Who Spread That Rumor: Finding the Source of Information in Large Online Social Networks With Probabilistically Varying Internode Relationship Strengths

Alireza Louni, *Student Member, IEEE*, and K. P. Subbalakshmi, *Senior Member, IEEE*

***Abstract*— We address the problem of estimating the source of a rumor in large-scale social networks. Previous works studying this problem have mainly focused on graph models with deterministic and homogenous internode relationship strengths. However, internode relationship strengths in real social networks are random. We model this uncertainty by using random, nonhomogenous edge weights on the underlying social network graph. We propose a novel two-stage algorithm that uses the modularity of the social network to locate the source of the rumor with fewer sensor nodes than other existing algorithms. We also propose a novel method to select these sensor nodes. We evaluate our algorithm using a large data set from Twitter and Sina Weibo. Real-world time series data are used to model the uncertainty in social relationship strengths. Simulations show that the proposed algorithm can determine the actual source within two hops, 69%–80% of the time, when the diameter of the networks varies between 7 and 13. Our numerical results also show that it is easier to estimate the source of a rumor when the source has higher betweenness centrality. Finally, we demonstrate that our two-stage algorithm outperforms the alternative algorithm in terms of the accuracy of localizing the source.**

***Index Terms*— Partial observation, probabilistic social relationship strength, rumor source estimation, rumor spreading, social networks.**

I. Introduction

ONLINE social networks are experiencing a meteoric rise in popularity. As of the second quarter 2014, the number of active Twitter users per month surpassed 271 million, a growth rate of 20% from the second quarter 2013. Facebook is also rapidly growing with a recent statistic stating that the number of active Facebook users per month has grown from 1.1 billion in the second quarter 2013 to 1.3 billion in the second quarter 2014 [1]. Studies demonstrate that information spreads much faster in social networks than any other communication media [2].

This ease of information dissemination can be a double-edged sword as social networks can also be used to spread rumors [3], [4] or computer malware. For instance, in 2013, a fake tweet originating from a hacked Associated Press Twitter account about bombings in the White House caused the Dow Jones Industrial Average to drop 145 points within 2 min [3]. The spread of information/misinformation in networks can be modeled using epidemic spreading models, such as the susceptible-infected (SI), the susceptible-infected-recovered (SIR) models, or the susceptible-infected-susceptible (SIS) [5]–[10]. Work has also been done on differentiating a rumor from regular information [11]–[14]. Since rumors and misinformation can spread rapidly through social networks, it is necessary to detect the sources of such misinformation for rapid damage control. It also allows us to understand the role of network structure in rumor dissemination, thereby facilitating the design of sophisticated policies to prevent further viral spreading of misinformation through social networks in the future. This paper presents a technique to locate the source of rumors for large networks whose structure is known, whereas it is impossible to observe the entire social network. Note that, in the rest of this paper, we use the terms "rumor" and "information" interchangeably since the techniques described in this paper are applicable to detect the source of any piece of information with a large reach within the network.

A. Related Work

Some of the first attempts to locate the source of information were based on the hypothesis that the source is likely to be a node with a high degree of centrality. Hence, various graph–theoretic measures, such as betweenness, closeness, and eigenvector centrality, were used to locate the source of diffusion [15]. However, in general, every node in the network has the potential to spread rumors. Therefore, conventional graph–theoretic centrality measures are not good estimators of the source.

The second generation of source localization methods uses information from a time snapshot of the infected nodes to construct a maximum likelihood (ML) estimator to identify a single source of rumor [16], [17]. Shah and Zaman [16] proposed rumor centrality, a number assigned to each infected node and indicates the likelihood of being the source of rumor.

Manuscript received March 26, 2016; revised March 1, 2017, November 29, 2017, and January 2, 2018; accepted January 21, 2018. Date of publication February 21, 2018; date of current version May 25, 2018. *(Corresponding author: Alireza Louni.)*

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: alouni@stevens.edu; ksubbala@stevens.edu).

Digital Object Identifier 10.1109/TCSS.2018.2801310

They showed that the node with the maximal rumor centrality is the ML estimate of the source when the underlying social graph is a regular tree, using the SI model for diffusion. The universality of rumor centrality in generic random trees is illustrated in [17]. Assuming the rumor travels via the shortest path, the breadth-first search (BFS) tree is utilized to develop rumor centrality for general graphs. This problem is later generalized to other cases, such as estimating a set of sources [18], [19] and modeling information diffusion using an SIR model [20]. Rumor centrality is extended for the SI model with multiple observations, collected on multiple different rumors [21]. The concept of rumor centrality, under the SI model, was developed to locate multiple sources when the number of sources is unknown [18]. The BFS tree is then assumed for rumor propagation to extend the method to general graphs. Prakash *et al.* [19] employed the eigenvectors of the Laplacian matrix to identify a set of sources in generic graphs. The problem of source localization in the SIS model is considered in [22]. In some situations, there is enough *a priori* knowledge to confine suspicion of a source to a subset of the total nodes in the network. Under these conditions, the optimal source locator becomes a maximum *a posteriori* estimator [23].

One of the problems with the approaches discussed so far [16]–[23] is that they all rely on observing the entire network. This is impractical for large social networks, such as Facebook or Twitter, because of computational complexity. One method to deal with this problem is to observe only a subset of designated nodes (called sensors) [24]. The sensors measure the time and ID of the neighbors from which they receive the rumor and use these to estimate the source. They proposed a ML estimator for tree graphs using measurements by the sensors and an extension to general graphs assuming the BFS tree for information spread. It is shown [24] that an average source localization error of less than four hops can be achieved by monitoring 20% of the network. In [25], we proposed a two-stage source localization algorithm using the fact that real social networks are highly clustered [26]. We showed that for a desired correct detection probability, the proposed algorithm needs 3% fewer sensors. For the specific case of tree structured networks, Celis *et al.* [27] solved the problem of optimal placement of sensors that minimizes the average source estimation error.

