A PARTICLE FILTER FOR SEQUENTIAL INFECTION SOURCE ESTIMATION

Wenchang Tang and Wee Peng Tay

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

ABSTRACT

In this paper we study the problem of identifying an infection source in a network based only on the network topology and a stream of infection timestamps. We propose a sequential source estimation algorithm (SSE) using a particle filter that is based on an approximate hidden Markov chain model, which can be interpreted as a "reverse" propagation process. Simulations using synthetic networks and experiments using real-world social network data suggest that SSE is able to estimate the true infection source to within a small number of hops with less than 20% of the infection timestamps being observed.

***Index Terms*—** Infection source, diffusion process, sequential source estimation, partial timestamps, particle filter

1. INTRODUCTION

Online social networks such as Twitter, Facebook and Sina Weibo are becoming increasingly important sources of information for its users [1]. Information on a social network can be posted, reposted and shared within social circles. However, some postings may be false rumors. These rumors can spread quickly and can potentially mislead the public [2], triggering undesirable widespread panic [3]. Similarly, a single piece of negative news on a celebrity may damage his public reputation and even his career. Regulatory agencies may wish to identify and prosecute such rumor monger. Such rumors can be modeled using a diffusion or infection process [4]. Epidemics spreading in a community can also be modeled using an infection process [5]. Finding the patient zero of an epidemic can aid medical scientists in analyzing the epidemic's pathogenesis. So locating the source of a rumor or an epidemic is often desirable. In this paper, we consider an infection source initiating a diffusion process in a network. Our goal is to identify this infection source based on knowledge of the network topology and a stream of infection timestamps that we observe sequentially.

Related work: The problem of inferring the source of an infection process over a network has attracted much recent attention. Most of these works perform source estimation based on a snapshot observation of the infection status of some or all nodes in the network. Various infection spreading models have been studied in the literature, including the susceptible-infected (SI) [6], susceptible-infected-recovered (SIR) [7], and susceptible-infected-susceptible (SIS) [8] models. In the SI model, an infected node remains infected forever, in the SIR model, it can recover and cannot be further infected, and in the SIS model, a recovered node can become infected again.

A rumor centrality estimator under the SI model was developed in [6], [9], while [10–12] developed estimators for identifying multiple infection sources under the SI model. The paper [13] considers infection source estimation when only a subset of infected nodes are observed, and a strategic game between infection spreading and source identification was studied in [14]. In [15], multiple observations of a SI spreading process are used for source estimation. In [7] and [16], infection source estimation is investigated under the SIR model, and [8] considers the SIS model. All the aforementioned works perform source estimation based only on the observed status of the nodes, and the network topology. Other related works assume additional a priori knowledge like the infection spreading rate, including [17] which performs source inference via belief propagation, and [18] which developed a dynamic message passing algorithm.

Network source estimation has also been performed using infection timestamps. In this framework, we observe the first infection times of a subset of nodes in the network, and together with knowledge of the network topology, the infection source is inferred. Clearly, if the infection times of all infected nodes are observable, then the problem becomes trivial. Therefore, in this approach, we aim to find estimators that can reasonably infer the source given only a small number of infection timestamps. In [19], an algorithm based on the difference in infection times of each node with a chosen reference node is proposed. Since the infection spreading time over each edge is assumed to be a Gaussian random variable, we call this algorithm GAU. As GAU was developed based on tree networks, our experiments show that it does not work well in dense graphs. Moreover, its time complexity is $O(N^3)$,[1] where N is the number of nodes, and the algorithm scales poorly for large graphs with thousands of infected nodes. Two ranking algorithms have been proposed in [20], which demonstrate improved estimation accuracy compared to GAU. The aforementioned algorithms all work in a "batch" mode, where all the infection timestamps are assumed to be available. If a new timestamp is subsequently made available, the whole estimation procedure has to be repeated, leading to higher computation complexity. Such algorithms are thus not suitable for online estimation if the underlying network is large.

