A Two-stage Algorithm to Estimate the Source of Information Diffusion in Social Media Networks

Alireza Louni
IEEE Student Member
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ, USA.
Email: alouni@stevens.edu

K. P. Subbalakshmi
IEEE Senior Member
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ, USA.
Email: ksubbala@stevens.edu

***Abstract*—We study the important problem of source localization in the context of information spreads in large social networks. Specifically, we design a Maximum-Likelihood source localization algorithm that is especially suited to large social networks. Our proposed algorithm requires about 3% fewer sensor nodes than other single stage algorithms for the same level of accuracy in detection. For practical social networks, which are typically large, this translates to a significantly fewer number of sensor nodes.**

I. Introduction

Social networks play a crucial role in the diffusion of information and adoption of new technologies. About 54% of Internet users in the United States are on Facebook [1]. 48% of their real world contacts happens on Facebook and many of these users intermittently forward various pieces of information about new technologies/services through their friendship/subscrption networks. Due to this large amount of online interaction, social networks are well suited to viral marketing and rapid information diffusion [2]. Therefore, it is of great interest to understand and model how information disseminates through social networks.

Some researchers have proposed models inspired by the spread of contagion in a population to model the spread of information in networks. Two such models include the susceptible-infected-susceptible (SIS) and the susceptible-infected-recovered (SIR) models [3]–[5]. In these models, information spreads among infected nodes (prior information adopter) and susceptible nodes. For example, in Twitter, the nodes that have received a tweet are considered infected whereas those that have not yet received the tweet are considered susceptible. Most of these works study the conditions under which a piece of information spreads across the entire network when not all nodes are infected and when some nodes are capable of recovering from infection. It is shown that information spreads to the entire network if the probability that an infected node will affect a susceptible node exceeds a threshold value (known as the epidemic threshold). In one of the models, a node adopts a piece of information if the fraction of neighbors who have already adopted it is greater than a threshold value. However, in many situations, people tend to observe the outcomes of prior adopters for a period and then decide on whether they will adopt the information or not. This subtlety is incorporated in other models in other works [3], [6].

Social networks can also be used to spread malicious gossip, untruthful information or computer malware. For example, a recent fake tweet about an explosion in the White House, caused the Dow Jones industrial average to drop 152 points within seconds [7]. Although significant research has been done on how a rumor becomes viral on Twitter, or a malware proliferates on the entire Facebook network, the reverse problem of identifying the source of diffusion has not been addressed very well. In this paper we will present a technique to determine the source of diffusion of information, given the structure of the network. We first review the existing literature on source localization in the next subsection before we present out motivation and model.

A. Related Works

The first generation techniques to find the source of information assumes that the source of information is likely to be a node with a high degree of centrality, where centrality is appropriately defined. For example, in [8], closeness, betweenness, and eigenvector centralities are used to locate the origin of the information. However, high central node as the source of diffusion may not be always true. In general, and without prior knowledge about the source of diffusion, we can only assume that the source is uniformly distributed over the network.

Another line of approach uses information from a snapshot of the infected nodes to identify a single source of diffusion. In this approach a maximum likelihood detector is designed to determine the single source of diffusion using information gathered about the nodes in the network (whether infected or susceptible) [9]–[13]. Similar approaches to detect multiple sources of information have also been proposed [14], [15]. While some works assume that all nodes in the network monitor and report their status, [16] only uses a subset of nodes (called sensors) in the network to determine the source location. A source is located by analyzing the arrival times of information at different sensor locations. It is shown in [16] that by monitoring 20% of the network, an average localization error of less than 4 hops can be achieved.

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

B. Our Motivation

Although the method described in [16] achieves good source localization accuracy, it requires a large percentage of nodes in the network to function as sensors. This will work for small networks, but when the network is very large, this will involve the willing participation of a large number of nodes as sensors. For example, Twitter has 41.7 million users. Twenty percent of this will amount to about 8 million nodes acting as sensors. This is, in general, not practical for large networks like Twitter. It is even more inefficient to observe the entire network as in [9]–[15]. It is therefore important for us to look for alternative solutions that will not require such a high percentage of nodes to act as sensors for the same level of accuracy in source localization. This paper proposes such an algorithm and is based on the fact that most real social networks exhibit a highly clustered topography. We propose a two-stage source localization algorithm where, in the first stage, a candidate cluster that is most likely to contain the source is identified. At the second stage, the source is located within the candidate cluster. It is later shown that when the desired performance in terms of the correct detection probability is given, our algorithm decreases the percentage of sensors significantly.

