ge 0$.

Result 5. *Using* $\rho_n^{l^c} \ge 0$, $l^c \ge \hat{l}_n$ *on the range of (24):*

$$\frac{l_n^* - \hat{l}_n}{n} \le \frac{\gamma_n}{(m - 1 - \log_e m)}. \qquad (25)$$

Note: $(m-1)\log_e m - 1 \ge 0$ and $m - 1 - \log_e m \ge 2 - \log_e 3 \ge \frac{1}{2} \ge 0$ for $m \ge 3$.

Therefore, $l_n^* - \hat{l}_n \le \frac{n\gamma_n}{m - 1 - \log_e m} \le 4m\log_e(nK+1)$. Since $\mu_{-1} \ge \frac{1}{m-1}$ and $m \ge 3$, we get $m^4\mu_{-1} \ge 9m$.

$$\therefore\ 0 \le l_n^* - \hat{l}_n \le m^4\mu_{-1}\log_e(nK+1). \qquad (26)$$

□

Claim 2. *For* $N_0 \in \mathbb{N}$: $\frac{\log_e(N_0K+1)}{N_0} \le \frac{1}{m^2}\left(\mu_{-1} - \frac{1}{m}\right)$,

$0 \le -(l_n^* - \hat{l}_n) \le m^4\mu_{-1}\log_e(nK+1),\ \forall\, n \ge N_0$.

Proof of Claim 2.

Examine $U(l^c;\kappa_n) < L(\hat{l}_n;\kappa_n)$ for $l^c - \hat{l}_n \le 0$ and $\rho_n^{l^c} \le 0$. Let $\tau := -\rho_n^{l^c} \ge 0$.

Result 6. *Applying (20),* $\frac{x}{1+x} \le \log(1+x) \le x$ *for* $x \ge 0$ *and* $\log(1-x) \le -x$ *for* $0 \le x < 1$ *on* $\frac{\gamma_n}{\log_e(2)} < G(\rho_n^{l^c}) - G(0)$, *we get:*

$$\tau^2 \log_e 2m + f_-(\kappa)\tau - m\mu_{-1}\gamma_n > 0, \qquad (27)$$

where $f_-(\kappa) := m\mu_{-1}\log_e m - K_{-1} - \gamma_n$.

Result 7. *Using* $\rho_n^{l^c} = -\tau \le 0$, $l^c \le \hat{l}_n$ *on the range of (27) for* $f_-(\kappa) \ge 0$:

$$l_n^* - \hat{l}_n \ge -\frac{2m^2\mu_{-1}\log_e(nK+1)}{f_-(\kappa)}. \qquad (28)$$

Result 8. $f_-(\kappa) + \gamma_n = m\mu_{-1}\log_e m - K_{-1} \ge 0$ *and* $\gamma_n \searrow 0$ *as* $n \to \infty$. *So* $f_-(\kappa) \ge 0$ *for sufficiently large* n.

In fact, for $N_0 \in \mathbb{N}$: $\frac{\log_e(N_0K+1)}{N_0} \le \frac{1}{m^2}\left(\mu_{-1} - \frac{1}{m}\right)$,

$$f_-(\kappa) \ge \frac{2}{m^2},\ \forall n \ge N_0.$$

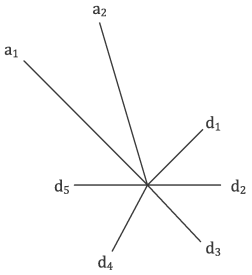


Fig. 3: A neuron with two axons and five dendrites.

From (27) and Result 8,

$$0 \leq -(l_n^* - \hat{l}_n) \leq m^4 \mu_{-1} \log_e(nK+1),\ \forall n \geq N_0. \quad (29)$$

□

Combining Equations (26) and (29), we obtain the statement of the theorem given in Equation (16). □

2.5 Qualitative Properties of the Types and Rumor Centers in Extended Star Networks

In this section, we consider some qualitative properties of the types center, and contrast them with those of the rumor center [1]. We consider a special case of the extended star network. Based on the visual similarity to a biological structure, let us define a *neuron* as a region of an extended star network that includes O, and whose arms only take on two distinct lengths. The shorter arms, called *dendrites*, have length L_0, while the longer arms, called *axons*, have length $L_0 + L_1$, and we stipulate that both L_0 and L_1 are strictly greater than 0. This is illustrated in Fig. 3.

Suppose our infection pattern is a neuron with a single axon. If there is only a single dendrite as well (meaning $m = 2$), then effectively this is the same as a line graph of length $2L_0 + L_1$. In this case, the distance center and the types center would both be on the axon at node $L_1/2$.

Firstly, consider an increase in the number of dendrites (and the corresponding m). (Note that this is different from choosing a larger m to begin with and allowing the "extra" arms to have length 0 at the outset.) The types center remains at $2L_0 + L_1$, because the second term in the numerator of (12) remains constant. However, the distance center would begin to move towards O, and eventually reach it and stay there. Thus, the types center and the distance center will no longer be the same. In contrast, [1] tells us that the distance center and rumor center are the same if the latter is unique, so we know that the types center (which is unique in this case) and the rumor center cannot be equivalent. (This is not inconsistent with the findings in [1], since they do not claim the rumor center to be optimal in the ML sense for non-regular trees, but we have identified a fairly simple yet clear example where these two may differ significantly.)