Our contributions. All of the previous works perform source estimation based on a single snapshot observation of either the node status or infection timestamps, or on multiple snapshot observations of different infection cascades. In practical applications, observations of either a node status or infection timestamp may arrive sequentially, and sometimes we do not know when the next observation will be available. In a sequential estimation setting, our goal is to develop algorithms that can estimate the source using all observations up to the current time, and be able to quickly update our estimate when new observations are available. In this paper we propose a novel sequential source estimation (SSE) algorithm that can estimate the infection source using a stream of infection timestamps arriving sequentially. To the best of our knowledge, this is the first sequential infection source estimation in the literature. Our proposed SSE algorithm is based on the construction of a "reverse" propaga-

This research is supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grants MOE2013-T2-2-006 and MOE2014-T2-1-028.

[1] We say that $f(N) = O(g(N))$ if $f(N)/g(N) \leq k$ for some fixed $k > 0$ as $N \to \infty$.

tion process, which allows us to perform particle filtering on the sequential observations. Simulation results show that our algorithm can quickly and accurately update source estimation as timestamps arrive sequentially.

The rest of this paper is organized as follows. In Section 2, we formulate the sequential source estimation problem. In Section 3, we construct a reverse infection propagation process and present our SSE algorithm. Simulation results on both synthetic and real-word data are given in Section 4. Section 5 concludes the paper. We use $\mathrm{Unif}\,(A)$ to denote the uniform distribution over the set A, $\mathcal{N}(\mu, \sigma^2)$ to be the Gaussian distribution with mean μ and variance σ^2, $\mathrm{Exp}(\lambda)$ to denote the exponential distribution with rate λ, and $\Gamma(\alpha, \beta)$ to be the Gamma distribution with shape parameter α and rate parameter β. We use $p(x \mid y)$ to represent the conditional distribution of x given y. The symbol $\sim$ indicates equality in distribution.

2. PROBLEM FORMULATION

We model the network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges. At an unknown time t_0, an infection source $s^* \in \mathcal{V}$ initiates a diffusion process. Let ∂v be the set of neighbors of node v and τ_{uv} as the random propagation delay associated with edge (u, v). We assume that all $\{\tau_{uv}\}$ are independent and identically distributed (i.i.d.) continuous random variables and each follows a distribution with probability density function $f_\tau(\cdot)$. Let $d(u, v)$ be the length of a shortest path between node u and node v in $\mathcal{G}$.

We suppose that we observe the infection timestamps (v_n, t_n), $n \geq 1$ sequentially, where v_n is the identity of the n-th node observed to be infected, and t_n is its first infection time. We do not assume that the infection times $\{t_n\}$ are ordered in any way as a node that was infected earlier could be observed only later. Let $\mathcal{O}_n = \{(v_k, t_k) : k \leq n\}$ be the set of observations up to the n-th timestamp, and let

$$s_n = s_n(\mathcal{O}_n) \tag{1}$$

be the infection source estimate based on $\mathcal{O}_n$. Our aim is to design s_n so that s_{n+1} can be updated from it easily, and s_n is in some sense close to the true source s^* for n sufficiently large.

3. SEQUENTIAL SOURCE ESTIMATION

In this section, we first construct a stochastic process that we use to represent the reverse infection propagation process. Then, using a series of approximations, we derive a particle filter for sequentially estimating the infection source given a stream of infection timestamps.

3.1. Reverse Propagation Process

We define a $\mathcal{V}$-valued discrete-time Markov chain $(X_n)_{n\geq 0}$ where $X_0 \sim \mathrm{Unif}\,(\mathcal{V})$, and for $n \geq 1$, the conditional probability of $X_n = x_n$ given $X_{n-1} = x_{n-1}$ is given by

