An observer deployment algorithm for locating the diffusion source timely in social network

Yubo Zhang[1, a], Xizhe Zhang[2,b], Bin Zhang[3,c]

[1,2,3]School of Computer Science and Engineering, Northeastern University, Shenyang, China

[a,c]yubo_zh@163.com, [b]zhangxizhe@ise.neu.edu.cn

Keywords: Observer; Information diffusion; Locate the source; Online social network; Timely

Abstract. Locating the source of information diffusion on the social network is a challenging and significant task. It is important to control the spread of wrong information on the social network. An existing method is to observe a few nodes in the network and estimate the location of the source from the data recorded by the observed nodes. The locating effect depends on the positions of the observed nodes. In this paper, we try to find a way to choose the observers that can locate the source earlier. Then we provide a timely observer deployment algorithm to solve the problem. Its basic idea is to use the average distance between every non-observer node and its nearest observer to measure the timeliness of an observers set. We carry out simulation experiments on model networks, the results show that the observers chosen by the algorithm can locate the source more timely.

Introduction

In recent years, spreading of information through online social networks is ubiquitous [1]. For example, you can discuss the hot topics on micro blog or promote your products through the Internet. But there are also some wrong information, such as computer virus and rumors [2, 3]. An approved way to control the spread of the wrong information is locating the information source timely and accurately [4, 5]. For the huge scale of the online social network, to locate the source by the data recorded by a few observed nodes (this kind of node is called observer) is a feasible method [5, 6, 7]. And the position of the observers in the network has an impact on the performance of the locating method.

Prior works on the deployment of observers are mainly about the way to get the highest accuracy of the locating result [8, 9]. However, we know that the sooner we locate the source the less affected by the wrong information. So there is another challenge is to find an observer deployment method to discover the rumor diffusion and locate the source timely.

Information diffusion model and locating method

Information diffusion model

In this paper, the social network is modeled by an undirected graph G={V, E}, V is nodes set, E is edges set. We consider the situation that G is known and there is only one information source in the network. The diffusion model is modeled as Figure 1.

In Figure 1(a), the weights on the edges are the diffusion delay (the time of the information go through the edge), o_1 and o_2 are observers, s is the source, n_1 is a usual non-observer. Figure 1(b) shows that a diffusion process starts from s at an unknown time t_s. At the time t_s , s sends an information m to all its neighbor nodes, when any node first received m, it will send m to all its neighbor nodes except the incoming node (the node which send m to it). The information goes along the weighted shortest paths. The observer needs to record its receiving time and incoming node, but non-observer does not.

As shown in Figure 1(b), the diffusion delay of m goes from s to o_1 through the path s->o_1 is 7, and the path s->n_1->o_1 is 4+2=6. Because the information goes along the weighted shortest paths, observer o_1 received m form n_1, not s. So o_1 recorded its incoming node is n_1 and receiving time is t_1. Here t_1= t_s+6, because t_s is unknown, t_1 is also unknown. And the information m didn't go through the edge between s and o_1.

Locating method

According to the diffusion data recorded by observers, we compute the maximum likelihood estimation to locate the information source. The specific calculation process can be shown as follows:

Step 1, get the spread record data of observers that has received the message;

Step 2, select a non-observer node s as the root, generating a breadth-first spanning tree;

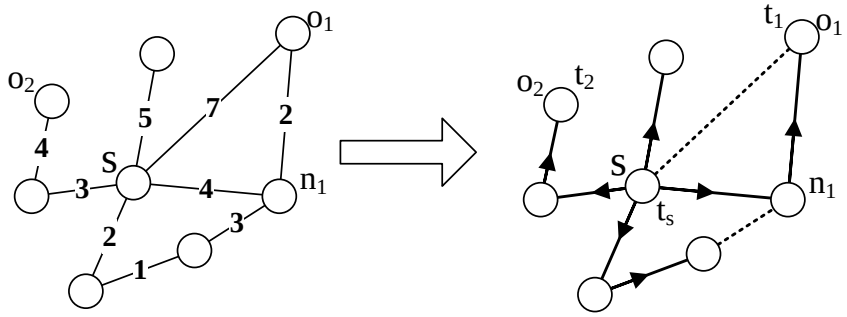
Step 3, assume s as the expected source, computing maximum likelihood estimation $\hat{s}$ in the generating tree with eq. 1 and eq. 2;

$$\hat{s} = \frac{\exp\left(-\frac{1}{2}(d-\mu_s)^T \Lambda_s^{-1}(d-\mu_s)\right)}{\sqrt{|\Lambda_s|}}. \tag{1}$$

$$[\Lambda_s]_{k,i} = \sigma^2 \cdot \begin{cases} |p(o_1, o_{k+1})| & k = i \\ |p(o_1, o_{k+1}) \cap p(o_1, o_{i+1})| & k \neq i \end{cases}. \tag{2}$$

Where, $[d]_k = t_{k+1} - t_k$, t_k denote the time observation node o_k recorded, $|p(u,v)|$ indicates the number of edges of the shortest path between node u and node v, μ and σ are the mean and variance of the diffusion delay of edges in the network.

Step 4, compute the maximum likelihood estimation $\hat{s}$ of every non-observer node, the node with the max $\hat{s}$ is the information source.



