A Markov Chain Monte Carlo Approach for Source Detection in Networks

Le Zhang, Tianyuan Jin, Tong Xu(✉), Biao Chang, Zhefeng Wang, and Enhong Chen

Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
{laughing,jty123,chbiao,zhefwang}@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn

Abstract. The source detection task, which targets at finding the most likely source given a snapshot of the information diffusion, has attracted wide attention in theory and practice. However, due to the hardness of this task, traditional techniques may suffer biased solution and extraordinary time complexity. Specially, source detection task based on the widely used Linear Threshold (LT) model has been largely ignored. To that end, in this paper, we formulate the source detection task as a Maximum Likelihood Estimation (MLE) problem, and then proposed a Markov Chain Monte Carlo (MCMC) algorithm, whose convergence is demonstrated. Along this line, to further improve efficiency of proposed algorithm, the sampling method is simplified only on the observed graph rather than the entire one. Extensive experiments on public data sets show that our MCMC algorithm significantly outperforms several state-of-the-art baselines, which validates the potential of our algorithm in source detection task.

Keywords: Social network · Source detection · Markov Chain Monte Carlo

1 Introduction

Recent years have witnessed a massive increase in the popularity of social networks, which emerged as important platforms for information propagation. Some studies focus on modeling the diffusion process of information [10,21]. Some studies focus on the application of information diffusion [13,15,22]. *Information source detection* [2,15,20] is one of the popular application, which aims to find the source of given information based on a snapshot of the propagation network, and further support many propagation-related scenarios, such as rumor source tracing and virus source identification.

Since firstly proposed in [15] as a preliminary study, the information source detection task has been discussed by many prior arts, which are based on centrality [15–17], sampling [7,11,13,18,20], spectral [6], or belief propagation [1].

X. Cheng et al. (Eds.): SMP 2017, CCIS 774, pp. 77–88, 2017.
https://doi.org/10.1007/978-981-10-6805-8_7

Usually, most efforts have been made only on a simplified network topology, such as a tree or a grid. But it is hard for these approaches to guarantee high accuracy in general graphs. At the same time, some other related works assumed that information diffusion process followed the basic epidemic models, like the Susceptible Infected Recovered (SIR) model [10] or the Susceptible Infected (SI) model [15]. However, finding source in the liner threshold (LT) model which is one of the most popular models for social network analysis is rarely studied.

Recent studies [11,13] have noticed these mentioned facts, they attempted to solve the problem via maximizing the similarity between estimated and observed diffusion graph. However, these studies may be misled by the network structure. As an example, in Fig. 1(a), suppose that the observed nodes set $A = \{1, 2, 3, 5\}$ is given. v_2 will probably be estimated as the source by the K-effectors [11] and SISI [13]. In fact, only v_1 can activate all the nodes successfully. Therefore, the problem [11,13] studied is substantially different from the source detection. Clearly, more comprehensive method is still required.

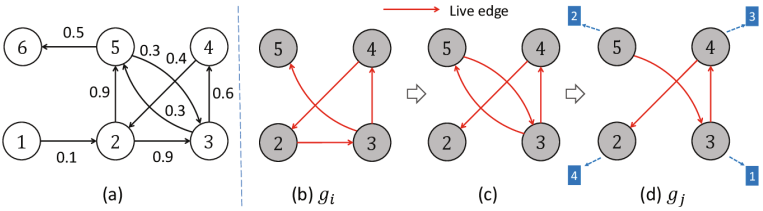


Fig. 1. (a): A social network instance in LT model. (b), (c) and (d) describe three possible diffusion process for given observed node set $A = \{2, 3, 4, 5\}$.