$$p(x_n \mid x_{n-1}) = \begin{cases} \alpha & \text{if } x_n = x_{n-1}, \\ \frac{1-\alpha}{|\partial x_{n-1}|} & \text{if } x_n \in \partial x_{n-1}, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $|\partial x_{n-1}|$ is the number of neighbors of node x_{n-1}, and $\alpha \in (0, 1)$ is denoted as self-transition probability. We interpret this as a "reverse" propagation process as follows: Consider the infection path $\rho(s^*, v_1)$ from s^* to the first observed infected node v_1. The node X_1 is a node in $\rho(s^*, v_1)$. We say that X_1 is an ancestor to v_1. Similarly, given v_2, we wish to find a common ancestor in $\rho(s^*, v_1) \cap \rho(s^*, v_2)$ to both v_1 and v_2. If a neighbor of X_1 is also its ancestor in $\rho(s^*, v_1) \cap \rho(s^*, v_2)$, we let X_2 be chosen to be this neighbor, otherwise we let $X_2 = X_1$. This procedure is then repeated. We approximate this stochastic process with the Markov chain described in (2), where α is an appropriately chosen constant. In Section 4, we indicate how to choose this constant through simulations.

We can now cast our sequential source estimation problem as an approximate hidden Markov model (HMM). Let $(x_n)_{n\geq 0}$ be given by the Markov chain defined in (2), and the hidden states be $(z_n)_{n\geq 1} = (x_n, \mathcal{O}_{n-1})_{n\geq 1}$, where $\mathcal{O}_0 = \emptyset$. Let the observations be $(y_n)_{n\geq 1} = (v_n, t_n)_{n\geq 1}$. The hidden states $(z_n)_{n\geq 1}$ do not form a Markov chain, but we will adopt the particle filtering framework of a HMM as an approximation. Experiments in Section 4 indicate that this approximation does not unreasonably impair our estimation results.

To perform particle filtering, we make further approximations as follows. For the state transition probability, we take

$$\begin{aligned} p(z_n \mid z_{n-1}) &= p(x_n, \mathcal{O}_{n-1} \mid x_{n-1}, \mathcal{O}_{n-2}) \\ &\approx p(x_n \mid x_{n-1}), \end{aligned} \tag{3}$$

where we ignore the contributions of $\mathcal{O}_{n-1}$ and $\mathcal{O}_{n-2}$. We assume that infections are transmitted from the source to each infected node through the shortest path, so that the infection propagation graph is a tree. Suppose that x_n is infected at time $\hat{t}_{x_n}$. Then,

$$p(y_n \mid z_n) \approx f(t_n - \hat{t}_{x_n}; d(v_n, x_n)), \tag{4}$$

where $f(\cdot; d)$ is the d-fold convolution of f_τ. We estimate $\hat{t}_{x_n}$ from $\mathcal{O}_{n-1}$ as follows:

$$\hat{t}_{x_n} = \arg\max_t \prod_{i=1}^{n-1} f(t_i - t; d(v_i, x_n)), \tag{5}$$

where we have assumed that the infection time of each v_i, $i \leq n-1$, is independent of the others. This is an approximation to allow easy computation of $\hat{t}_{x_n}$.

To illustrate (5), we now derive the estimator $\hat{t}_{x_n}$ for the case where the propagation delay over each edge follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Then $f(\cdot; d)$ is the density of $\mathcal{N}(d\mu, d\sigma^2)$. It can be shown that in this case,

$$\hat{t}_{x_n} = \frac{\sum_{i=1}^{n-1}(t_i - d_i\mu)/d_i}{\sum_{i=1}^{n-1} d_i^{-1}}, \tag{6}$$

where $d_i = d(v_i, x_n)$. By taking the expectation of (6), it is clear that $\hat{t}_{x_n}$ in (6) is an unbiased estimate of t_{x_n}. In fact, this is true for any propagation delay distribution. For any propagation delay distribution for which no analytical form for $\hat{t}_{x_n}$ exists, either a numerical method can be used to solve (5) or we simply use (6) with μ being the mean of the distribution.