C. Organizations

This paper is organized as follows. In Section II-A, the information diffusion model is described. The ML estimator is discussed in Section II-B. The two-stage algorithm is designed to localize the single source, in Section III. Finally, numerical results and conclusion are presented in Section IV and V respectively.

II. Problem Formulation

A. Information Diffusion Model

In this subsection, we describe our social network model and assumptions. The information diffusion network is modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges in the graph. For example, in a social network, $\mathcal{V}$ is the set of individuals in the network, and an edge exists between any two individuals if they are engaged in any kind of social relationship such as friendship or followership. The graph $\mathcal{G}$ is a weighted graph, with $w(e)$, the weight associated with edge $e \in \mathcal{E}$ representing the value of social relationship between the nodes connected by the edge e. The value of $w(e_{i,j})$ indicates the strength of the i and j $(i, j \in \mathcal{V})$. It is assumed that the diffusion graph, $\mathcal{G}$, is known[1]. Furthermore, we assume the susceptible-infected (SI) model to study information diffusion in the network. Any susceptible node can become infected independently of other nodes. The source of diffusion (s^*) is modeled as a uniformly distributed random variable over the set $\mathcal{V}$. The unknown source of information, s^*, starts to spreads the information through the network at unknown time t^*. The time delay

[1]It is valid in practice, e.g., when a user tweets a rumor, followers can potentially retweet it further. So, in this case the diffusion graph is the follower graph.

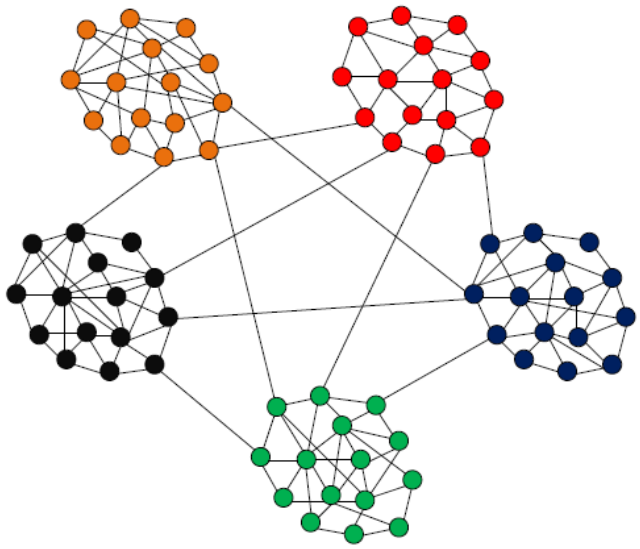

Fig. 1. An example of a modular network graph $\mathcal{G}$ with five clusters.

with which an infected node, say m, can infect a susceptible neighbor, say n, is Gaussian distributed $\mathcal{N}(\mu_{m,n}, \sigma^2_{m,n})$ [17]. To preserve generality of analysis, we assume that the underlying network is non-homogenous. Hence, the average time delay $\mu_{m,n}$ takes on different values for different edges in the network. In general, $\mu_{m,n}$ is a function of the strength of social ties between m and n or $w(e_{i,j})$. For example, when the relationship between m and n is strong, it would take less time for n to retweet a tweet from m to its own followers. We also assume that information disseminates from the source s^* to each node $v \in \mathcal{V}$ along the shortest path connecting them [3]. Let $\mathcal{L} = (l_1, l_2, ..., l_k)$ be a set of sensors used to estimate the source location where $l_i \in \mathcal{V}$. The network graph is composed of close-knit clusters (or communities) of strong within-cluster ties and sparse and weak between-clusters ties (Fig. 1). Social influence between the nodes will obviously affect the spread of information within a network. In other words, nodes with strong ties interact frequently and influence each other more, so, average time delay $\mu_{m,n}$ along strong ties is smaller than weak ties.