All the past works in diffusion source localization have focused on analyzing static and binary graphs, where edge weights with binary and deterministic values represent the presence or absence of a social relationship between nodes. However, since social relationships are more complex in real life and can influence the spread of information between two nodes, all edges (relationships) cannot be treated the same regardless of their strength [28]. For example, a node in the Twitter network may follow hundreds of nodes, but retweet only posts received from nodes with which it shares a strong tie [29]. Furthermore, in real social networks, these relationship strengths can vary over time. Examples of causes for such variations could include: 1) the changes in social network activities at different times of the day; 2) varying trust between the two nodes; and 3) varying interest in the topic [30]. As an example, the histogram of the information propagation delay along two edges on Sina Weibo social network [31], [32] is shown in Fig. 1, clearly demonstrating the heterogeneity of the social relationships. *We model this heterogeneity by using probabilistically varying weights for the edges.*

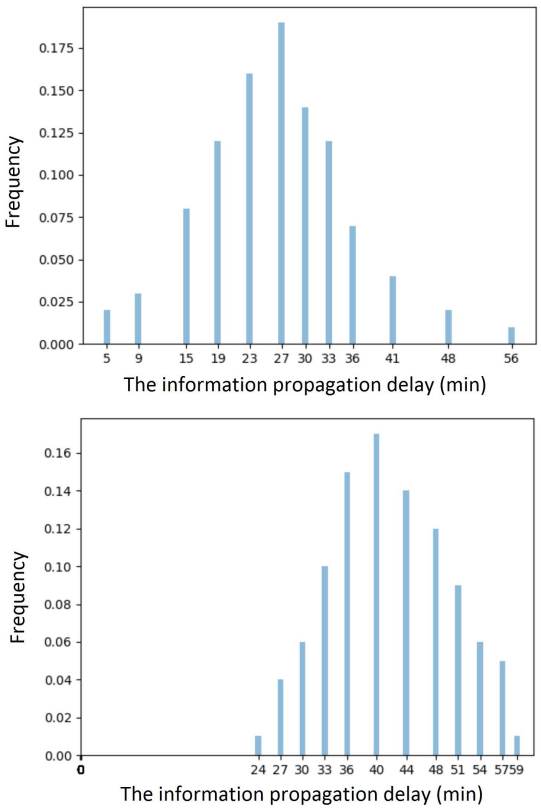


Fig. 1. Histogram of the information propagation delay for two random edges in Sina Weibo social network. Note that the pmf of the delay can be very different for different edges in the network, thus pointing to heterogeneous relationship strengths.

B. Our Contribution

In this paper, we consider the estimation of the source of the rumor in a more realistic model, where the edge weights can vary randomly. Our main contributions are the following. As mentioned before, binary deterministic ties fail to capture relationship strength. We quantify uncertainty of social networks through a probabilistic weighted graph in which the uncertainty in the weight for each edge in the network is modeled by a probability mass function (pmf). We develop a new two-stage source localization algorithm for probabilistic weighted graphs. We investigate the performance of the proposed algorithm and demonstrate its efficiency in experiments with very large real-world social media graphs. We experimentally observe that localization gets harder as the degree of the source of rumor reduces. It is shown that, on average, our estimations are within a few hops of the true source of rumor in the network graph. We study the scenario for the case in which a source initiates a rumor and the sensors use only one single snapshot of the network to estimate the source.

The rest of this paper is organized as follows. In Section II, we present the system model and problem formulation. In Section III, we design algorithms to find the source. In Section IV, we present simulation experiments on real data to verify the effectiveness of our proposed algorithm. Finally, we conclude and summarize in Section V.

II. Problem Formulation

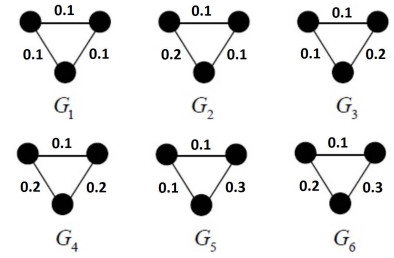
A. Graph Model Representation of Rumor Diffusion

We model the social network as a graph, with nodes representing the entities that could spread the rumor. For example, in a Twitter network, the nodes would be individual Twitter handles (or IDs). We can then define edges as the connections between these nodes (for, e.g., friendship status between people in a Facebook network). We can assign weights to these edges (between 0 and 1) to represent the strengths of the relationship between these nodes, with weight 0 representing the least resistance to propagation of information (hence strongest ties) and weight 1 representing the most resistance to propagation (hence weakest ties) along that edge.

As mentioned in Section I-B, we build a model for the diffusion network, taking into account the varying strengths (edge weights) between the nodes. Since it is computationally complex to monitor the weights continuously, we observe the network and weights at specific time intervals (sample the weights). This gives us a sampling of the possible weights for each edge. Let w_{ij} be the weight of the ith edge at some instance j. Let the number of distinct values for the weight of the edge i be M_i. Note that the number of distinct weight values can be different for different edges (see Fig. 2). We can now construct an $|E| \times M$ matrix of weights, $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M]$, where $M = \prod_{i=1}^{|E|} M_i$. The jth column of this matrix is a vector, $\mathbf{w}_j$, with elements representing one possible combination of weights for each edge. We can then construct one graph, $G_j = (V, E, \mathbf{w}_j)$ for every vector $\mathbf{w}_j$, where V and E denote the set of nodes and edges, respectively. We use the term "possible graph" to refer to such a graph. Note that there are M possible graphs. Assuming weight independence among edges, the probability of occurrence of graph G_j is given by

$$\Pr(G_j) = \prod_{i=1}^{|E|} \Pr(w_{ij}). \tag{1}$$

An example graph with 3 vertices and 3 edges and different weight sets is shown in Fig. 2.



$$\mathbf{W} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 & 0.2 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.2 & 0.3 & 0.3 \end{bmatrix}$$

Fig. 2. $\mathbf{W}$ is the matrix of possible weights for the three edges in the graph. Note that edges take on weight values from different sets. In this example, edge e_1 always has weight 0.1; e_2 can take on values from $\{0.1, 0.2\}$ and e_3 can take on values $\{0.1, 0.2, 0.3\}$. This gives us six possible graphs shown in the figure.

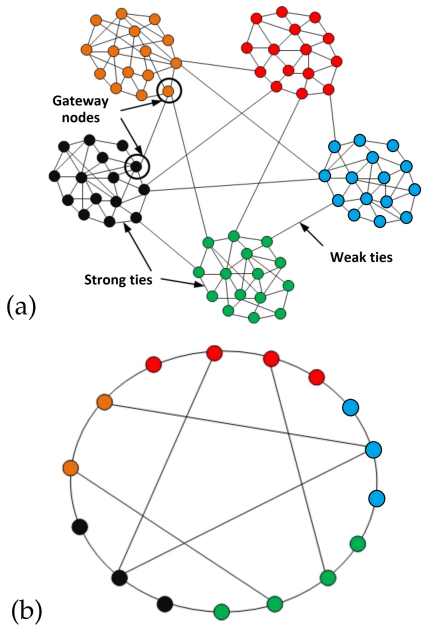


Fig. 3. (a) Five-cluster social network connecting three clusters via three weak ties. (b) Graph of the gateway nodes is used to find the most likely candidate cluster.