To illustrate a second significant difference between these two centrality measures, consider a neuron with n dendrites and $n+1$ axons, where $n > 2$. It can be easily shown that the (unique) rumor center in this case is at O (otherwise there would be at least n equivalent rumor centers by symmetry, while [1] guarantees us that a tree can have at most 2 rumor centers). However, (12) tells us that there are $n+1$ equiprobable types centers, located at node $L_1/4$ on each of the axons.

TABLE 1: ML source estimate coincides with types center.

m	Arm Lengths	ML source (Assume Arm 1)
3	4, 2, 2	1
3	20, 10, 10	5
3	200, 100, 100	50
3	40, 40, 20	5 (Arms 1 & 2)
3	200, 200,100	25 (Arms 1 & 2)
3	20, 20, 0	5 (Arms 1 & 2)
3	100, 100, 0	25 (Arms 1 & 2)
3	300, 200, 100	75
3	5, 1, 1	2
3	50, 10, 10	20
3	500, 100, 100	200
4	500, 100, 100, 100	200
5	500, 100, 100, 100, 100	200
5	500, 500, 100, 100, 100	150 (Arms 1 & 2)
5	160, 120, 80, 0, 0	55
5	1000, 800, 600, 400, 200	250

2.5.1 Computational Results

Since the types center derivation in Section 2.3 relies on large deviation theory, we include some computational results in Table 1. These results were derived using the exact combinatorial expression in (3), not the subsequent approximations. Note that in each case, the types center coincides exactly with the ML source estimate.

The early examples show how the results scale for different sized regions with the same proportions, and suggest that the types center in (12) often works exactly, even for very small cases (despite the fact that we used large deviation methods to derive it). The later ones show the results to hold for larger m and more diverse arm lengths. (Of course, this table only represents a tiny fraction of possible arm length combinations, which are all reasonably small, and were chosen to have results that are easy to interpret, so it is entirely possible that not every case will work this well, especially at larger scales.)

3 Heuristics for Extension of the Types Center Metric to General Trees

In this section, we extend the definition of the types center to cases with a single infection source and a general tree as the underlying network. We propose an algorithm to compute this heuristic types center for an infection on a tree network. We prove that, for an extended star network, the algorithm converges to the types center defined previously. We provide an analysis of the computational complexity involved, and compare the accuracy of the heuristic types

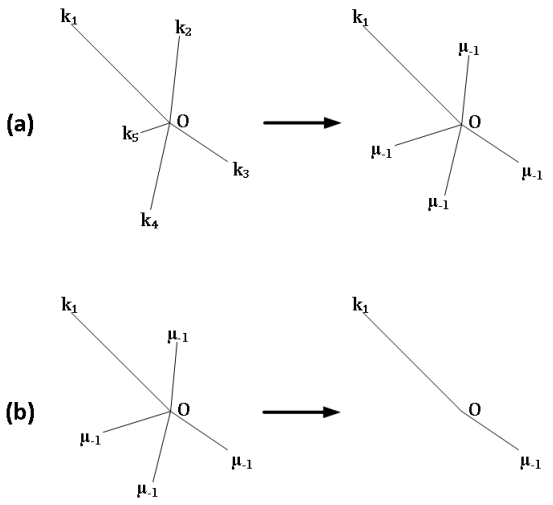


Fig. 4: Modifications to the Extended Star Network.

center to that of the rumor center, on regular trees as well as non-regular trees.

3.1 The Relative-leaf Counting Algorithm

The expression found in (12) is satisfying in multiple ways. On one hand, it provides very strong performance (its deviation from the ML source grows only as the logarithm of the network size, and it is anecdotally even better in many cases, as shown in Table 1). On the other hand, the computational work required is negligible – for a fixed κ, it is a simple expression involving only non-iterative computation. Therefore, it is natural to ask whether this approach, which works so well on extended star networks, can be generalized to address rumor source identification problems on more general graphical structures. Here, we will consider the case of trees, as is commonly done in rumor source identification literature (such as [1], [5] and [7]).

In considering how such an extension might be done, we will make two observations about how the method in Section 2.3 works.

For our first observation, we note that in (12), replacing the shorter arms with arms of their average length μ_{-1} does not change the types center, because $\hat{l}$ is only a function of k_1 and μ_{-1}. Looking further back at (3), we see that modifying $k_2, \cdots, k_m$ in this way only affects the second multiplicand within the summation, and only by a constant scaling factor which is independent of k_1, r and l.

$$P(l;\kappa') = \frac{k_2! * \cdots * k_m!}{(\mu_{-1}!)^{m-1}} \cdot P(l;\kappa) \tag{30}$$
$$\text{where } \kappa = (k_1, k_2, \cdots, k_m),\ \kappa' = (k_1, \mu_{-1}, \cdots, \mu_{-1}).$$

Therefore, for every l, the probability of observing the new infection pattern is a constant scaling of the probability of observing the original infection pattern. This not only preserves the location of the maximum, but also the relative probabilities for all values of l on arm 1. Thus, we can replace an extended star network with a "neuron" (Section 2.5) without changing the source estimation along arm 1 in any way, as shown in Figure 4(a).

For our second observation, we revisit our first observation from Section 2.5. There, we noted that increasing the number of "dendrites" (shorter arms of equal length) does not change the types center along arm 1. This is also consistent with (12), which only depends on the average length of the shorter arms, not on their number.