B. Diffusion Source Estimator Design

Recent works propose an estimator to identify the most likely information source based on the knowledge of infection status of all nodes in the network. However, it is practically impossible to track the status of all nodes on the social network. Firstly, the computational complexity of the estimator increases greatly with the number of nodes, making it impractical for a typical social media network with millions of nodes. Secondly, we cannot extract the status of just any node on the social network due to privacy concerns. So, we are limited to track the status of a specific set of a limited number of nodes in the network. This small percentage of willingly participating nodes, hereafter called sensors, are chosen to estimate the source location. Sensors observe the arrival times of any piece

of information from all their test candidates. A Maximum Likelihood estimator is then used to determine the most likely source of information. Since the time that the source starts to spread information, t^*, is typically unknown, time difference between sensor pairs, $d_i \triangleq (t_i + t^*) - (t_1 + t^*) = t_i - t_1$, can be used for estimation, where t_i is the time at which information is received at the i^{th} sensor. Therefore, the new estimator becomes

$$\hat{s} = \max_{s \in \mathcal{V}} f(\boldsymbol{D}|s) \tag{1}$$

where $\boldsymbol{D} = (d_1, d_2, ..., d_{k-1})$ is the arrival time difference between sensor pairs. $f(\boldsymbol{D}|s)$ is the pdf of $\boldsymbol{D}$, given s is the source of information. The random vector $\boldsymbol{D}$ is a multivariate Gaussian distribution, since the individual distributions of the time delay are independent Gaussian themselves.

So, the optimal ML estimator can be written as

$$\hat{s} = \max_{s \in \mathcal{V}} \frac{1}{\det(\boldsymbol{\Lambda}_s)^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{D} - \boldsymbol{\mu}_s)^T (\boldsymbol{\Lambda}_s)^{-1} (\boldsymbol{D} - \boldsymbol{\mu}_s)) \tag{2}$$

where $\boldsymbol{\mu}_s = [\mu_1, \mu_2, ..., \mu_i, ..., \mu_{k-1}]$ and μ_i represents the mean value of difference in arrival times between the first and the i^{th} sensors. $\boldsymbol{\Lambda}_s(i,j)$ is the cross-correlation of arrival time difference between the i^{th} and the j^{th} sensors.

The number of sensors, k, trades off computational complexity for accuracy of the source location estimation. It is due to this fact that the amount of information for locating the source increases, as the number of sensors grows.

III. Source Localization Algorithm

In this section, we propose an algorithm to identify the source of diffusion. Our proposed algorithm (See Table I) consists of two stages. First, the most likely candidate cluster is identified using the ML estimator given in Eqn. 2. In the second stage, the ML estimator is applied again using only the nodes inside the cluster flagged by the first stage as the most likely cluster.

First we need to discover the clusters existing in this network. We adopt the community (or cluster) finding method described in [18], which is faster than other existing algorithms, with a complexity of $O(N^2)$, where N is the number of nodes in the network. We then construct a new graph, $\mathcal{G}_{gw}$, of nodes connecting clusters via between-cluster ties (called the gateway nodes). For each cluster, we need to select sensors, preferably, the nodes that receive the largest amount of information from the source. Since, it is assumed that information flows along the shortest path into the network, the most appropriate measure of centrality in this case would be betweenness centrality[2]. Fig. 2 illustrates an example of the two-stage algorithm.

[2]Betweenness centrality refers to the number of times that a node lies on the shortest paths between any pair of nodes in the network.

TABLE I
The Proposed Two-stage Algorithm to Locate the Source of Diffusion

The First Stage:

1: Find each cluster using the Newman community finding method [18].
2: Find the gateway nodes (set $\mathcal{V}_{gw} \in \mathcal{V}$).
3: Select the top m central nodes and employ them as sensors (set $\hat{\mathcal{V}}_{gw} \in \mathcal{V}_{gw}$).
4: Compute the observation vector $\boldsymbol{D}$.
5: For every node $s \in \mathcal{V}_{gw} \backslash \hat{\mathcal{V}}_{gw}$
  Compute the BFS tree rooted at s.
  Compute $\boldsymbol{\mu}_s$ and $\boldsymbol{\Lambda}_s$ with respect to the BFS tree.
  Compute the likelihood function for s using Eqn. 2.
 End
6: Find s^*_{gw} that maximizes the likelihood function (Eqn. 2).
7: Select the cluster that is associated with s^*_{gw} as the candidate cluster.