In real-life social networks, the structure of the graph G_j is modular and can be thought of as comprising of clusters of nodes that frequently interact with each other. These clusters are in turn connected to each other via weak ties [Fig. 3(a)]. The strong ties are responsible for dissemination within the clusters, whereas the weak ties allow the information to go "viral" across clusters [33], [34]. Let the unknown source of rumor, $v^* \in V$, initiate the rumor at an unknown time t^*. We assume that the rumor is propagated in the network according to a continuous-time SI model. Any susceptible node can become infected independently of other nodes. It is reasonable to assume that the rumor traveled through the shortest path from the source v^* to the sensor nodes. This concept has been formalized by several researchers [16]–[24] by stating that the information diffusion tree is a BFS tree. The time taken for a node m to repost information from n to its own neighbors on the network G_j depends on the strength of the social tie between m and n. It takes less time for information to diffuse inside a dense cluster of strong ties than across weak ties. Furthermore, nodes repost what their neighbors posted with different time delay values (e.g., depending on the time of day, their other activities, etc.). Similar to [24], we assume a Gaussian distribution for the information propagation delay, d_i, along edge $e_i \in E$ (we later relax this assumption). That is,

given weight w_{ij}, d_i is distributed as

$$d_i|w_{ij} \sim \mathcal{N}\big(w_{ij} \cdot \mu_{\max}, \sigma_{ij}^2\big) \tag{2}$$

where the average information propagation delay for the weakest social relationship ($w_{ij} = 1$) is $\mu_{\max}$.

As mentioned before, utilization of sensors is a good way to reduce the complexity of the search for the source of information. However, to decrease this complexity further, we use the modularity of social networks. Specifically, we propose a two-step approach to the problem [25]. The first stage involves identifying the cluster that most likely contains the source of diffusion, and in the second stage, we search inside the most likely cluster to locate the specific source node. The search space in this two-stage method is limited to the nodes connecting clusters at the intercluster level and the nodes inside the most likely candidate cluster at the intra-cluster level, which results in much lower computational complexity as compared to the methods in [24] and [35]. In the first stage, we construct a new graph consisting of the gateway nodes and their interconnections, $G_j^{\text{gate}} = (V^{\text{gate}}, E^{\text{gate}}, \mathbf{w}_j^{\text{gate}})$. The gateway nodes are those nodes in a cluster that connect to nodes in other clusters via weak ties. We denote the matrix of edge weights for this new gateway graph by $\mathbf{W}^{\text{gate}} = [\mathbf{w}_1^{\text{gate}}, \mathbf{w}_2^{\text{gate}}, \ldots, \mathbf{w}_M^{\text{gate}}]$, where $M = \prod_{i=1}^{|E^{\text{gate}}|} M_i$. A set of k_1 nodes, $S = \{s_1, s_2, \ldots, s_{k_1}\}$, is selected from V^{gate} to act as sensors. The sensors measure the first time arrival of the rumor to estimate the most likely candidate cluster. Since the time that the source starts to spread the rumor, t^*, is typically unknown, the difference in arrival times at sensor pairs, $\Delta t_{i1} \triangleq (t_i + t^*) - (t_1 + t^*) = t_i - t_1$, can be used for estimation, where t_i and t_1 are the times at which the rumor is received at the ith sensor and the first sensor in the gateway graph, respectively. Let the arrival time difference vector be $\mathbf{\Delta t} = (\Delta t_{21}, \Delta t_{31}, \ldots, \Delta t_{k_1 1})$. The observation vector $\mathbf{\Delta t}$ depends on $\mathbf{w}_j^{\text{gate}}$. The edge weight between two gateway nodes is calculated by summing the edge weights along their corresponding shortest path. Given the weight vector $\mathbf{w}_j^{\text{gate}}$, the arrival time difference vector is multivariate Gaussian, as the individual distributions of the time delay are independent Gaussian themselves.[1]

Because of the lack of *a priori* knowledge about the source, ML estimator is the optimal estimator for the source of the rumor, i.e., it minimizes the estimation error [37], [38]. This can be written as

$$\hat{v}^{(1)} = \underset{v \in V^{\text{gate}}}{\arg\max}\ \mathcal{P}(\mathbf{\Delta t}|v) \tag{3}$$

where $\mathcal{P}(\mathbf{\Delta t}|v)$ is the pmf of the observation vector, given $v \in V^{\text{gate}}$ and the SI model is used. Considering the statistical distribution of $\mathbf{\Delta t}$, the optimal ML estimator is

$$\hat{v}^{(1)} = \underset{v \in V^{\text{gate}}}{\arg\max} \sum_{j=1}^{M} \Pr\left(G_j^{\text{gate}}\right) \frac{1}{(2\pi)^{\frac{k_1-1}{2}} \det\left(\mathbf{\Lambda}_{v,j}\right)^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})(\mathbf{\Lambda}_{v,j})^{-1}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})^T\right) \tag{4}$$

where $\boldsymbol{\mu}_{v,j}(r)$ is the mean value of difference in arrival times between the first and the $(r+1)$th sensors. $\mathbf{\Lambda}_{v,j}(a,b)$ is the cross correlation of difference in arrival times between the ath and the bth sensors. $\Pr(G_j^{\text{gate}})$ is the probability of the jth possible gateway graph G_j^{gate}. Both $\boldsymbol{\mu}_{v,j}(r)$ and $\mathbf{\Lambda}_{v,j}$ are for the case in which the BFS tree is rooted at the node v in the jth instance. Assuming independence among edges, the probability of the jth possible gateway graph is calculated as $\Pr(G_j^{\text{gate}}) = \prod_{i=1}^{|E^{\text{gate}}|} \Pr(w_{ij})$ where w_{ij} $(1 \le i \le |E^{\text{gate}}|)$ are the elements of the jth column of the matrix $\mathbf{W}^{\text{gate}}$. In general, the number of all possible graphs can be extremely large for a real social network. Hence, to reduce the search complexity, we search only among the $\hat{m}$ most likely gateway graphs corresponding to the $\hat{m}$ most likely weight vectors $\mathbf{w}_j^{\text{gate}}$, where $\hat{m} \ll M$. To find the sufficient number of possible graphs $\hat{m}$, we adopt the same method as in [39], where $\hat{m}$ is computed by experimentally determining the smallest number of possible graphs that does not cause a significant difference in the average shortest path.[2] By substituting $\hat{m}$ with M, (4) can be expressed as

$$\hat{v}^{(1)} = \underset{v \in V^{\text{gate}}}{\arg\max} \sum_{j=1}^{\hat{m}} \Pr\left(G_j^{\text{gate}}\right) \frac{1}{\det\left(\mathbf{\Lambda}_{v,j}\right)^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})(\mathbf{\Lambda}_{v,j})^{-1}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})^T\right). \tag{5}$$