In light of the above, we will explore how to efficiently estimate the source in large social networks without bias. To be specific, the linear threshold (LT) model, one of the most popular models for social network analysis, will be used to describe the diffusion process. Then, the task of information source detection will be formalized as a maximum likelihood estimation (MLE) problem, where we try to pick up a node, i.e., the source, to maximize the likelihood of diffusion graph. To deal with that, considering that the size of the sample space is exponentially large compared with the number of edges in the network, we designed a Markov Chain Monte Carlo (MCMC) to efficiently detect the source. The contributions of our paper could be summarized as follows:

1. We formalize the problem of source detection under the LT model by maximum likelihood estimation (MLE). And to the best of our knowledge, we are the first to adapt the MCMC approach to solve the problem.
2. To further improve efficiency of the proposed algorithm, we reduce the sample space by sampling on the observed subgraph rather than the entire graph.
3. We conduct comprehensive experiments on a real network to validate the performance of the proposed approach. The results demonstrate the effectiveness of the proposed approach.

2 Related Work

In general, our related work could be divided into two categories, i.e., Source Detection and Markov Chain Monte Carlo Approach.

Source Detection. Due to the widespread applications in many real-world scenarios, a series of techniques for source detection have been studied. In the seminal works, Shah et al. [15,17] is the first to formulate the source detection and established the concept of rumor centrality under the SI model. Then they further improved the effectiveness of rumor centrality for source detection in generic trees [16]. Dong et al. [4] inferred a source with the maximum posteriori estimator on regular tree-type networks given a priori information. Wang et al. [19] found that multiple independent observations can significantly increase the detection probability for trees. Lappas et al. [11] formulated the source detection as the K-effectors which selects k active nodes that can best explain the observed nodes set in social networks. Fanti et al. [5] introduced a messaging protocol, which guarantees obfuscation of the source under the assumption that the network administrator utilizes the maximum likelihood (ML) estimator to identify the source. Chang et al. [3] derived a maximum posteriori estimator to find the source under the SI model. Nguyen et al. [13] used a similar formulation of K-effectors for source detection and provided an efficient guaranteed method.

Markov Chain Monte Carlo. Markov Chain Monte Carlo is a randomized sampling method which could sample from a probability distribution based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. Due to its widespread applications in many important problems like rare event sampling [8] and Bayesian Inference [14], this approach has been richly studied in recent years. Along this line, a series of twist methods have been proposed like Gibbs sampling and Slice sampling.

In conclusion, on the one hand, most existing works assume that information diffusion process followed the basic epidemic models, such as the SI model or SIR model. Whereas that solving the source detection problem based on the LT model is rarely studied. On the other hand, although MCMC has been wildly used in rare event sampling, it is rarely used in source detection. Hence, we explore the MCMC approach to efficiently estimate the source in general graphs based on the LT model.

3 Problem Formulation

In this section, we first give the formal definition of the information source detection problem. Then, we introduce the widely used diffusion model called the LT model. At last, the basic Monte Carlo method is proposed to solve it.

3.1 Problem Definition

Given a social network $G = (V, E)$, where V is the node set, $E \subseteq V \times V$ is the set of edges and a snapshot of the information diffusion when diffusion process

ends. Denote by A the set of infected nodes observed in V. Assuming that node v is the source, denote by $P(A|G, v)$ the probability that the infected node set we observe in G is A. According to the principle of Maximum Likelihood Estimation (MLE), the estimated source node is $arg\ max_v\ P(A|G, v)$. Formally, we define the source detection problem as follows.

Problem 1. **Information Source Detection Problem**. Given a graph $G = (V, E)$ and the infected node set A when diffusion process ends, the information source detection problem asks for finding a single source node $\hat{s}$ that most likely start the diffusion, that is $\hat{s} = arg\ max_v\ P(A|G, v)$.

3.2 Diffusion Model

To case the modeling, the widely-used Linear-Threshold (LT) model will be introduced to describe the information diffusion. In this model, we have an extra influence weight matrix B to evaluate the importance of edges. In particular, for a given edge (u, v), it corresponds to a weight $B_{u,v}$ such that $\sum_{u \in N^{in}(v)} B_{u,v} \leq 1$ where the $N^{in}(v)$ is the set of in-neighbors of v. The dynamic of information propagation in the LT model unfolds as follows.