We can now describe the steps involved in our particle filter. Let $x_{1:n}$ represent $(x_i)_{i=1}^n$. For each $n \geq 1$, we wish to obtain a set of weighted particles $\{x_n^m, w_n^m\}_{m=1}^M$ so that we can approximate

$$p(x_n \mid y_{1:n}) \approx \sum_{m=1}^{M} w_n^m \delta\left(x_n - x_n^m\right), \tag{7}$$

where $\delta(\cdot)$ is the Dirac delta function. For each $n \geq 1$, we perform the following steps:

1. Sample particles from a proposal density

$$x_n^m \sim q\left(x_n \mid x_{1:n-1}^m, y_{1:n}\right). \tag{8}$$

2. Update $\{x_{n-1}^m, w_{n-1}^m\}$ to $\{x_n^m, w_n^m\}$ where

$$w_n^m \propto \beta_n w_{n-1}^m, \tag{9}$$

and

$$\beta_n = \frac{p(x_n^m \mid x_{n-1}^m) f(t_n - \hat{t}_{x_n}; d(v_n, x_n))}{q\left(x_n \mid x_{1:n-1}^m, y_{1:n}\right)}. \tag{10}$$

Various particle filter algorithms differ in their choices of proposal density $q(\cdot)$ in (8). The popular bootstrap filter in [21] uses transition function (3) as the proposal density, and thus β_n in (10) is reduced to the likelihood function $f(t_n - \hat{t}_{x_n}; d(v_n, x_n))$. In our SSE algorithm, we also adopt this strategy. To avoid degeneracy, we also implement a resampling step in which $\{x_n^m, w_n^m\}$ is resampled to obtain M equally-weighted particles $\{\overline{x}_n^m, 1/M\}$, which are then used in place of x_n^m in the sampling step (8). We adopt the widely used systematic resampling method [22], which outperforms other resampling schemes in most scenarios and is easy to implement.

Finally, the infection source is found by finding the maximum a posteriori estimate using (7), i.e., $s_n = x_n^{m_n}$, where

$$m_n = \arg\max_m w_n^m. \tag{11}$$

The detailed SSE algorithm is shown in Algorithm 1. For each new observed y_n where $n \leq |\mathcal{V}|$, we first perform a breadth-first search tree rooted at v_n and store the pairwise distances in a hash table. This process incurs a time complexity of $O(|\mathcal{V}|^2)$. We then compute $\hat{t}_{x_n^m}$ for $m = 1, ..., M_n$ according to (6), where $M_n \leq |\mathcal{V}|$ is the number of distinct nodes used as particles. The complexity of this step is $O(nM_n)$. We have $nM_n \leq |\mathcal{V}|^2$, therefore the overall time complexity for each new observed y_n is $O(|\mathcal{V}|^2)$. Experiments in Section 4 also show that M_n generally decreases as n increases because the resampling step allows us to discard nodes with low weight. At each new observation, SSE thus incurs a lower time complexity than GAU, which requires $O(|\mathcal{V}|^3)$ time complexity if it is to be recomputed for each new observation.

Algorithm 1 Sequential Source Estimation (SSE)

Input: Adjacent matrix of the graph $\mathcal{G}$, a stream of timestamps $(v_n, t_n)_{n=1}^N$, and $f_\tau(\cdot)$
Output: Infection source estimates $(s_n)_{n=1}^N$

```
for n = 1, ..., N do
  if n = 1 then
    Sample x_1^m from Unif(V), for m = 1, ..., M.
    Let x̄_1^m = x_1^m, w_1^m = 1/M.
  else
    Sample x_n^m ~ p(x_n^m | x̄_{n-1}^m) in (2) for m = 1, ..., M.
    Find distinct nodes {x_n^m}_{m=1}^{M_n} and count their respective
    numbers {c_n^m}_{m=1}^{M_n} where Σ_{m=1}^{M_n} c_n^m = M.
    for m = 1, ..., M_n do
      Compute t̂_{x_n^m} according to (6).
      Compute weight w_n^m = c_n^m p(y_n | z_n^m) according to (4).
    end for
    Normalize {w_n^m} so that Σ_{m=1}^{M_n} w_n^m = 1.
    Use systematic resampling to obtain M equally weighted
    particles {x̄_n^m, 1/M}_{m=1}^M.
    s_n = x_n^{m_n} where m_n = arg max_m w_n^m.
  end if
end for
```

4. EXPERIMENTAL EVALUATIONS

4.1. Synthetic Data

We first perform experiments on Erdos-Renyi (E-R) graphs $\text{ER}(n, np)$ where n is the number of nodes, and np is the expected degree of each node. We choose $n = 1000$, and $np = 2$ or 4 to simulate a graph that is relatively sparse or dense respectively. Some properties like the average diameter and pairwise distance are listed in Table 1. In each simulation, we assume that the propagation delay over every edge follows either a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ or an exponential distribution $\text{Exp}(\lambda)$.