The Second Stage:

1: Select the top n central nodes from the set of nodes within the candidate cluster (set $\mathcal{V}_s \in \mathcal{V}$) and employ them as the sensors (set $\hat{\mathcal{V}}_s \in \mathcal{V}_s$).
2: Compute the observation vector $\boldsymbol{D}$.
3: For every node $s \in \mathcal{V}_s \backslash \hat{\mathcal{V}}_s$
  Compute the BFS tree rooted at s.
  Compute $\boldsymbol{\mu}_s$ and $\boldsymbol{\Lambda}_s$ with respect to the BFS tree.
  Compute the likelihood function for s using Eqn. 2.
 End
4: Locate the source s^* that maximizes the likelihood function (Eqn. 2).

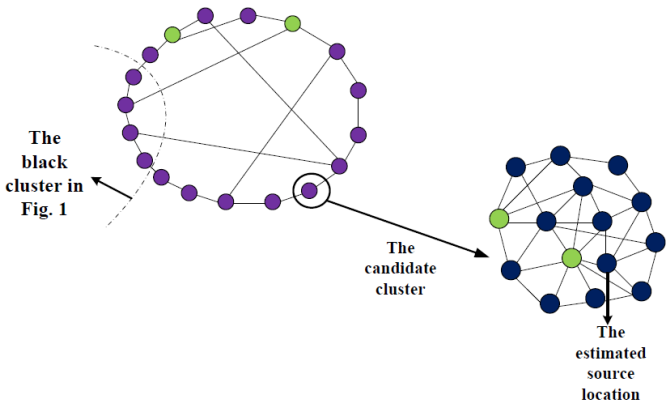

Fig. 2. **The left side**: the graph $\mathcal{G}_{gw}$ of the gateway nodes, in which the green nodes represent the sensors ($m = 2$) at the first stage. At the end of this stage, e.g., the node circled is selected as gateway node that guides us to the candidate cluster (s^*_{gw}). **The right side**: the cluster that most likely contains the diffusion source. The green nodes represent the sensors($n = 2$) which measure the arrival times of messages and then estimate the source location(s^*).

IV. Simulation Results

In this section, we present the numerical results for the performance analysis of the proposed algorithm. We apply the proposed algorithm to a clustered network with variable topologies. We simulate information spread on several different network topologies using the SI model. Since there is no prior knowledge of the source of diffusion, we generate a uniformly distributed source in $[1, N]$ where N is the number of nodes in the network. It is assumed that the inter-arrival times are independent and identically Gaussian distributed,

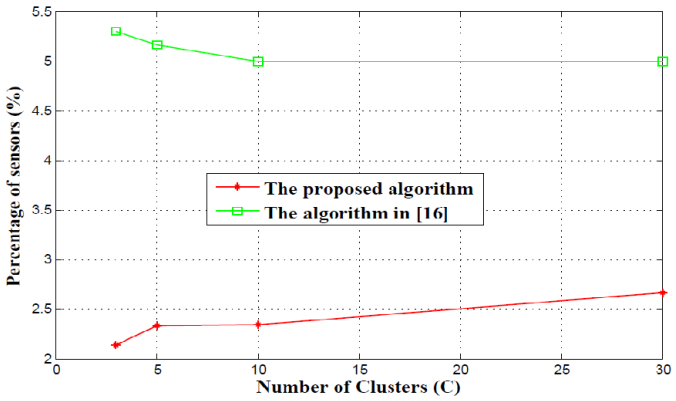

Fig. 3. The percentage of sensors versus the number of clusters ($p = 0.10$ and $n = 3000$) to achieve a detection rate in the interval range [82%,87%].

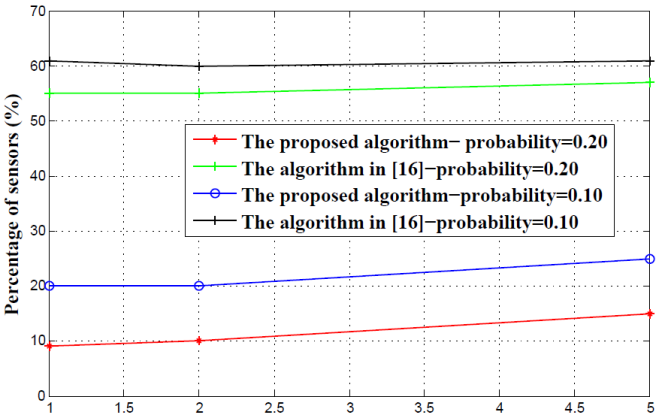

Fig. 4. The percentage of sensors versus the number of clusters ($p = 0.10$ and 0.20 and $n = 100$) to achieve a detection rate in the interval range [82%,87%].

with $\frac{\mu_{m,n}}{\delta_{m,n}}$ for any pair of neighboring nodes m and n being set at 4.