The opposite argument is also true. Decreasing the number of dendrites does not change the location of the types center along arm 1. If we reduce the number of dendrites to 1, we essentially replace a "neuron" with a "line," as shown in Figure 4(b). Unfortunately, this reduction lacks the precision of the previous one – while the location of the maximum is preserved, the relative values of the likelihood of the other candidate nodes do not scale uniformly as before. Nevertheless, we consider this a useful heuristic for simplifying trees.

Using these two simplifications and the observation from (12) that the distance to the types center along the longest arm depends only on the average length of the shorter arms, we propose applying a sequence of such simplifications to any tree-shaped infection pattern, successively reducing it to a simpler structure (while maintaining and updating the average arm length information accordingly), until we are left with an extended star network.

We can then use the aggregated average length data to define the heuristic types center, just as we did before. From here on, we will also refer to the heuristic types center as the types center. To calculate the types center, consider the undirected graph that describes the underlying *tree*, given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let the infection graph be $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{V}_i \subseteq \mathcal{V}$ are the infected nodes and $(u,v) \in \mathcal{E}_i \subseteq \mathcal{E}$ iff $u, v \in \mathcal{V}_i$ and $(u,v) \in \mathcal{E}$.

Practically speaking, we will proceed by starting at the leaves of the tree $\mathcal{G}_i$ and working our way "in," keeping track of the distance to the *first generation of uninfected descendants*. We call this distance "arm length."

Whenever we reach a branching point, we wait until we have the arm lengths for *all neighbors but one* (i.e. the "children"), and posit that the remaining neighbor must therefore be the "parent."[2] We then implement both of the aforementioned simplifications in principle. We consider the branching point as the beginning of an infected arm with length (distance to the first generation of uninfected descendants) equal to the average of its children's arm lengths plus one. We call such a branching point a "relative-leaf." This relative-leaf is then appended to the list of leaves.

In mathematical terms, for every undirected edge $(u,v) \in \mathcal{E}$ with $u \in \mathcal{V}_i$ and $v \in \mathcal{V}$, $\bar{l}_u(v)$ is the average "arm length" of v as seen from u, as described above.

$$\bar{l}_u(v) = \begin{cases} 1 + \sum\limits_{w \in \mathcal{N}^{\mathcal{G}}(v)\setminus\{u\}} \frac{\bar{l}_v(w)}{|\mathcal{N}^{\mathcal{G}}(v)\setminus\{u\}|}, & \text{if } v \in \mathcal{V}_i,\ |\mathcal{N}^{\mathcal{G}_i}(v)| \neq 1. \\ 1, & \text{if } v \in \mathcal{V}_i,\ |\mathcal{N}^{\mathcal{G}_i}(v)| = 1. \\ 0, & \text{if } v \notin \mathcal{V}_i,\ u \in \mathcal{V}_i. \end{cases} \tag{31}$$

$\bar{l}_u(v)$ measures a weighted average distance of v from the first generation of uninfected descendants, which belong to the largest subtree on $\mathcal{G}$, that contains v but excludes u.

2. We assume that the algorithm is implemented in such a way that there is no ambiguity about "parenthood" due to length values arriving simultaneously from multiple neighboring nodes.

We continue processing leaves and relative-leaves until we reach a putative "root," all of whose neighbors have been accounted for as children. At this juncture, we examine the accumulated average arm lengths of root's children, $\bar{l}_{root}(u),\ \forall u \in \mathcal{N}^{\mathcal{G}}(root)$. Then, if the longest arm is more than one hop longer than the average of the shorter arms, we revise our estimate of the root (moving one hop along the longest arm), then recompute.

To illustrate this point, we look back to extended star networks and (12). We know that the types center for an extended star network is $x := \frac{1}{2}(k_1 - \mu_{-1})$ hops away from the hub node O on the longest arm (arm 1). The types center has $\begin{cases} m \text{ neighbors, if } \hat{l}=O. \\ 2 \text{ neighbors, if } \hat{l}\neq O. \end{cases}$ In both cases, $\hat{l}$ is such that $\bar{l}_{\hat{l}}(u)$ is (nearly) the same for all $u \in \mathcal{N}^{\mathcal{G}}(\hat{l})$. That is, for $u : (\hat{l}, u) \in \mathcal{E}$,

$$\{\bar{l}_{\hat{l}}(u)\}_u = \begin{cases} \{k_1 - x, x + \mu_{-1}\}, \text{ if } \hat{l} \neq O. \\ \{k_1, k_2, \cdots, k_m\}, \text{ if } \hat{l} = O \Leftarrow \text{constant } k_i. \end{cases}$$

More precisely, the estimate of the *root node* is modified until $\bar{l}_{root}(u),\ \forall u \in \mathcal{N}^{\mathcal{G}}(root)$ satisfy (32) from Theorem 2.

The algorithm for implementing this process is described below.

Algorithm 1. The relative-leaf counting algorithm for source detection in tree networks.