A source node $s \in V$ is assumed to be the initial disseminator of the information. Denote by S_t the nodes activated in step t ($t = 0, 1, 2, \ldots$) and $S_0 = s$. Initially, each node u selects a threshold θ_u in range $[0, 1]$ uniformly at random. At step $t > 0$, an inactive node u would be activated if $\sum_{w \in N^{in}(u) \cap (\bigcup_{i<t} S_i)} B_{w,u} \geq \theta_u$. The process terminates when $S_t = \emptyset$.

Kempe [9] proved that the LT model is equal to the live-edge graph processes. In the LT model, a live-edge graph instance can be obtained by following rules. Each node v picks one incoming edge with the probability of $\sum_{u \in N^{in}(v)} B_{u,v}$. The selected edges are called live and the others are called dead. For a given live-edge graph denoted by g, each node i reachable from source s is active.

3.3 Basic Solution

We use $R(g, v)$ to denote a set of nodes that are reachable from v in g. Thus,

$$\begin{aligned} P(A|G, v) &= \sum_{g \subseteq G} [\Upsilon_G(g) I(A = R(g, v))], \\ &= \mathbb{E}_{g \sim G}[I(A = R(g, v))] \end{aligned} \tag{1}$$

where I is an indicator function defined as:

$$I(c) = \begin{cases} 1 \text{ if } c \text{ is true;} \\ 0 \text{ otherwise;} \end{cases}$$

and $\Upsilon_G(g)$ denotes the probability distribution of G, i.e.,

$$\Upsilon_G(g) = \prod_{v \in g \bigwedge (u,v) \in g} B_{u,v} \prod_{v \in g \bigwedge N_g^{in}(v) = \emptyset} \vartheta_v, \tag{2}$$

where $N_g^{in}(v)$ denotes the set of in-neighbors of v in g and $\vartheta_v = \sum_{u \in N_G^{in}(v)} B_{u,v}$. Nevertheless, the evaluation of $P(A|G, v)$ is computationally prohibitive since it is related to counting the number of linear extensions of a ordered node set. A trivial method-Monte Carlo Method(MC) could be used to estimate $P(A|G, v)$ by drawing graph $g \sim G$. However, due to the often exponential small number $P(A|G, v)$, the running time of MC is not acceptable. To show this, consider a linear graph $G(V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ and $E = \{(v_i, v_{i+1})|1 \leq i < n\}$. Suppose that $V = V'$ and $B_{v_i,v_{i+1}} = 1/2$, then we have $P(A|G, v_1) = 1/2^{n-1}$. We call a sampled graph positive sample, if there exist a node $v \in A$ could activate A having $R(g, v) = A$. To overcome the above problem, an ideal method is to make that all the output samples positive samples. In the next section, we will achieve this with a Markov Chain Monte Carlo (MCMC) method.

4 Solving Source Detection with MCMC

In this section, we explore the MCMC method for source detection.

We define a set of positive samples as

$$G' = \{g|\exists v,\ s.t.\ I(R(g, v) = A)\}. \tag{3}$$

and the distribution of G' as

$$\Upsilon_{G'}(g) = \begin{cases} \Upsilon_G(g)/Z & \text{g } \in G'; \\ 0 & \text{otherwise;} \end{cases}$$

where $Z = \sum_{g \in G'} \Upsilon_G(g)$

In MCMC, we sample in G' rather than G. The key property that makes MCMC approach work well is that $\hat{s} = arg\ max_v \mathbb{E}_{g \sim G'}[I(R(g, v) = A)]$. Now, we sample in G' with the MCMC approach.

First, we need to generate an instance of live-edge graph $g_1 \in G'$. We randomly pick a node in A and do the BFS in $G(A)$, where $G(A)$ is the infected subgraph induced by node set A and their inter edges. Till the observed nodes set in BFS process is A, we get a live-edge graph g. Then, by adding the dead-edges we expand the corresponding live-edge graph g in $G(A)$ to the graph g_1 in G. At last, we get a graph $g_1 \in G'$.