In the second stage, the search space will be limited to the nodes inside the cluster that is associated with $\hat{v}^{(1)}$. Let $G_j^{\text{cluster}} = (V^{\text{cluster}}, E^{\text{cluster}}, \mathbf{w}_j^{\text{cluster}})$ be the graph of the nodes inside the most likely candidate cluster. k_2 sensors are employed at this stage to collect information about the rumor. Just as in the case of the gateway graphs, we search among the $\hat{m}$ most likely graphs corresponding to the $\hat{m}$ most likely weight vectors $\mathbf{w}_j^{\text{cluster}}$ to locate the source of diffusion in the second stage. The corresponding optimal ML estimator is

$$\hat{v}^{(2)} = \underset{v \in V^{\text{cluster}}}{\arg\max} \sum_{j=1}^{\hat{m}} \Pr\left(G_j^{\text{cluster}}\right) \frac{1}{\det\left(\mathbf{\Lambda}_{v,j}\right)^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})(\mathbf{\Lambda}_{v,j})^{-1}(\mathbf{\Delta t} - \boldsymbol{\mu}_{v,j})^T\right) \tag{6}$$

where $\Pr(G_j^{\text{cluster}})$ is the probability of the jth possible gateway graph and $\mathbf{\Delta t}$ is the observation vector at the sensors. Note that the optimization problems in (5) and (6) have no closed-form solution; thus, we run a brute-force search through all the suspected nodes. The number of suspected nodes is equal

[1]Note that $\mathbf{\Delta t}$ can be closely approximated by a multivariate Gaussian if the edge delays have a distribution other than a Gaussian, due to the regular central limit theorem [36]. However, this is not applicable for the cases in which the edge delays have a heavy-tail distribution.

[2]The average shortest path distance is given by $\overline{\text{SPD}} = \sum_{G_i \in G_s, |G_s|=m} P(G_i)\overline{\text{SPD}(G_i)}$ where $\overline{\text{SPD}(G_i)} = (1/(|V| \cdot (|V| - 1))) \sum_{i \in V; i \neq j} \sum_{j \in V} \text{SPD}_{i,j}$ and $\text{SPD}_{i,j}$ is the shortest path distance between nodes i and j in G_i.

to the size of the most likely candidate cluster, which provides us with the following advantages. First, the percentage of sensors significantly reduces compared with the alternative algorithms [24], [35] for the same level of accuracy. This decreases the dimension of the matrix $\mathbf{\Lambda}_{v,j}$ in.(6), thereby reducing the computational complexity of $(\mathbf{\Lambda}_{v,j})^{-1}$. Second, the likelihood function in (6) should be calculated for much smaller number of nodes than all the nodes in the network.

Algorithm 1 Two-Stage Algorithm

1: **procedure** TWO-STAGE(V, E, $\mathbf{W}$, $\hat{m}$, k_1, k_2)
2: V^{cluster}, E^{cluster} = FC(V, E,$\mathbf{W}$, $\hat{m}$, k_1) //the 1st stage
3: $\hat{v}^{(2)}$ = FS(V^{cluster}, E^{cluster},$\mathbf{W}^{\text{cluster}}$,$k_2$, $\hat{m}$) //the 2nd stage
4: Return $\hat{v}^{(2)}$
5: **end procedure**

Algorithm 2 Find the Most Likely Candidate Cluster (First Stage)

1: **procedure** FC(V, E, $\mathbf{W}$, $\hat{m}$, k_1)
2: V^{gate},E^{gate},$\mathbf{W}^{\text{gate}}$ = CF(V, E, $\mathbf{W}$)
3: Find the $\hat{m}$ most likely graphs: G_{s}
4: Pick Sensors: Top(FBCS(G_{s}), k_1)
5: Compute $\mathbf{\Delta t}$
6: Initialize likelihood = 0
7: **for** each node v in V^{gate} **do**
8: Compute likelihood(v) using Eqn. 5
9: **end for**
10: $\hat{v}^{(1)}$ = max(likelihood)
11: Return V^{cluster}, E^{cluster}
12: **end procedure**

III. Source Localization Algorithm

In this section, we propose an algorithm to identify the source of diffusion in a social network with varying relationship strength. Our proposed two-stage algorithm is presented in Algorithm 1. The first stage of the proposed procedure (*FC*) is depicted in Algorithm 2. As shown in Algorithm 2, we first need to discover the clusters/communities existing in the network (line 2). We use the Louvain method [40] to find the clusters. The method is a greedy algorithm with time complexity of $O(|V|\log|V|)$. Since we assume that the structure of network communities does not change between the times rumor is initiated and received at the sensors, the community detection can be done offline. The gateway graph is constructed using the gateway nodes of these clusters. Since it is reasonable to expect that any piece of information flows along the shortest paths into the network, the most appropriate sensor nodes will be the nodes with high betweenness centrality (BC), where, BC of a node v is defined as $\text{BC}(v) = \sum_{s\neq v\neq t}(N^{\text{SP}}_{s,t}(v)/N^{\text{SP}}_{s,t})$. $N^{\text{SP}}_{s,t}(v)$ is the number of shortest paths from s to t passing through node v and $N^{\text{SP}}_{s,t}$ is the total number of shortest paths from node s to node t [41], [42].ê The number of shortest paths varies with the graph structure, therefore, it is not accurate to compare BC values across the possible graphs. Thus, we focus on *betweenness centrality order*, where the nodes are ranked based on their BC values and betweenness centrality score (BCS) of 1 is assigned to the node with the highest BC value [30]. In our case, although the size of the graph is the same, the varying weights between the nodes imply varying connectivity; hence, we approximate the expected BCS for each node $v \in V^{\text{gate}}$ using the $\hat{m}$ most likely graphs as