We first generate 1000 timestamps initiated by a random node, and then randomly choose a portion of the timestamps to perform sequential source estimation. In Algorithm 1, we use $M = 2000$ particles. Simulation results for $\text{ER}(1000, 2)$ and $\text{ER}(1000, 4)$ are shown in Fig. 1 and Fig. 2, respectively. Each curve is averaged over 200 experiments. The estimation deviation is defined as the distance between the estimated source and the real source. From parts (a) and (c) in both Fig. 1 and Fig. 2, we see that α should be chosen from $(0.9, 1.0)$ to get a better performance. From parts (b) and (d) in both Fig. 1 and Fig. 2, we see that with 10% to 20% of the total timestamps, SSE finds an estimate that is on average within 2 hops of the real source if $\alpha = 0.95$.

We compare SSE with GAU proposed in [19] in Fig. 3. Since GAU requires computing a breadth-first search tree rooted at every infected node, it has high complexity for large graphs. Therefore, we perform experiments on smaller graphs $\text{ER}(200, 2)$. From Fig. 3(a) and Fig. 3(b), we see that SSE has smaller average estimation deviation and higher detection rate for both Gaussian and exponential spreading when there are less than 50% timestamps.

We also evaluate the performance of our algorithm with Facebook network provided by SNAP.[2] We extract a subgraph with 1034 nodes and 26749 edges for infection spreading using a Gaussian distribution for the edges' propagation delay with $\mu/\sigma = 4$. We then randomly choose timestamps to perform sequential source estimation (we let $\alpha = 0.95$ and average over 300 experiments). Results in Fig. 4 show that the average estimation deviation decreases with more timestamps, and approaches to around 1.4 hop with 10% of the total timestamps available. However, the performance is not as good compared to the E-R graphs, because the Facebook network is much denser.

Graph	$\lvert\mathcal{E}\rvert/\lvert\mathcal{V}\rvert$	Diameter	Pairwise distance
$\text{ER}(1000, 2)$	1	21.6	8.98
$\text{ER}(1000, 4)$	2	11	5.13
Facebook	25.9	9	2.95
Weibo	1.02	15	3.48

Table 1. Some graph properties of the networks used.

4.2. Weibo Data

We evaluate the performance of SSE with a sample of Sina Weibo data provided by the WISE 2012 challenge.[3] We first extract a sub-

[2] https://snap.stanford.edu/data/

[3] http://www.wise2012.cs.ucy.ac.cy/challenge.html

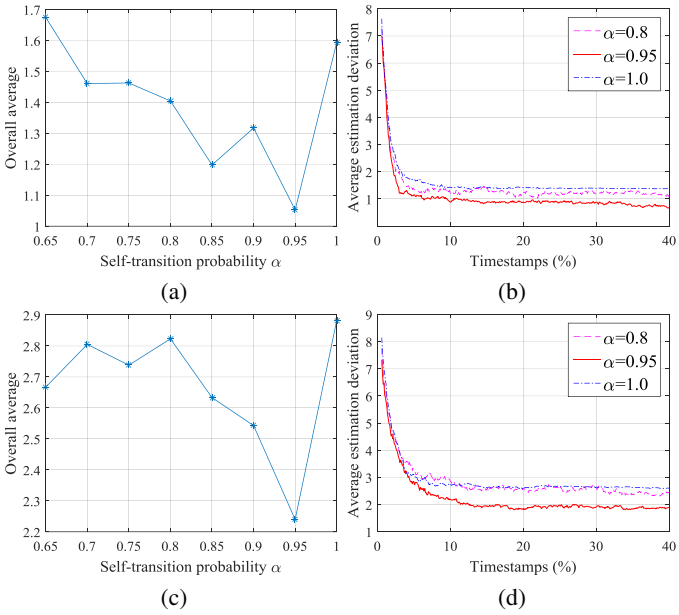

Fig. 1. Sequential source estimation on ER$(1000, 2)$. (a) and (b) are based on Gaussian spreading with $\mu/\sigma = 2$, while (c) and (d) are based on exponential spreading with $\lambda = 1$. In (b) and (d) we average the estimation deviation over 200 experiments. In (a) and (c) we obtain the overall average by averaging the estimation deviation over 200 experiments and 40% timestamps.