Inputs: $\mathcal{G}$ (network graph), $\mathcal{V}_i$ (infected nodes)
Output: center ($\hat{l}$-equivalent for the tree)

/ Initialization */*
$root \leftarrow \{\}$
$leaves \leftarrow \{\}$
for all $v \in \mathcal{V}_i$ ***do***
  $v.children \leftarrow \{\}$
  $v.armLens \leftarrow \{\}$
  $v.avgLen \leftarrow \{\}$
end for

/ Discover Leaves */*
for all $v \in \mathcal{V}_i$ ***do***
  $\mathcal{N}_0^{\mathcal{G}}(v) \leftarrow uninfectedNeighbors(v)$ [3]
  $v.children \leftarrow \mathcal{N}_0^{\mathcal{G}}(v)$
  $v.armLens \leftarrow \bigcup_{i=1}^{|\mathcal{N}_0^{\mathcal{G}}(v)|} \{0\}$
  if $numNeighbors(v) - |v.children| == 0$ ***then***
    $root \leftarrow v$
  end if
  if $numNeighbors(v) - |v.children| == 1$ ***then***
    $leaves \leftarrow leaves \cup \{v\}$
    $v.avgLen \leftarrow 1$
  end if
end for

*/*Discover Parent, Average Arm Lengths, Update Leaves List*/*
while $leaves \neq \{\}$ ***do***
  $v \leftarrow leaves(1)$ */*Random leaf for asynchronous version.*/*

  $par \leftarrow Neighbors(v) \setminus v.children$ */* par = parent */*
  $par.children \leftarrow par.children \cup \{v\}$
  $par.armLens \leftarrow par.armLens \cup \{v.avgLen\}$

  $leaves \leftarrow leaves \setminus \{v\}$

  if $numNeighbors(par) - |par.children| == 1$ ***then***
    $par.avgLen \leftarrow 1 + avg(par.armLens)$
    $leaves \leftarrow leaves \cup \{par\}$
  end if
  if $numNeighbors(par) - |par.children| == 0$ ***then***
    $root \leftarrow par$
    $leaves \leftarrow leaves \setminus \{par\}$
  end if
end while

*/*Move root along longest arm until arm lengths are balanced.*/*
while $max(root.armLens) - 1 > avg(root.armLens \setminus \{max(root.armLens)\})$ ***do***
  / If multiple maxima exist, run each individually. */*
  $maxIdx \leftarrow index(max(root.armLens))$
  $\tilde{v} \leftarrow root.children(maxIdx)$
  $root.children \leftarrow root.children \setminus \{\tilde{v}\}$
  $root.armLens \leftarrow root.\ armLens \setminus \{root.\ armLens(maxIdx)\}$
  $root.avgLen \leftarrow 1 + avg(root.armLens)$
  $\tilde{v}.children \leftarrow \tilde{v}.children \cup \{root\}$
  $\tilde{v}.armLens \leftarrow \tilde{v}.armLens \cup \{root.avgLen\}$
  $\tilde{v}.avgLen \leftarrow 1 + avg(\tilde{v}.armLens)$
  $root \leftarrow \tilde{v}$
end while
$center \leftarrow root$

Theorem 2. *On a tree graph, the relative-leaf counting algorithm* ***converges*** *to a solution, $v^* \in \mathcal{V}_i$, such that*

$$\bar{l}_{v^*}(u^*) - \bar{l}_{u^*}(v^*) \leq 0, \quad \text{where } u^* = \underset{u \in \mathcal{N}^{\mathcal{G}}(v^*)}{\arg\max}\ \bar{l}_{v^*}(u). \tag{32}$$

The rationale behind (32) is as follows:

On the extended star network, $\hat{l}$ on (longest) arm 1 is locates the midpoint between two arms of length k_1 and $\mu_{-1} = \frac{k_2 + \cdots + k_m}{m-1}$. (Refer to Equation 12)

On the general tree $\mathcal{G}$, u^* is the child of v^* with the longest arm of length $\bar{l}_{v^*}(u^*)$. We would want

$$\bar{l}_{v^*}(u^*) - \frac{\sum_{u \in \mathcal{N}^{\mathcal{G}}(v^*) \setminus \{u^*\}} \bar{l}_{v^*}(u)}{|\mathcal{N}^{\mathcal{G}}(v^*) \setminus \{u^*\}|} \leq 1.$$

$$\text{But, } \bar{l}_{u^*}(v^*) = 1 + \frac{\sum_{u \in \mathcal{N}^{\mathcal{G}}(v^*) \setminus \{u^*\}} \bar{l}_{v^*}(u)}{|\mathcal{N}^{\mathcal{G}}(v^*) \setminus \{u^*\}|}.$$

Corollary 1. *For the extended star network, the relative-leaf counting algorithm* ***converges to the types center****, as given by (12).*

The proofs of Theorem 2 and Corollary 1 can be found in the Appendix.

3. Assume that each infected node has a table of the nodes it has not infected with the rumor.

3.2 Analysis of Computational Complexity

In this section, we present the run times of the types and rumor centers. Furthermore, we study the worst-case complexity of calculating the types and rumor centers. We find that the relative-leaf counting algorithm for calculating the heuristic types center on trees offers an improvement in computational efficiency when compared to the message-passing algorithm for finding the rumor center [2].

We begin by considering average case running time, based on empirical testing in simulation. We choose 5 different non-regular trees at random. (For testing purposes, we generate trees with random uniform degree distribution, the degrees 2, 3, 4, 5 and 6 being assigned equiprobably and independently.) For each random tree, we create 100 different infection patterns of various sizes (for each set, we define a hard minimum size, and compute an average size as well), and measure the running time needed to compute each of the metrics. As shown in Table 2, the run times

N_{min}	N_{avg}	Types Center	Rumor Center
100	111.82	0.4372 s	0.5026 s
70	82.92	0.3545 s	0.3991 s
50	62.60	0.2779 s	0.3181 s
40	53.33	0.2453 s	0.2796 s
25	36.15	0.1945 s	0.2069 s

TABLE 2: Run time (averaged over 100 infections of varying sizes N on each of 5 different random non-regular trees.)

increase fairly linearly with N_{avg} for both the types and rumor centers. Except for the smallest infection set, the types center consistently requires about 12-13% less computation time than the rumor center. As expected, we notice that the gap between the run times for the types and rumor centers increases with increasing N_{avg} on an absolute basis.