Second, we use the Algorithm 1 to generate a Markov chain. Algorithm 1 gives a local move in G'. Each local move will change a node's in-live-edge from the previous subgraph g_i.

At last, we measure $I(R(g_i, v) = A)$ on each graph g_i and figure out the maximum one as the source. Formally, we describe it in Algorithm 2. The following theorem demonstrates the correctness of the MCMC approach.

Theorem 1. *The Markov chain created by Algorithm 1 has a only stationary distribution and has* $\hat{s} = arg\ max_v \mathbb{E}_{g_i \sim G'}[I(R(g_i, v) = A)]$.

Proof. To prove Theorem 1, we give the following two lemmas first.

Algorithm 1. Local move

Require: G, A, g_i.
Choose a node $v \in V$ uniformly randomly; $g_{i+1} = g_i$;
Let $\sum_{w \in N^{in}(v)} B_{w,v} = \vartheta(v)$;
Delete the in-live-edge of node v in g_{i+1};
In g_{i+1}, pick v's in-live-edge (u, v) with probability $B_{u,v}$;
if $g_{i+1} \notin G'$ **then**
  $g_{i+1} = g_i$;
end if
Ensure: g_{i+1}.

Algorithm 2. MCMC for source detection

Require: G, A, g_1, parameter K;
Create new array *count* with size $|A|$;
$k = 0$, $g = g_1$;
while $k < K$ **do**
  $T = \{v | R(v, g) = A\}$;
  for $v \in T$ **do**
    $count[v] = count[v] + 1$;
  end for
  Local move(G, A, g);
end while
$s = arg\ max_v count[v]$;
Ensure: s.

Lemma 1. *The Markov chain created by Algorithm 1 is irreducible.*

Proof. Given two instance of state $g_i, g_j \in G'$, we set the g_i as the input in Algorithm 1 and we want to prove that there exist a sequence that contains both g_i and g_j. By the local move method, we establish the sequence as follows.

First, for each edge $e \in g_i \bigwedge e \notin G(A)$, we can change the edge into the dead edge by a local move. Also, we do the same operation to g_j. Thus, without generality, suppose that all the edges in g_i and g_j are also the edges in $G(A)$.

Second, we give the order of changing the node's edges in g_i by following method. Since $g_i \in G'$, we can find out a node v which could activated A in g_i and mark it as 1. Then, we mark the node u that can reach v via a live edge in g_j as 2. We do above process repeatedly, till the present node is marked or the present node cannot be reached from other nodes in g_j. Next, we mark the left nodes by the pre-order traversal in g_j.

At last, we do the local move in g_i by the marked order. Each local move will make a node's in-live-edge in g_i be the same as g_j. Figure 1 shows an example of above process. It is obvious that each intermediate graph belongs to G'. Lemma follows.

Lemma 2. *The Markov chain is aperiodic.*

Proof. Given state g_i, for any possible next state g_{i+1}, the probability of $P(g_{i+1} = g_i) \neq 0$. Thus the Markov chain is aperiodic.

Since the Markov Chain is irreducible and aperiodic, it only has a stationary distribution. Then, we have Lemma 3.

Lemma 3. *In Algorithm 1, the transition probability of graph g_i and g_{i+1} satisfy with*

$$\frac{P(g_{i+1}|g_i)}{P(g_i|g_{i+1})} = \frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)}, \tag{4}$$

where $P(g_{i+1}|g_i)$ is the probability of transiting from state g_i to next possible state g_{i+1}.

Proof. Suppose that the different live-edges in g_i and g_{i+1} are (u_1, v) and (u_2, v). There are three different cases.