$$\overline{\text{BCS}(v)} = \sum_{j=1}^{\hat{m}} \Pr\left(G_j^{\text{gate}}\right)\text{BCS}_j^{\text{gate}}(v) \tag{7}$$

where $\text{BCS}_j^{\text{gate}}(v)$ is the BCS for the vth node in the jth possible graph G_j^{gate}. The computational complexity is $O(\hat{m}\cdot|E^{\text{gate}}|\cdot|V^{\text{gate}}|)$, where $|E^{\text{gate}}|$ and $|V^{\text{gate}}|$ denote the number of edges and nodes in the graph G_j^{gate}, respectively. The algorithm that finds the BCS values FBCS is shown in Algorithm 4. As shown in Algorithm 2, procedure FC selects the top k_1 nodes with high BC (line 4). In lines 7–9 of Algorithm 2, the procedure FC finds the most likely cluster (V^{cluster}, E^{cluster}). The procedure FS, shown in Algorithm 3, implements the second stage of the rumor localization. Similarly, procedure FBCS selects k_2 nodes as sensors to measure the arrival times of the rumor (line 3). Finally, the node that maximizes the likelihood value in line 10 is chosen as the source of the rumor. The time complexity of the algorithm is $O(a|V|^3)$, where $|V|$ is the number of gateway nodes in the first stage and the number of nodes in the most likely cluster in the second stage. a is the ratio of the nodes acting as the sensor in each stage. In addition to the community detection, sensor selection, gateway graphs, the mean and variance of edge delays, and finding $\hat{m}$, in (5) and (6), can be done offline. The algorithm is implemented in Python using NetworkX for network analysis [43]. We used two Linux Ubuntu machines with 2.40 GHz CPU and 8 GB memory. The run time of the algorithm is 4500 s on average, for a Sina Weibo network data set with 23 000 nodes and 43 470 edges, where the percentage of sensors is set to 0.4%.

Algorithm 3 Find the Source of Rumor (Second Stage)

1: **procedure** FS(V^{cluster}, E^{cluster}, $\mathbf{W}^{\text{cluster}}$, k_2, $\hat{m}$)
2: Find the $\hat{m}$ most likely graphs: G_{s}
3: Pick Sensors: Top(FBCS(G_{s}), k_2)
4: Compute $\mathbf{\Delta t}$
5: Initialize likelihood = 0
6: **for** each node v in V^{cluster} **do**
7: Compute likelihood(v) using Eqn.6
8: **end for**
9: $\hat{v}^{(2)}$ = max(likelihood)
10: Return $\hat{v}^{(2)}$
11: **end procedure**

IV. Simulation Results

In this section, we present the numerical results of the proposed algorithm. Experiments are conducted on three different

TABLE I
DETAILS OF DATA SETS

Network	Twitter	Weibo			Synthetic
	Values	Max	Ave	Min	Values
Number of nodes	22,175	23,516	21,915	20,495	23,372
Number of edges	40,480	44,446	42,772	40,990	33,101
Diameter[3]	13	10	8	7	15
Average shortest path length	5.79	5.41	4.97	4.49	6.19
Number of clusters	159	209	196	185	137

Algorithm 4 Find the BCSs (Required to Find the Sensors)

procedure FBCS(G_s)
  Initialize $\overline{\mathbf{BCS}} = \mathbf{0}$
  for each graph G_j in G_s **do**
    Compute the betweenness centrality values for the nodes in G_j: $\mathrm{BC}(G_j)$
    $\mathrm{BCS}(G_j) = \mathrm{HeapSort}(\mathrm{BC}(G_j))$
    for each node v in V **do**
      $\overline{\mathrm{BCS}(v)} = \Pr(G_j)\mathrm{BCS}_j^G(v) + \overline{\mathrm{BCS}(v)}$
    end for
  end for
  Return $\overline{\mathbf{BCS}}$
end procedure

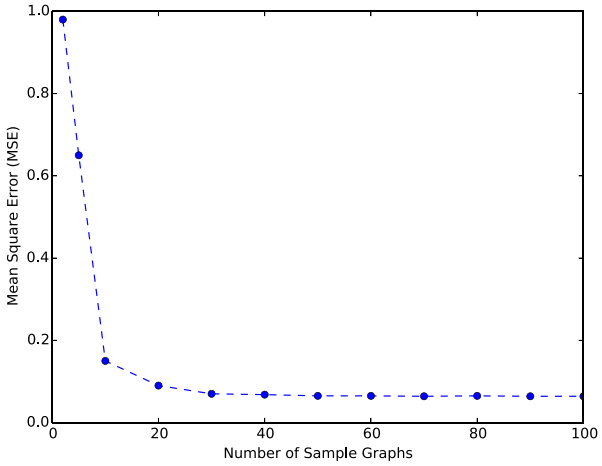


Fig. 4. MSE versus the number of sample graphs (the Twitter data set).

data sets: 1) a Twitter [44] subnetwork that we extracted; 2) the Sina Weibo network [32]; and 3) synthetic data on a Twitter subnetwork extracted by Leskovec and Mcauley [45]. Table I summarizes the details of these data sets. The network data sets are chosen with the similar number of nodes to make a fair comparison. For the first case, we crawled the Twitter networks for users who mentioned "Python" or "data" on their posts and then tracing followers links up to three hops. The original network size was 498 302 nodes. Since there are several inactive nodes in this network, which do not contribute to the spread of rumors, we trim this network by throwing away all nodes that have not exhibited any activity from October 5, 2014 to October 26, 2014. This gives us an overall size of 22 175 nodes for the network (Table I). To quantify the average information propagation delay, we extract the time difference between a tweet and its retweet. The mean shift method [46] is used to cluster the propagation delay values. For each edge e_i, we have $\Pr(\mu_i = \mu_{il}) = n_l / \sum_{k=1}^{M_i} n_k$, where μ_{il} and n_l $(1 \le l \le M_i)$ are the average information propagation delay and the number of points in the lth cluster, respectively. M_i is the total number of clusters. Based on (2), $w_{ij} = \mu_{ij}/\mu_{i1}$, where $\mu_{i1} = \max(\mu_{i1}, \mu_{i2}, \ldots, \mu_{iM_i})$.