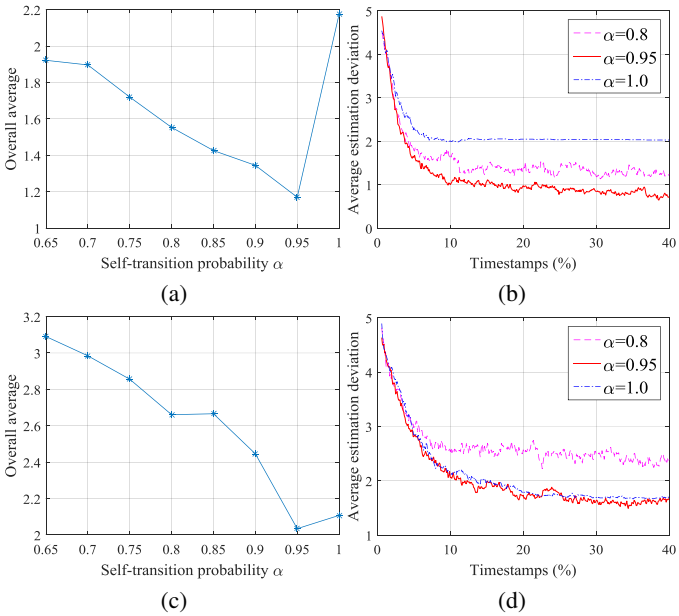

Fig. 2. Sequential source estimation on ER$(1000, 4)$. (a) and (b) are based on Gaussian spreading with $\mu/\sigma = 2$, while (c) and (d) are based on exponential spreading with $\lambda = 1$. In (b) and (d) we average the estimation deviation over 200 experiments. In (a) and (c) we obtain the overall average by averaging the estimation deviation over 200 experiments and 40% timestamps.

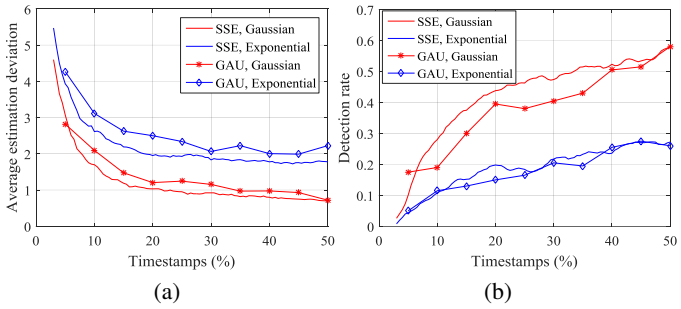

Fig. 3. Comparison between SSE and GAU on ER$(200, 2)$ for both Gaussian and exponential spreading, averaged over 300 experiments. Comparison is based on average estimation deviation in (a) and detection rate in (b). In SSE, we set $\alpha = 0.95$.