We compare the performance of the proposed algorithm with the alternative algorithm [16], which selects sensors using their degree of centrality and identifies the source location in a single stage. We quantify the performance of each algorithm by measuring the percentage of sensors to achieve a predetermined detection rate[3].

In the first experiment, the number of nodes in the network is set to 3000, and we set the probability that an edge is present between two nodes (p) to 0.10. The fraction of neighbors within the cluster (r) is set to 0.99. The two-stage algorithm is applied to this network to estimate the location of the source. The aim is to achieve a detection rate in the interval range[82%,87%] (in order to be able to compare it with results in [16]). As seen from Fig. 3, the percentage of sensors increases slightly as the number of clusters increases from 3 to 30 for our algorithm, which is exactly the opposite of the trend observed for the single stage algorithm in [16]. This may be due to this fact that for networks with more clusters even a small error at the first can cause a larger error in the location of the source. From Fig. 3, we can see that using the proposed algorithm, around 2% of the total nodes are required to achieve the detection accuracy of 80% or more. This is 3% less than the number of sensors needed for the algorithm in [16]. For large networks, this will translate to a big difference in the number of nodes that need to act as sensors.

We then vary p from 0.10 to 0.20 to illustrate how this parameter affects the performance of our algorithm. A small network is considered with 100 nodes. We also set the fraction of neighbors within the cluster to 0.95. Fig. 4 demonstrates how our method outperforms the alternative algorithm in terms of the percentage of sensors. From Fig. 4, we see that when the link probabilities are higher, we only need a smaller percentage of sensors. Fig. 3 and Fig. 4 clearly demonstrate that the network topology significantly affects the performance of the proposed method. Essentially low complexity, reliable detection of source can be achieved if the network is large and the clusters are relatively dense, which are typical characteristics of a real social network.

Fig. 5 shows the impact of network heterogeneity on the performance of the proposed algorithm under a variety of network topologies. The network is more heterogeneous if the average time delay $\mu_{m,n}$ takes on more diverse values for different edges in the network. For simulation purposes, it is assumed that $\mu_{m,n}$ takes on uniformly distributed values between 1 and b; the larger the value of b, the more heterogeneous the network. More specifically, $b = 1$ represents a homogenous network in which all $\mu_{m,n}$ take on the same value (see Fig.5).The performance of the proposed algorithm is quantified by measuring the correct localization rate. As shown in Fig. 5, the performance of the proposed algorithm improves as the network heterogeneity increases. The increase depends on network topology. For instance, when the network is not well connected (a network with smaller p), the performance sharply increases when b goes above 1. Furthermore, as the network heterogeneity exceeds a threshold value (25 for $n = 100$ and 22 for $n = 1000$), a sharper increase in the performance for all four network topologies can be observed.

In order to evaluate the performance of the two-stage algorithm in real social networks, we apply our proposed algorithm to a sampled Twitter network. The network is obtained by a snowball sampling from a seed set of computer scientiest/statistician and tracing followers links up to three hops [19]. This network was monitored from February 5, 2014 to February 17, 2014. The data-set contains $3,479$ users, $518,021$ tweets and $43,189$ retweets. The weights of graph edges are measured as: $w(e_{A,B}) = \frac{r(A,B)}{\sum_{X \in N_A} r(A,X)}$, where $r(A,B)$ is the number of times B retweeted A and N_A is the set of neighbors of A. Fig. 6 shows that the performance improves for both of the algorithms slightly, as the number of sensors grows.

[3]The detection rate is defined to be the fraction of experiments in which the estimator coincides with the actual source.