Using the Louvain method, 159 clusters are found, where the average number of nodes in each cluster is 146. Therefore, on an average, only 146 nodes need to be searched in the second stage.

We obtain the sufficient number of possible graphs, $\hat{m}$, by determining the smallest number of possible graphs that does not increase the average shortest path distance [39]. The average shortest path distance is given by $\overline{\mathrm{SPD}} = \sum_{G_i \in G_s, |G_s|=m} P(G_i)\overline{\mathrm{SPD}(G_i)}$, where $\overline{\mathrm{SPD}(G_i)} = (1/(|V| \cdot (|V|-1))) \sum_{i \in V; i \ne j} \sum_{j \in V} \mathrm{SPD}_{i,j}$ and $\mathrm{SPD}_{i,j}$ is the shortest path distance between nodes i and j in G_i. The metric we used in this experiment is the mean square error (MSE) between the average shortest path distances of m $(2 \le m \le 100)$ and 500 possible graphs (a large number of possible graphs as our ground truth).[3] Since the MSE converges to 0.05 after 30 sample graphs in Fig. 4, we conclude that 30 sample graphs ($\hat{m} = 30$) are sufficient for our test needs. Similarly, $\hat{m}$ for the Sina Weibo and the synthetic network are 12 and 21, respectively. We then sort the possible graphs using their probability of occurrence [see (1)]. We finally pick the top $\hat{m}$ probable possible graph (or weight vectors) to construct the MLE.

Our second data set is derived from Sina Weibo, includes a followership network with 58 655 849 nodes and 265 580 802 edges, and a total of 370 million tweets and retweets [31], [32]. We selected 100 original tweets that constituted 100 different real diffusion networks. Note that the original tweets are chosen such that the size of each corresponding network is close enough to the size of the extracted Twitter network (Table I). We also evaluate the algorithm using synthetic data on a Twitter subnetwork extracted by Leskovec and Mcauley [45]. Table I summarizes the details of the synthetic network. In the synthetic data, the edge weights are independent and identically Gaussian distributed. We set $\mu/\sigma = 4$, where μ and σ are the mean and standard deviation, respectively. We simulate the information spread on Twitter and the synthetic networks using the SI model. Since there is no prior knowledge of the source of diffusion, we generate

[3] $\mathrm{MSE}(m) = (\overline{\mathrm{SPD}(m)} - \overline{\mathrm{SPD}(500)})^2$

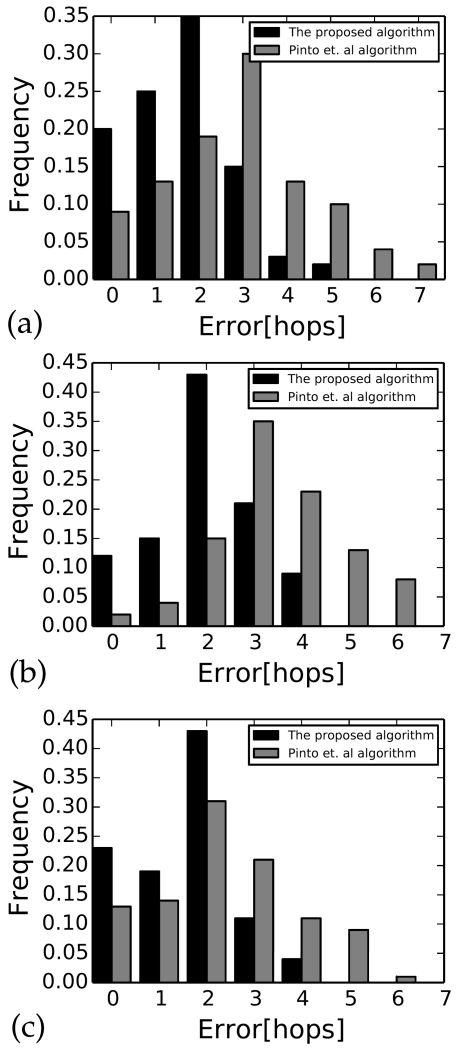


Fig. 5. Histogram of the error for the two-stage and the Pinto *et al.* algorithms where the percentage of sensors is set to be 0.4%. (a) Twitter network. (b) Sina Weibo network. (c) Synthetic network. The details of data sets are listed in Table I.

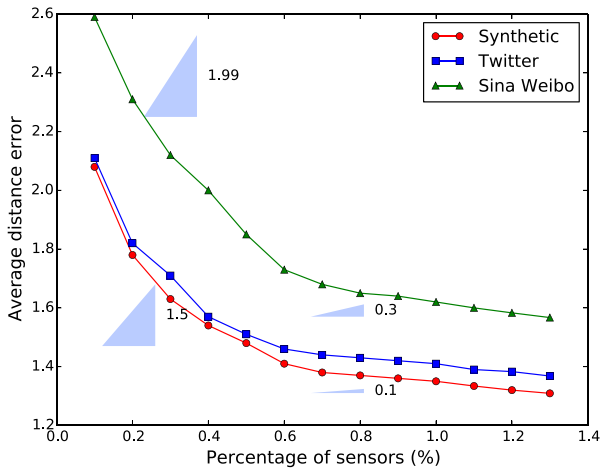


Fig. 6. Effect of the percentage of sensors on the average distance error (the details of data sets are listed in Table I). The slopes of the curves are also shown.

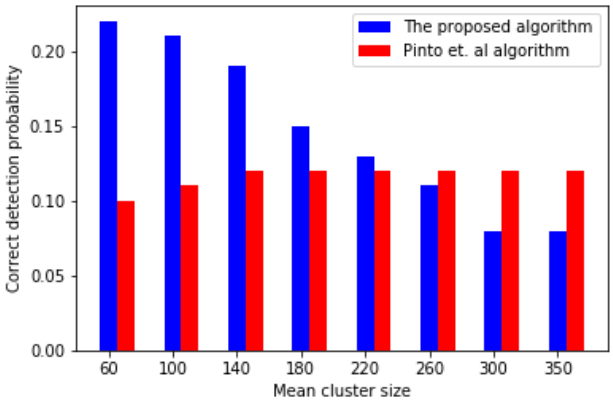


Fig. 7. Correct detection probability versus mean cluster size for a synthetic network with 25 000 nodes.

a uniformly distributed source in $[1, |V|]$. The following results are obtained by averaging over 100 independent runs. The percentage of sensors is fixed at 0.4% in the following experiments.