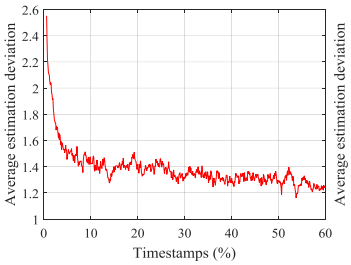

Fig. 4. Facebook network

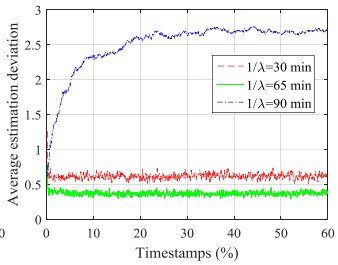

Fig. 5. Weibo data

graph with 4642 nodes and 4755 edges (if user a retweets user b's tweet then there is an edge between nodes a and b) using 44017 tweets. Some properties of the graph generated are summarized in Table 1. Secondly, we choose a message that was retweeted 2877 times and we use the first 2000 timestamps for SSE. The total time spanned by these tweets is about 5 hours. Finally, we learn the mean time of a tweet spreading through an edge from historical tweets, which yields an estimate of about 65 minutes. We set $\alpha = 0.95$ in SSE and adopt an exponential spreading model with different rates λ in SSE. We perform 300 experiments, and for each experiment, we randomly choose the ordering of the timestamps. Results averaged over all the experiments are shown in Fig. 5. We see that the performance is best when $1/\lambda = 65$ minutes, which matches the historical estimate, and is worse when $1/\lambda$ is too small or too big.

5. CONCLUSION

We have developed a sequential infection source estimation algorithm for identifying an infection source using a stream of infection timestamps. Our method is based on constructing a reverse propagation process, which allows us to build a particle filter for sequential estimation. Each sequential update has low complexity, which allows the source estimate to be updated quickly whenever a new timestamp is available. Experiments on both synthetic and real-world data suggest that our approach can find the infection source to within a small number of hops from the true source.

6. REFERENCES

[1] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 2006, pp. 611–617.

[2] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in *Proc. ACM Int. Conf. Web Search and Data Mining*, 2011, pp. 65–74.

[3] C. Stephen, "China's panic-buying of salt," http://www.marketwatch.com/story/what-chinas-panic-buying-of-salt-means-2011-03-20, accessed March 20, 2011.

[4] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, Jul. 2013.

[5] C. Scoglio, W. Schumm, P. Schumm, T. Easton, S. R. Chowdhury, A. Sydney, and M. Youssef, "Efficient mitigation strategies for epidemics in rural regions," *PLoS One*, vol. 5, no. 7, pp. 1–8, 2010.

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

[7] K. Zhu and L. Ying, "A robust information source estimator with sparse observations," in *Proc. IEEE Int. Conf. Comput. Commun.*, April 2014, pp. 2211–2219.

[8] W. Luo and W. P. Tay, "Finding an infection source under the SIS model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 2013, pp. 2930–2934.

[9] D. Shah and T. Zaman, "Rumor centrality: a universal source detector," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, 2012, pp. 199–210.

[10] 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, 2013.

[11] V. Fioriti, M. Chinnici, and J. Palomo, "Predicting the sources of an outbreak with a spectral technique," *Appl. Math. Sci.*, vol. 8, no. 133-136, pp. 6775–6782, 2014.

[12] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *Proc. IEEE Int. Conf. Data Mining*, Dec 2012, pp. 11–20.

[13] W. Luo, W. P. Tay, and M. Leng, "How to identify an infection source with limited observations," *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 4, pp. 586–597, 2014.

[14] ——, "Infection spreading and source identification: A hide and seek game," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4228–4243, Aug 2016.

[15] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rumor source detection with multiple observations: Fundamental limits and algorithms," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1, 2014, pp. 1–13.

[16] K. Zhu and L. Ying, "Information source detection in the SIR model: A sample-path-based approach," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 408–421, Feb 2016.

[17] F. Altarelli, A. Braunstein, L. Dall'Asta, A. Lage-Castellanos, and R. Zecchina, "Bayesian inference of epidemics on networks via belief propagation," *Phys. Rev. Lett.*, vol. 112, p. 118701, Mar 2014.

[18] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, "Inferring the origin of an epidemic with a dynamic message-passing algorithm," *Phys. Rev. E*, vol. 90, p. 012801, Jul 2014.

[19] 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.

[20] K. Zhu, Z. Chen, and L. Ying, "Locating the contagion source in networks with partial timestamps," *Data Mining and Knowledge Discovery*, pp. 1–32, 2015.

[21] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[22] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Proc. IEEE Int. Symp. Image and Signal Process. and Anal.*, 2005, pp. 64–69.