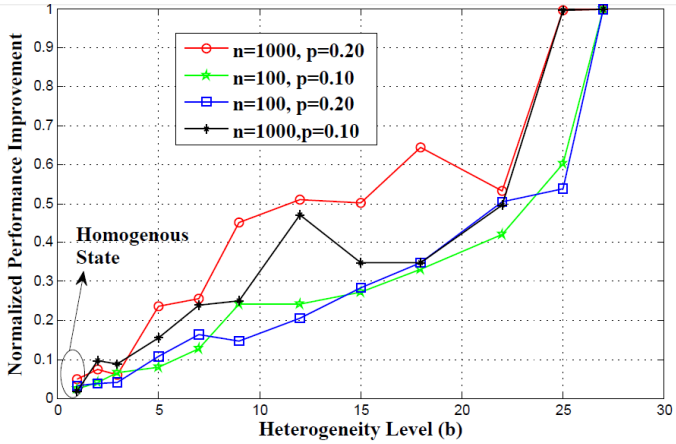

Fig. 5. Impact of network heterogeneity on the performance of the proposed algorithm.

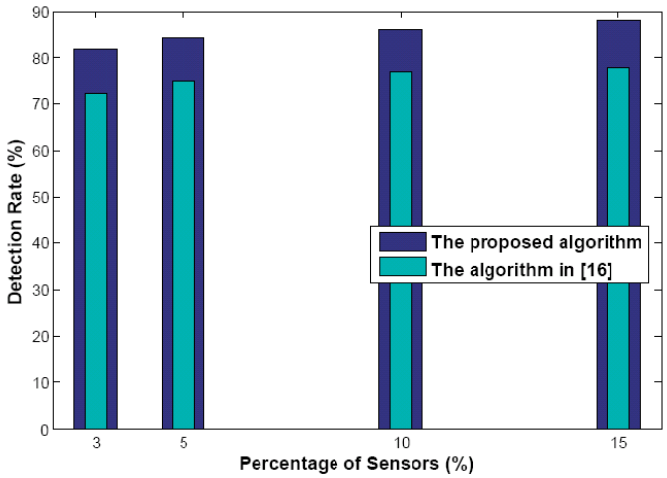

Fig. 6. The detection rate versus the percentage of sensors.

V. Conclusion

In this paper, the problem of locating the source of information in large-scale social networks is considered. We use the SI model to study information diffusion in the network. A two-stage algorithm is proposed in which at the first, the mostly likely candidate cluster is identified. In the second stage, the source is located within the candidate cluster. We evaluate the performance of the proposed algorithm on several different network topologies. Performance is measured in terms of the percentage of nodes in the network that need to act like sensors. We observe significant performance by using our two stage localization algorithm. We also see that as the heterogeneity of the network increases, the localization accuracy increases greatly.

References

[1] "Social network ad spending to approach $1.7 billion this year," *eMarketer*.

[2] F. P. S. Hill and C. Volinsky, "Network-based marketing: Identifying likely adopters via consumer networks," *Statistical Science*, vol. 21, no. 2, p. 256276, 2006.

[3] M. O. Jackson, *Social and Economic Networks*. Princeton University Press, 2008.

[4] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, pp. 167–256, 2003.

[5] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society of London. Series A*, vol. 115, no. 772, pp. 700–721, 1927.

[6] M. Jackson and L. Yariv, "The diffusion of behavior and equilibrium properties in network games," in *American Economic Review*, vol. 97, pp. 92–98, 2007.

[7] G. Strauss, A. Shell, R. Yu, and B. Acohido, "SEC, FBI probe fake tweet that rocked stocks," Apr. 2013.

[8] C. H. Comin and L. da Fontoura Costa, "Identifying the starting point of a spreading process in complex networks," *Phys. Rev. E*, vol. 84, p. 056105, Nov. 2011.

[9] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: theory and experiment," in *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '10, pp. 203–214, 2010.

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

[11] D. Shah and T. Zaman, "Rumor centrality: a universal source detector," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pp. 199–210, 2012.

[12] W. Dong, W. Zhang, and C. W. Tan, "Rooting out the rumor culprit from suspects," *arXiv:1301.6312 [cs.SI]*, 2013.

[13] K. Zhu and L. Ying, "Information source detection in the sir model: A sample path based approach," in *Proc. of Inform. Theory and Applications Workshop*, ITA '13, 2013.

[14] 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*, pp. 1483–1489, 2012.

[15] B. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 11–20, 2012.

[16] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Phys. Rev. Lett.*, vol. 109, p. 068702, Aug. 2012.

[17] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pp. 721–730, 2009.

[18] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, p. 066133, Jun 2004.

[19] S. H. Lee, P.-J. Kim, and H. Jeong, "Statistical properties of sampled networks," *Phys. Rev. E*, vol. 73, p. 016102, 2006.