Case 1: $(u_1, v) \in g_i$ and $(u_2, v) \in g_{i+1}$. In this case, we have $\frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)} = \frac{B_{u_2,v}}{B_{u_1,v}}$. From Algorithm 1, $\frac{P(g_{i+1}|g_i)}{P(g_i|g_{i+1})} = \frac{B_{u_2,v}/n}{B_{u_1,v}/n} = \frac{B_{u_2,v}}{B_{u_1,v}}$.

Case 2: $(u_1, v) \in g_i$ and $(u_2, v) \notin g_{i+1}$. In this case, $\frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)} = \frac{1-\vartheta(v)}{B_{u_1,v}}$. From Algorithm 1, $\frac{P(g_{i+1}|g_i)}{P(g_i|g_{i+1})} = \frac{(1-\vartheta(v))/n}{B_{u_1,v}/n} = \frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)}$.

Case 3: $(u_1, v) \notin g_i$ and $(u_2, v) \in g_{i+1}$. Similar to case 2, $\frac{P(g_{i+1}|g_i)}{P(g_i|g_{i+1})} = \frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)}$. Thus, $\frac{P(g_{i+1}|g_i)}{P(g_i|g_{i+1})} = \frac{\Upsilon_{G'}(g_{i+1})}{\Upsilon_{G'}(g_i)}$.

From Lemma 3, π is the stationary distribution. From the definition of G', obviously, $\hat{s} = arg\ max_v \mathbb{E}_{g\sim G'}[I(R(v, g) = A]$. Theorem follows.

5 Sampling in Infected Subgraph

In this section, to improve the efficiency of MCMC, we simplify the sampling method that scales according to observed graph size rather than the size of the entire graph.

When an infected subgraph is obtained at the end of the diffusion process, the snapshot of the process is observed only at $G(A)$. We constrain information diffusion to $G(A)$. In this case, let $P(A|G(A), v)$ be the probability that the infected node set we observe in $G(A)$ is A. We arrange the elements in set $\{P(A|G(A), v)|v \in A\}$ by descending order and let $Rank[P(A|G(A), v)]$ be the rank of $P(A|G(A), v)$ in $\{P(A|G(A), v)|v \in A\}$. We obtain the following theorem:

Theorem 2. $Rank[P(A|G, v)] = Rank[P(A|G(A), v)]$.

Proof. By setting the nodes in A that are initially activated, let β be the probability that no node in $V \setminus A$ has been activated. Since the snapshot is observed when the information diffusion process terminated, each node in A try to activate

the node in $V \setminus A$. Thus, $P(A|G,v) = \beta P(A|G(A),v)$. We normalize $P(A|G,v)$ to $N(A|G,v) = \frac{P(A|G,v)}{\sum_{u \in A} P(A|G,v)}$. Then,

$$N(A|G,v) = \frac{P(A|G,v)}{\sum_v P(A|G,v)} = \frac{\beta P(A|G(A),v)}{\sum_v \beta P(A|G(A),v)} = N(A|G(A),v).$$

Since $Rank[P(A|G,v)] = Rank[N(A|G,v)]$, $Rank[P(A|G(A),v)] = Rank[N(A|G(A),v)]$ and $N(A|G,v) = N(A|G(A),v)$. Thus $Rank[P(A|G,v)] = Rank[P(A|G(A),v)]$.

This means we can simplify the sampling strategy only in the infected subgraph.

6 Experiment

In this section, we conduct experiments of our source detection algorithm on a real network dataset. Our experiments are conducted on a machine with an Intel i5 CPU and 16 GB of memory. All experiments are implemented in JAVA.

6.1 Experimental Setting

Data Sets. We conduct our experiments on a real network dataset: **Wiki-Vote** [12]. This dataset is composed of all Wikipedia voting data from the inception of Wikipedia till January 2008. And it contains 7115 nodes and 103,689 directed edges.