Now we turn our attention to worst case running time, which we examine through asymptotic analysis. Calculation of the types and rumor centers occurs in $\mathcal{O}(N)$ computations[4], where $N = |\mathcal{V}_i|$. All the quantities in the message-passing algorithm for the rumor center are integers in the range of $[1,\ N!]$. However, the relative-leaf counting algorithm for the types center utilizes real (rational) numbers in the range $[1, N]$. Assume that for the relative-leaf counting algorithm we utilize $\rho_N \leq \log(N)$ decimal places.

a_1) At a node of degree d, the rumor center needs:

a) d summations of worst-case complexity: $\mathcal{O}(\log(N))$.
b) $2d$ multiplications of worst-case complexity[5]: $\mathcal{O}(\log p \log\log p \log\log\log p)$, where $p = \mathcal{O}(N!)$ [6].

a_2) At a node of degree d, the *heuristic* types center needs:

- at least d and at most $2d$ summations of worst-case complexity: $\mathcal{O}(\log(N) + \rho_N)$.
- at least 1 and at most 2 divisions of worst-case complexity: $\mathcal{O}(q \log q \log\log q)$, where $q = \log N + \rho_N$.

b) For the rumor center, the node selected as **root** in the message-passing algorithm of [2] needs to compute the factorial of $N-1$ and its product with $N-1$ numbers.

- Worst-case complexity for the factorial is: $\mathcal{O}(\log p \log\log p \log\log\log p)$, $p = \mathcal{O}(N \log N)$.
- $N-1$ divisions of worst-case complexity: $\mathcal{O}(\log p \log\log p \log\log\log p)$, where $p = \mathcal{O}(N!)$.

Therefore, the worst-case complexity of the relative-leaf counting algorithm at any node is $\mathcal{O}(q \log q \log\log q)$, where $q = \log N + \rho_N$ and the worst-case complexity of the message-passing algorithm at any node is $\mathcal{O}(\log p \log\log p \log\log\log p)$, where $p = \mathcal{O}(N!)$.

3.3 Performance of the Types and Rumor Centers in General Trees

In this section, we simulate infections on d-regular (Section 3.3.1) as well as non-regular (Section 3.3.2) infinite trees in order to compare the performance of rumor center and types center approaches. Then, we provide some overall remarks on how these two different rumor source estimators compare. Throughout the section, we examine performance by generating infections over different types of graphs, and then comparing the proximity of the actual infection source to both the types center and the rumor center estimates.

3.3.1 Regular Trees

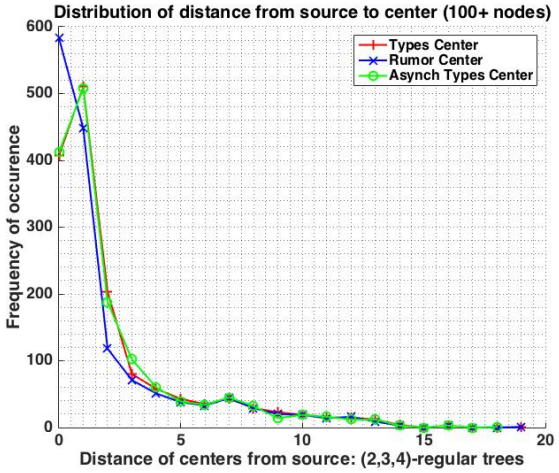
In our first round of performance testing, we simulate infections on regular infinite trees of degrees 2, 3 and 4. Note that in such regular trees, the rumor center is proven to be the optimal ML source estimator [2]. Therefore, the question of interest in these cases is whether types center achieves similar performance as rumor center. To that end, we generated exponentially distributed infection times to create infected subtrees with at least 100 nodes. In practical implementation, the types center can be computed in a synchronous (as described above) or asynchronous fashion (in which case each node might experience a random delay before it relays information to its parent). Allowing for this delay may affect the order in which nodes get processed, but allows more flexibility to execute a parallel implementation of the algorithm. Therefore, in our simulations we examine the performance of both synchronous and asynchronous approaches.

The types and rumor centers are computed for 500 infection instances for each value of d, and the statistics of their distances from the actual source are plotted in Figures 5a (degrees 2, 3, and 4) and 5b (degrees 3 and 4 only, thus focusing on shorter-tailed cases). These results show that the types center distances are indeed very similar to the ML-optimal rumor center distances. Moreover, in Figures 6a and 6b we compare the difference between the distance of the rumor and the types centers from the actual source for the same infection instances. These figures reveal that the two centers achieve very close proximity to the source as desired, with types center being no more than 3 hops farther

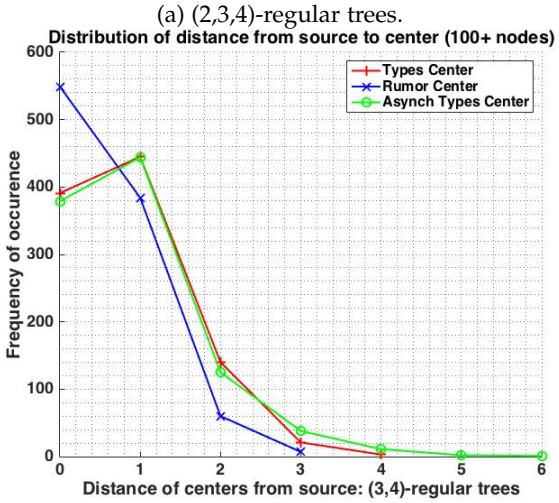
4. We will disregard the computation time for the finding the node(s) with the highest (arm length, rumor centrality) attributes.