Fig. 5 illustrates the histogram of the error in source localization for the three network data sets. We also compare the performance of our two-stage algorithm to the single-stage algorithm in [24], where sensor nodes are picked based on degree of centrality rather than BC. Fig. 5(a) shows that the algorithm is able to pinpoint the source exactly 20% of the time in the Twitter network, but 80% of the time, the actual source is within two hops of the estimated source, when the diameter of the diffusion graph is 13. In comparison, the Pinto *et al.* algorithm makes an error of three hops or more 80% of the time. For the Weibo network [Fig. 5(b)], the proposed algorithm finds the source within two hops, 69% of the time. The network diameter varies between 7 and 10, in this case. However, the Pinto *et al.* algorithm finds the source within two hops, 21% of the time. The longer tails of the error is another indicator that the accuracy of our method is better than that of the method in [24].

Next, we study the effect of the number of sensors in the network on the accuracy of the estimated source. Fig. 6 shows the percentage of nodes used as sensors versus the accuracy as measured in number of hops between the estimated source and the actual source. As can be seen from this graph, the accuracy increases rapidly as the number of sensors increases from 0.1% to 0.6% of the total nodes in the network, but the increase slows down, by the time the number of sensors reaches 1% (the slopes of the curves, depicting the rate of decrease of errors with respect to increase in the percentage of sensors, is also shown in Fig. 6). There is a tradeoff between the computational complexity (the percentage of sensors) and the accuracy of locating the source. Fig. 6 can be used to obtain a reasonable compromise between the computational complexity with the accuracy of locating the source, where we would like to ensure the scalability of our algorithm to large social networks for a desired accuracy.

We now study the effect of the network modularity on the accuracy of estimating the source. To do that, we generated networks with different modularity synthetically using

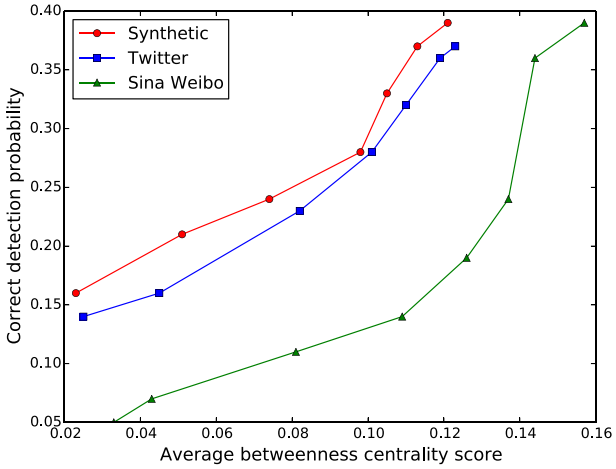


Fig. 8. Effect of average BCS on the correct detection probability where the percentage of sensors is set to be 0.4% (the details of data sets are listed in Table I).

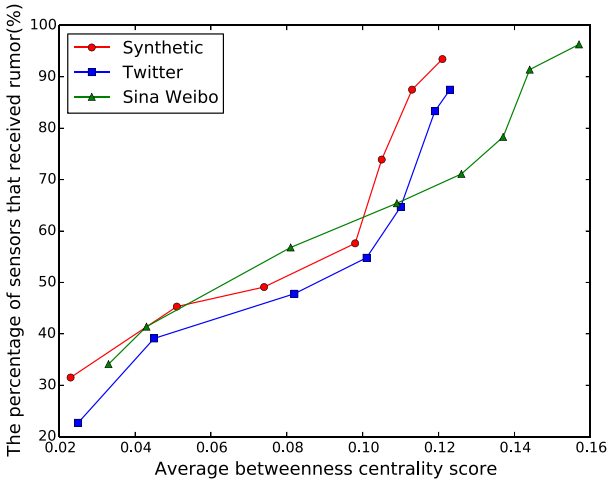


Fig. 9. Effect of average BCS on the number of sensors receiving the rumor where the percentage of sensors is set to be 0.4% of the entire network (the details of data sets are listed in Table I).

the method discussed in [47]. The methodology in [47] is a modified version of the planted l-partition model, where cluster sizes have a Gaussian distribution with given mean μ_c and variance σ_c. It leads to the networks that represent the heterogeneity of real social networks better, both in terms of the degree centrality and the cluster sizes. The edge weights are Gaussian distributed where μ/σ is set to be 4. Fig. 7 shows the probability of detecting the source correctly as we increase the average of the cluster size. It can be seen that the accuracy reduces as we increase the number of clusters (the time complexity is decreased). This is due to this fact that for networks with more clusters, even a small error at the first stage can cause a larger error in the location of the source. However, the accuracy of the single-stage algorithm does not change significantly with the network modularity. Next, we investigate some properties of the source node and its effect on the accuracy of detection. Fig. 8 shows the effect of an average BCS of the source on the probability of accurately estimating it. We note that as the BCS of the actual source node increases, our algorithm is able to locate this source with better accuracy. This is intuitively understandable, since as the BCS of the source increases, the probability that a higher number of sensors observing the rumor increases (Fig. 9) because the source node lies on the shortest paths to more nodes in the network.

V. Conclusion

In this paper, the problem of locating the source of a rumor in large-scale social networks is addressed. We model the uncertainty of internode relationship strengths using a probabilistically weighted graph. We develop a two-stage algorithm for partially observed social networks, in which in the first stage, the most likely candidate cluster to contain the source of the rumor is identified. In the second stage, the source is estimated from the set of nodes inside the most likely candidate cluster. This translates to a significant improvement in terms of computational complexity for large social networks. We perform numerical simulations to validate our algorithm on a large data set from Twitter and Sina Weibo. Our results show that the average estimation error in the Twitter network does not exceed two hops of the actual source for most of the time when the network diameter is 13 and the percentage of sensors is 0.4%. The numerical results show that the source localization becomes harder for social networks with more clusters and a smaller diameter. We also observe that the localization accuracy greatly increases as the BC of the source increases. Finally, we demonstrate that our two-stage method leads to significant improvement in terms of the accuracy of locating the source compared to existing approaches.