In our experiments, the probability of the propagation of an edge (u, v) is set to $\frac{\alpha}{indegree(v)}$, where the $indegree(v)$ is the set of in-neighbors of v. To make our experiments more compelling, we set the value of α to 0.5, 0.75 and 1 respectively. We randomly choose a node as source and run the LT model until no more nodes are infected, then we obtain the set of these infected nodes. We repeat the process 1,000 times to get 1,000 random datasets. In order to challenge the effectiveness of our algorithm, we guarantee that each dataset contains more than 5 infected nodes and contains at least two candidate sources whose likelihood to infect others is larger than 0.

Algorithms and Their Explanations. To compare the proposed algorithm with existing algorithms, we also implement some other algorithms. The algorithms used in the experiments are as follows:

- JC: Jordan center [23]. This selects an activated node with the maximal distance to the others as source.
- Ef: A heuristic algorithm proposed in [11]. In [11], it proposes the "DP", "Sort", "OutDegree" algorithms to find the k sources. However, to the single source, "DP=Sort". In the experiments, we use this algorithm and call it Ef.
- $MCMC$: the method proposed in Sect. 4.

Evaluation Index. We apply the following widely used measures to evaluate the effectiveness of our methods.

- *Detection Rate*. Detection rate is the probability that the node identified by the algorithm is the actual source.
- $\gamma - accuracy$. γ-accuracy is the probability that the actual source ranked among top γ.
- *Error Distance*. Error Distance is defined as the distance between detected source node and true source node assuming edges are undirected.

6.2 Effectiveness Validation

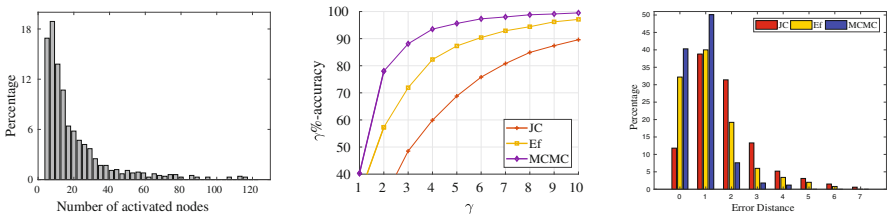


Fig. 2. When α is equal to 1.0. Statistics histogram of activated subgraph used in experiments (a) The $\gamma - accuracy$ result (b) and the error distance (c) of different algorithms.

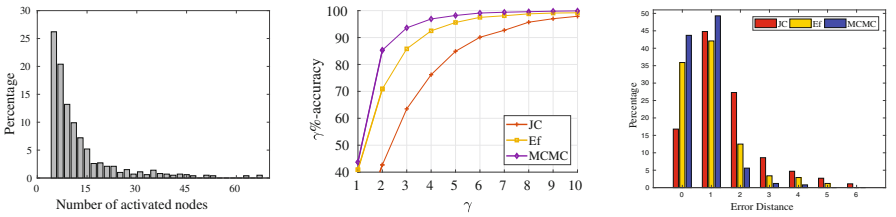
Comparison of Results. Figure 2(b) shows the $\gamma - accuracy$ computed by each method when α is set to 1. Note that the intersection of the polyline and the y-axis indicates the detection rate. To some degree, detection rate is on behalf of the precision of a method. The higher the value is, the better performance the algorithm achieves. Obviously, the $MCMC$ method yields the best performance. In more than 40% of the total experiments, the $MCMC$ method can find the true source. The detection rate of the $MCMC$ method is 20% higher than that of the Ef method and even two times more than that of the JC method.

For the $\gamma - accuracy$, we make our algorithm output a list of candidate source nodes sorted in descending order of likelihood, the value of γ represents the index of actual node in the list which is no more than γ. According to Fig. 2(b), it is clear that when the value of γ is fixed, the accuracy of the $MCMC$ method is higher than that of the other two methods. In more than 90% of the total experiments, the rank of the actual source is among top 3 in the list of candidates which is produced by the $MCMC$ method. In about 75% of the total experiments, the Ef method can make sure the rank of the actual source is among top 3. In addition, the JC method can satisfy the same condition in no more than 50% of the total experiments.