5. as per Schönhage-Strassen's algorithm.

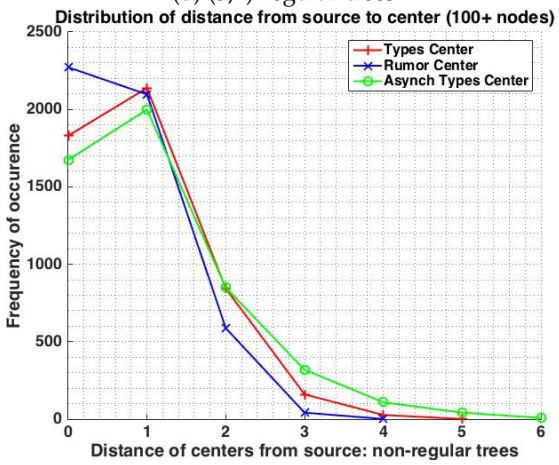
6. Since the multiplicands are product of the number of nodes in each subtree.



(a) (2,3,4)-regular trees.



(b) (3,4)-regular trees.



(c) Non-regular trees.

Fig. 5: Accuracy of source estimators on trees of size ≥ 100.

than the rumor center in all cases, and being at most 1 hop farther than the rumor center in $\sim 95\%$ of the cases.

3.3.2 Non-regular Trees

Having observed the proximity of types and rumor centers in regular trees, we next simulate infections on non-regular infinite trees. To that end, we generate trees with random uniform degree distribution, the degrees 2, 3, 4, 5 and 6 being assigned equiprobably and independently at every node. Note that for such non-regular trees, neither rumor nor types center has any optimality guarantees. Applying an exponential distribution on infection times as before, we create infected subtrees with at least 100 nodes over the non-regular trees. The types and rumor centers are computed for 20 infection instances on each of 250 different non-regular trees, and the statistics of their distances from the actual source are plotted in Figure 5c. When compared to the regular-tree cases, we observe that the performance difference between the types and rumor center appears to reduce. In fact, this observation is further reinforced in Figure 6c, which shows that the types center actually had a *lower* distance from the true source about 25% of the time, the same distance about 37% of the time, and a higher distance about 38% of the time when compared to the rumor center. As such, we observe that the estimation provided by the types center becomes more competitive against the estimation provided by the rumor center for non-regular trees. Moreover, as seen in Figure 5 for both regular and non-regular trees, the asynchronous implementation of the relative-leaf counting algorithm reduces accuracy minimally, while utilizing the same amount of computation time as the synchronous operation of the relative-leaf counting algorithm.

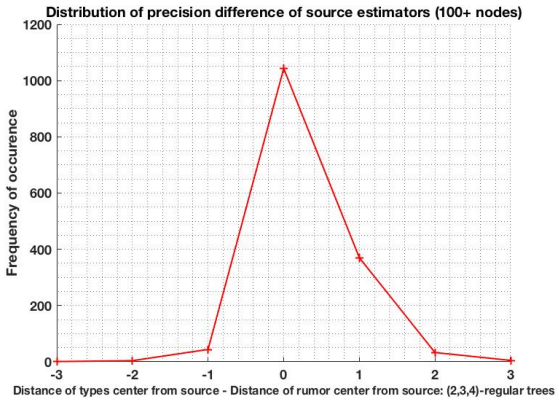
N	Types Center		Rumor Center		Asynchronous Types Center	
	μ	σ^2	μ	σ^2	μ	σ^2
100	0.8838	0.8374	0.6806	0.7110	1.0696	1.0610
70	0.9940	0.9242	0.8758	0.7465	1.1836	1.1307
50	1.1790	0.9879	0.9794	0.8248	1.3928	1.2314
40	1.3128	1.0755	1.0246	0.8663	1.4860	1.2580
25	1.3932	1.0940	1.0956	0.9155	1.5612	1.2688

TABLE 3: Mean and variance of distance from true source to estimates for 5000 infections of size N on non-regular trees.

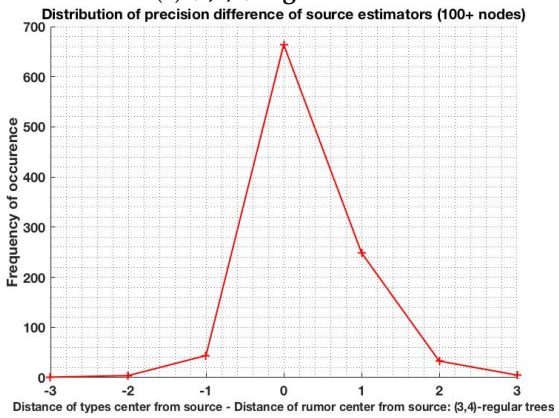
Table 3 investigates the performance of rumor center as well as synchronous and asynchronous types centers as the size of infections grow. They show the reassuring property that the estimation errors of each center decrease in mean and in variance as the infection size grows. However, the growth of the infection size also increases the computational complexity, and thus the run time of the estimators (as discussed in Section 3.2 and in Table 2).