References

[1] *Number of Monthly Active Facebook Users Worldwide as of 4th Quarter 2017*. Accessed: Jan. 2018. [Online]. Available: http://www.statista.com/
[2] B. Doerr, M. Fouz, and T. Friedrich, "Why rumors spread so quickly in social networks," *Commun. ACM*, vol. 55, no. 6, pp. 70–75, Jun. 2012.
[3] G. Strauss, A. Shell, R. Yu, and B. Acohido, "SEC, FBI probe fake tweet that rocked stocks," USA Today, Apr. 2013.
[4] E. Morozov, "Swine flu: Twitters power to misinform," Foreign Policy, Washington, DC, USA, Apr. 2009.
[5] M. O. Jackson, *Social and Economic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[6] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. R. Soc. Lond. A, Math. Phys. Sci.*, vol. 115, no. 772, pp. 700–721, 1927.
[7] M. Barthélemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, "Velocity and hierarchical spread of epidemic outbreaks in scale-free networks," *Phys. Rev. Lett.*, vol. 92, no. 17, p. 178701, 2004.
[8] T. Zhou, J.-G. Liu, W.-J. Bai, G. Chen, and B.-H. Wang, "Behaviors of susceptible-infected epidemics on scale-free networks with identical infectivity," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 5, p. 056109, 2006.
[9] H. W. Hethcote, "Qualitative analyses of communicable disease models," *Math. Biosci.*, vol. 28, nos. 3–4, pp. 335–356, 1976.
[10] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[11] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *Proc. 31st IEEE Int. Conf. Data Eng. (ICDE)*, Apr. 2015, pp. 651–662.
[12] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *Proc. 13th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2013, pp. 1103–1108.
[13] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proc. ACM SIGKDD Workshop Mining Data Semantics*, 2012, Art. no. 13.

[14] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 1589–1599.
[15] C. H. Comin and L. da Fontoura Costa, "Identifying the starting point of a spreading process in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, p. 056105, Nov. 2011.
[16] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: Theory and experiment," *SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 203–214, Jun. 2010.
[17] D. Shah and T. Zaman, "Rumor centrality: A universal source detector," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 199–210, Jun. 2012.
[18] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2850–2865, Jun. 2013.
[19] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *Proc. IEEE 12th Int. Conf. Data Mining (ICDM)*, 2012, pp. 11–20.
[20] K. Zhu and L. Ying, "Information source detection in the SIR model: A sample path based approach," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2013, pp. 1–9.
[21] Z. Wang, W. Dong, W. Zhang, and C.-W. Tan, "Rooting our rumor sources in online social networks: The value of diversity from multiple observations," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 663–677, Jun. 2015.
[22] W. Luo and W. P. Tay, "Finding an infection source under the SIS model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 2930–2934.
[23] W. Dong, W. Zhang, and C. W. Tan. (2013). "Rooting out the rumor culprit from suspects." [Online]. Available: https://arxiv.org/abs/1301.6312
[24] 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.
[25] A. Louni and K. P. Subbalakshmi, "A two-stage algorithm to estimate the source of information diffusion in social media networks," in *Proc. IEEE INFOCOM Workshop Dyn. Social Netw.*, Apr. 2014, pp. 329–333.
[26] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, Mar. 2001.
[27] L. E. Celis, F. Pavetić, B. Spinelli, and P. Thiran, "Budgeted sensor placement for source localization on trees," *Electron. Notes Discrete Math.*, vol. 50, pp. 65–70, Dec. 2015.
[28] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 981–990.
[29] L. Weng, A. Flammini, A. Vespignani, and F. Menczer, "Competition among memes in a world with limited attention," *Sci. Rep.*, vol. 2, Mar. 2012, Art. no. 335.
[30] J. J. Pfeiffer, III, and J. Neville, "Methods to determine node centrality and clustering in graphs with uncertain structure," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media (ICWSM)*, 2011, pp. 590–593.
[31] *Sina Weibo*. [Online]. Available: http://weibo.com/
[32] *WISE 2012 Challenge*. Accessed: Jan. 2018. [Online]. Available: http://www.wise2012.cs.ucy.ac.cy/challenge.html
[33] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic, "The role of social networks in information diffusion," in *Proc. 21st Int. Conf. World Wide Web (WWW)*, 2012, pp. 519–528.
[34] M. Granovetter, "The strength of weak ties," *Amer. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, 1973.
[35] W. Luo, W. P. Tay, and M. Leng, "How to identify an infection source with limited observations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 586–597, Aug. 2014.
[36] *Supplemental Material*. Accessed: Jan. 2018. [Online]. Available: https://doi.org/10.1103/PhysRevLett.109.068702
[37] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
[38] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1974.
[39] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, "K-nearest neighbors in uncertain graphs," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 997–1008, Sep. 2010.
[40] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech., Theory Experim.*, vol. 2008, no. 10, p. P10008, 2008.
[41] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, Dec. 1966.
[42] P. Bonacich, "Power and centrality: A family of measures," *Amer. J. Sociol.*, vol. 92, no. 5, pp. 1170–1182, 1987.
[43] (Jan. 2018). *NetworkX*. [Online]. Available: https://networkx.github.io/
[44] (Jan. 2018). *Twitter Developer Platform*. [Online]. Available: https://dev.twitter.com/
[45] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 539–547.
[46] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[47] U. Brandes, M. Gaertler, and D. Wagner, *Experiments on Graph Clustering Algorithms*. Berlin, Germany: Springer, 2003, pp. 568–579.

Alireza Louni received the B.Sc. and M.Sc. degrees in electrical engineering with a focus on communication systems from the Sharif University of Technology, Tehran, Iran, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Stevens Institute of Technology, Hoboken, NJ, USA.

He was a Data Scientist for some data companies in Silicon Valley, CA, USA. He is currently a Data Scientist with Quid, Inc. His current research interests include the mathematical modeling of social media networks and the applications of machine learning and data mining for massive real-world data sets.

K. P. (Suba) Subbalakshmi is currently a Professor with the Department of ECE, Stevens Institute of Technology, Hoboken, NJ, USA, and a Jefferson Science Fellow (JSF). As a JSF, she focused on technology policy issues with the Department of State in 2016. She is also a co-founder of two technology startups that commercializes her research. Her work is funded by NSF, DoD, and Industry. Her current research interests include the areas of social media analysis, security and forensics, cognitive radio networks, and cognitive mobile cloud computing.

Dr. Subbalakshmi is a founding Associate Editor for the IEEE TRANSACTIONS ON COGNITIVE NETWORKS. She was a recipient of the Innovator Award from the New Jersey Inventors Hall of Fame.