Figure 2(c) shows the distribution of error distances when α is set to 1. Error distance reflects how far the detected nodes is away from the actual source. Note that the larger of the error distance is, the worse performance the corresponding method achieves. It is clear that all source nodes identified by the proposed algorithm are four hops around the source node. In more than 90% of the total experiments, the detected nodes detected by the $MCMC$ method are two hops around the source node. At the same time, the Ef method and JC method not only have fewer results with 0 or 1 error distance, but have heavier tail as well.

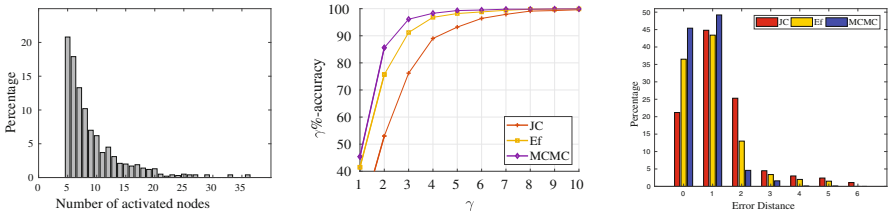
In comparison, the $MCMC$ method always yields the best performance, and the Ef method is slightly worse than the $MCMC$ method. Because the idea of the Ef method is to maximize the similarity between estimated and observed diffusion graph which is different from the fundamental meaning of source detection. Contrarily, the JC method is obviously worse than the $MCMC$ method. The JC method just uses the structure of the social network instead of considering the probability of each node to be the source. This is why the JC method always performs the worst.

Parameter Analysis. Figures 2, 3 and 4 show the performance of each method based on the different value of α. According to the result, we find that all methods are sensitive to the parameter. With the increment of the value of α, the probability of an inactive node be activated becomes higher, so that there are more active nodes. However, the performance of each method becomes worse. Because the difficulty of the source detection problem is related to the number of activated nodes. Even so, the performance of the $MCMC$ method just drops a little, it always holds on to the number-one spot among the three methods. At the same time, the gap between the $MCMC$ method and the other two methods gets larger and larger. To some extent, the $MCMC$ method is more stable than the other in large network.



(a) (b) (c)

Fig. 3. When α is equal to 0.75. Statistics histogram of activated subgraph used in experiments (a) The $\gamma - accuracy$ result (b) and the error distance (c) of different algorithms.



(a) (b) (c)

Fig. 4. When α is equal to 0.5. Statistics histogram of activated subgraph used in experiments (a) The $\gamma - accuracy$ result (b) and the error distance (c) of different algorithms.

7 Conclusion

In this paper, we address the source detection problem in the LT model, and then derive a MCMC approach to improve the accuracy for general graphs. To further improve the efficiency, we reduce the sample space by sampling on the observed subgraph rather than the entire graph. Experiments on real social network demonstrate the performance of our approach.

In future, parallelizing our method in source detection problem based on other diffusion models is an interesting direction of our work. And the efficiency of our method can be further improved. At the same time, our work just concentrates on the single source detection problem, expanding our method for multiple source detection problem is a meaningful work.

Acknowledgments. This research was partially supported by grants from the National Natural Science Foundation of China (Grant No. U1605251) and the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010). Also, this research was supported by the Anhui Provincial Natural Science Foundation (Grant No. 1708085QF140), and the Fundamental Research Funds for the Central Universities (Grant No. WK2150110006).