One interpretation of these results is that, in applications where computation time is at a premium, the types center might be an acceptable alternative to the rumor center for non-regular trees, since it provides very similar performance at a lower computational cost. This may be especially true for larger cases, where the savings in absolute terms may be substantial. In such cases, the relatively marginal performance difference may not be of tremendous concern, especially since the error magnitude actually gets *smaller* with increasing network size (for either metric).

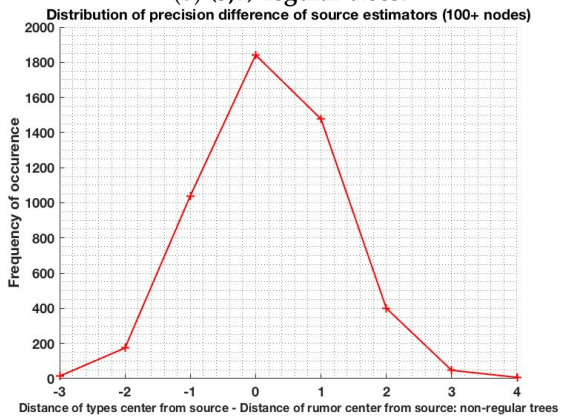
Another possibility is that, in cases where computation time is somewhat less of a concern, but we wish to put a premium on accuracy, we could actually run *both* algorithms, and use the results together in some fashion – either as cross-validation, or using a synthesis strategy that utilizes both



(a) **(2,3,4)-regular trees.**



(b) **(3,4)-regular trees.**



(c) **Non-regular trees.**

Fig. 6: **Case by case comparison of the types and rumor centers on trees of size ≥ 100.**

centers. For example, in an application where the estimated center is to be used as a starting point for in-depth forensic analysis (perhaps involving closer investigation of a particular person, computer on a network, etc., all of which can be far more costly or disruptive than running an algorithm), then the search space could be modified to prioritize nodes that are near one or both centers, or on/near the path between them, etc. Especially in applications where nodes might have high degree, and even a small distance can lead to a large search space, any refinement of such a space could provide significant benefits on the back end, which could more than justify the higher computational costs in such cases (less than double, according to our results).

4 Conclusion

In this work, we aim to efficiently locate the source of an infection in tree-structured networks under the SI infection model. Starting with the case of extended star networks, we derive the types center, a method of types-based approximation to the ML source estimator on the infection graph. We show empirically that this estimator is exact for a selection of relatively small test cases, and prove that it is asymptotically accurate to within an $\mathcal{O}(\log(n))$ neighborhood within the network, providing highly efficient source identification even in large networks (especially compared to the error margins seen in other, similar problems, such as the $\mathcal{O}(\sqrt{n})$ best possible accuracy for single source identification in a line network [1]). We also show that this estimator has different qualitative properties than the rumor centrality measure for infections on extended star networks.

Next, we extend this estimation approach from extended star networks to general trees, using the same principles to design the *relative-leaf counting algorithm*. Preliminary simulation testing shows that as network size increases, the required computation time goes up, but the estimation error goes down. When compared to the rumor centrality measure, the heuristic types center offers a tradeoff – competitive error performance (sometimes higher, sometimes lower, though slightly higher on average), but lower computation time. Therefore, our algorithm makes sense either as an alternative method in applications where computation cost is at a premium, or as a supplemental tool to be used in combination with rumor centrality in a diversity-based approach, in cases where improved performance (or a reduced search space) would justify the (less than 2x) higher computational expense.

References

[1] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Transactions on information theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[2] ——, "Detecting sources of computer viruses in networks: theory and experiment," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 1. ACM, 2010, pp. 203–214.

[3] ——, "Rumor centrality: a universal source detector," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1. ACM, 2012, pp. 199–210.

[4] ——, "Finding rumor sources on random trees," *Operations Research*, vol. 64, no. 3, pp. 736–755, 2016.

[5] W. Luo, W. P. Tay, and M. Leng, "On the universality of jordan centers for estimating infection sources in tree networks," *IEEE Transactions on Information Theory*, 2017.

[6] ——, "How to identify an infection source with limited observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 586–597, 2014.

[7] W. Luo and W. P. Tay, "Estimating infection sources in a network with incomplete observations," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 301–304.

[8] ——, "Finding an infection source under the sis model," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2930–2934.

[9] K. Zhu and L. Ying, "Information source detection in the sir model: A sample-path-based approach," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 1, pp. 408–421, 2016.

[10] W. Hu, W. P. Tay, A. Harilal, and G. Xiao, "Network infection source identification under the siri model," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1712–1716.

[11] S. Spencer and R. Srikant, "On the impossibility of localizing multiple rumor sources in a line graph," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 2, pp. 66–68, 2015.

[12] F. Ji, W. P. Tay, and L. R. Varshney, "Estimating the number of infection sources in a tree," in *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*. IEEE, 2016, pp. 380–384.

[13] F. Ji and W. P. Tay, "Identifying rumor sources with different start times," in *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. IEEE, 2016, pp. 1–5.

[14] W. Luo and W. P. Tay, "Identifying infection sources in large tree networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*. IEEE, 2012, pp. 281–289.