References

1. Altarelli, F., Braunstein, A., DallAsta, L., Ingrosso, A., Zecchina, R.: The patient-zero problem with noisy observations. J. Stat. Mech. **2014**, P10016 (2014)
2. Altarelli, F., Braunstein, A., DallAsta, L., Lage-Castellanos, A., Zecchina, R.: Bayesian inference of epidemics on networks via belief propagation. Phys. Rev. Lett. **112**, 118701 (2014)
3. Chang, B., Zhu, F., Chen, E., Liu, Q.: Information source detection via maximum a posteriori estimation. In: ICDM 2015, pp. 21–30. IEEE Press, Atlantic City (2015)
4. Dong, W., Zhang, W., Tan, C.W.: Rooting out the rumor culprit from suspects. In: ISIT 2013, pp. 2671–2675. IEEE Press, Turkey (2013)
5. Fanti, G., Kairouz, P., Oh, S., Viswanath, P.: Spy vs. spy: rumor source obfuscation. In: SIGMETRICS 2015, pp. 271–284. ACM Press, Portland (2015)

6. Fioriti, V., Chinnici, M.: Predicting the sources of an outbreak with a spectral technique. arXiv preprint arXiv:1211.2333 (2012)
7. Jain, A., Borkar, V., Garg, D.: Fast rumor source identification via random walks. Soc. Netw. Anal. Min. **6**, 62 (2016)
8. Kelner, J.A., Madry, A.: Faster generation of random spanning trees. In: 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 13–21. IEEE Press, Washington (2009)
9. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: SIGKDD 2013, pp. 137–146. ACM Press, Chicago (2003)
10. Kermack, W.O., McKendrick, A.G.: Contributions to the mathematical theory of epidemics. II. The problem of endemicity. Proc. Roy. Soc. London A Math. Phys. Eng. Sci. **138**, 55–83 (1932)
11. Lappas, T., Terzi, E., Gunopulos, D., Mannila, H.: Finding effectors in social networks. In: SIGKDD 2010, pp. 1059–1068. ACM Press, Washington (2010)
12. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: WWW 2010, pp. 641–650. Raleigh (2010)
13. Nguyen, H.T., Ghosh, P., Mayo, M.L., Dinh, T.N.: Multiple infection sources identification with provable guarantees. In: CIKM 2016, pp. 1663–1672. ACM Press. Indianapolis (2016)
14. Schoot, R., Kaplan, D., Denissen, J., Asendorpf, J.B., Neyer, F.J., Aken, M.A.: A gentle introduction to Bayesian analysis: applications to developmental research. Child Dev. **85**, 842–860 (2014)
15. Shah, D., Zaman, T.: Detecting sources of computer viruses in networks: theory and experiment. In: SIGMETRICS 2010, vol. 38, pp. 203–214 (2010)
16. Shah, D., Zaman, T.: Rumor centrality: a universal source detector. In: SIGMETRICS 2012, pp. 199–210. ACM Press, London (2012)
17. Shah, D., Zaman, T.: Rumors in a network: who's the culprit? TIT **2011**(57), 5163–5181 (2011)
18. Tong, G., Wu, W., Guo, L., Li, D., Liu, C., Liu, B., Du, D.-Z.: An Efficient Randomized Algorithm for Rumor Blocking in Online Social Networks. arXiv preprint arXiv:1701.02368 (2017)
19. Wang, Z., Dong, W., Zhang, W., Tan, C.W.: Rumor source detection with multiple observations: fundamental limits and algorithms. In: SIGMETRICS 2014, pp. 1–13. ACM Press, Austin (2014)
20. Zhai, X., Wu, W., Xu, W.: Cascade source inference in networks: a Markov chain Monte Carlo approach. Comput. Soc. Netw. **2**, 17 (2015)
21. Yang, Y., Chen, E., Liu, Q., Xiang, B., Xu, T., Shad, S.A.: On approximation of real-world influence spread. In: ECML-PKDD 2012, pp. 548–564. UK (2012)
22. Xu, T., Zhong, H., Zhu, H., Xiong, H., Chen, E., Liu, G.: Exploring the impact of dynamic mutual influence on social event participation. In: SDM 2015, pp. 262–270. Canada (2015)
23. Jordan, C.: Sur les assemblages de lignes. J. Reine Angew. Math. **70**, 81 (1869)