[15] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2850–2865, 2013.

[16] W. Luo and W. P. Tay, "Identifying multiple infection sources in a network," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*. IEEE, 2012, pp. 1483–1489.

[17] K. Zhu, Z. Chen, and L. Ying, "Catch'em all: Locating multiple diffusion sources in networks with partial observations." in *AAAI*, 2017, pp. 1676–1683.

[18] Z. Chen, K. Zhu, and L. Ying, "Detecting multiple information sources in networks under the sir model," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 17–31, 2016.

[19] K. Zhu and L. Ying, "A robust information source estimator with sparse observations," *Computational Social Networks*, vol. 1, no. 1, p. 3, 2014.

[20] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, "Rumor source identification in social networks with time-varying topology," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 166–179, 2018.

[21] Z.-L. Hu, X. Han, Y.-C. Lai, and W.-X. Wang, "Optimal localization of diffusion sources in complex networks," *Royal Society open science*, vol. 4, no. 4, p. 170091, 2017.

[22] F. Altarelli, A. Braunstein, L. Dall'Asta, A. Lage-Castellanos, and R. Zecchina, "Bayesian inference of epidemics on networks via belief propagation," *Physical review letters*, vol. 112, no. 11, p. 118701, 2014.

[23] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, "Inferring the origin of an epidemic with a dynamic message-passing algorithm," *Physical Review E*, vol. 90, no. 1, p. 012801, 2014.

[24] G. Fanti, P. Kairouz, S. Oh, and P. Viswanath, "Spy vs. spy: Rumor source obfuscation," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1. ACM, 2015, pp. 271–284.

[25] G. Fanti, P. Kairouz, S. Oh, K. Ramchandran, and P. Viswanath, "Hiding the rumor source," *IEEE Transactions on Information Theory*, 2017.

[26] ——, "Rumor source obfuscation on irregular trees," in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 2016, pp. 153–164.

[27] W. Luo, W.-P. Tay, and M. Leng, "Infection spreading and source identification: A hide and seek game." *IEEE Trans. Signal Processing*, vol. 64, no. 16, pp. 4228–4243, 2016.

[28] S. Spencer and R. Srikant, "Maximum likelihood rumor source detection in a star network," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2199–2203.

[29] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

Himaja Kesavareddigari received her Bachelor of Technology degree in Electrical Engineering from the Indian Institute of Technology, Madras in 2013. She was a Software Engineer at Samsung Research Institute Bangalore from 2013 to 2014. In 2014, she started working towards her doctoral degree in Electrical and Computer Engineering at The Ohio State University.

Her research interests include control and optimization of interdependent networks.

Sam Spencer (S'95 / M'04) received his B.A. degree in Mathematics and Computational and Applied Math (Mathematical Sciences) from Rice University in 1995, and his M.S. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 1998. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. Between 2006 and 2012, he worked as a Senior Electrical Engineer in the Advanced Technology Center at Rockwell Collins in Cedar Rapids, IA. His other work experience includes the Massachusetts Institute of Technology, ADTRAN, and the US Department of Defense.

He has been awarded a National Science Foundation (NSF) Graduate Fellowship, as well as a Computational Science and Engineering Graduate Fellowship, and was a William Lowell Putnam Mathematics Competition honoree. He holds five US patents in areas such as signal detection, feature estimation, and cognitive networked electronic warfare. His research interests include social and societal networks, signal analysis, information security, algorithms, information theory, communications systems, and waveform design.

Atilla Eryilmaz (S'00 / M'06 / SM'17) received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2001 and 2005, respectively. Between 2005 and 2007, he worked as a Postdoctoral Associate at the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Since 2007, he has been at The Ohio State University, where he is currently a Professor of Electrical and Computer Engineering.

Dr. Eryilmaz's research interests include design and analysis for communication networks, optimal control of stochastic networks, optimization theory, distributed algorithms, pricing in networked systems, and information theory. He received the NSF-CAREER Award in 2010 and two Lumley Research Awards for Research Excellence in 2010 and 2015. He is a co-author of the 2012 IEEE WiOpt Conference Best Student Paper, subsequently received the 2016 IEEE Infocom, 2017 IEEE WiOpt, and 2018 IEEE WiOpt Conference Best Paper Awards. He has served as TPC co-chair of IEEE WiOpt in 2014, and of ACM Mobihoc in 2017, and is an Associate Editor of IEEE/ACM Transactions on Networking since 2015, and of IEEE Transactions on Network Science and Engineering since 2017.

R. Srikant R. Srikant (S '90-M '91-SM '01-F '06) received his B.Tech. from the Indian Institute of Technology, Madras in 1985, his M.S. and Ph.D. from the University of Illinois in 1988 and 1991, respectively, all in Electrical Engineering. He was a Member of Technical Staff at AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois at Urbana-Champaign, where he is the Fredric G. and Elizabeth H. Nearing Professor in the Department of Electrical and Computer Engineering, and a Professor in the Coordinated Science Lab.

He is the recipient of the 2015 INFOCOM Achievement Award and the 2019 IEEE Koji Kobayashi Computers and Communications Award, and has received several Best Paper awards including the 2015 INFOCOM Best Paper Award and the 2017 Applied Probability Society Best Publication Award. He was the Editor-in-Chief of the IEEE/ACM Transactions on Networking from 2013-2017. His research interests include communication networks, machine learning, and applied probability.