(a) (b)

Information go through the edge

Information do not go through the edge

Fig. 1 Information diffusion process

Timely observer deployment algorithm

The timeliness of a set of observers is the time delay from the diffusion beginning to locate the source. When there is one observer has received the information, the locating process will start. So the distance between the first received observer and the source is shorter, the timeliness of this set of observers is higher. In view of every node is a potential source in social network, it can be concluded that the average distance between every node and its nearest observer in the network can be used to measure the timeliness of a set of observers. Because we can't get diffusion delay of every edge, we use hops number of the shortest path between two nodes as their distance. Based on this conclusion, we design a timely observer deployment algorithm (TODA). The algorithm is aim at choosing a set of observers with the min measure. The algorithm is based on the genetic algorithm [10], and its constraint condition and objective function are eq. 3 and eq. 4.

$$\text{Constraint condition:} \begin{cases} \sum_{i=1}^{N} x_i \leq k \\ x_i \in \{0,1\}, (i = 1,2,\cdots N) \end{cases}. \tag{3}$$

Objective function: $\min f(x_1, x_2, \cdots x_N) = \min$ average nearest distance. (4)

Here, vector (x_1, x_2, …, x_N) is a chromosome, gene x_i is the state that whether the node o_i is selected be an observer, 1 is yes, 0 is no; k is the scale of observer set; N is the number of nodes in the social network. The algorithm can be expressed as table 1. There, MG is the max generation, g(i) is the population of generation i, M is the population size.

Table 1 Timely observer deployment algorithm

Timely observer deployment algorithm

begin
t=1;
initialize g(1);
evaluate g(1);
if t<=MG;
t+1;
evaluate g(t);
select fathers from g(t) to g(t+1);
if g(t+1).size<M;
crossover fathers to g(t+1);
mutation fathers to g(t+1);
end;
end;
end;

In the algorithm, there are five main operations on the chromosomes in the genetic process.

(1) Initialize. Generate M chromosomes as the first generation g(1), assign 0 or 1 to chromosome genes randomly.

(2) Evaluate. Computer the evaluation of each chromosome in the generation, the evaluation function is eq. 5.

$$f(x_1, x_2, \cdots, x_N) = \frac{\sum_{i \in V} \min \left\{ \left| p(v_i, o_j) \right| \right\}_{j=1}^{j=k}}{N}. \quad (5)$$

(3) Select. Select the two highest evaluation chromosomes from the last generation, and copy them to the next generation.

(4) Crossover. As shown in figure 2, in two father chromosomes, to retain some of their genes, and exchange the remaining part, get two new chromosomes into the next generation.

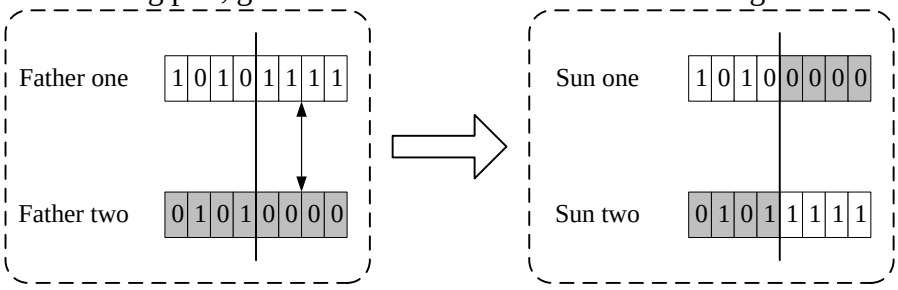


Fig. 2 Chromosomes crossover process

(5) Mutation. As shown in figure 3, in one father chromosomes, to choose one gene randomly, change it into its opposite value, get one new chromosomes into the next generation.

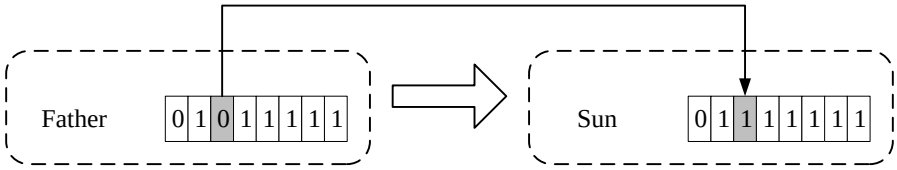


Fig. 3 Chromosomes mutation process

Simulation experiment

To validate our algorithm, we do simulation experiment on the typical model networks, ER model [11] and BA model [12]. The node degree distributions of these two models are very different even with the same nodes and edges. The former is Gaussian distribution, and latter is exponential distribution. We generate two networks with different network density by each model. Detailed experimental data is in table 2. Here, N is the number of nodes in the network; L is the number of edges; AD is average degree of the nodes; AP is average path between any two nodes; ND is network diameter, the longest path between any two nodes[13, 14]. AD usually used to describe the network density, AP and ND usually used to describe the network topology.

Table 2 Model network data

Name	N	L	AD	AP	ND
BA1	1962	6000	6.11	4.26	8
BA2	1892	3998	4.23	5.16	11
ER1	1996	6074	6.08	4.41	8
ER2	1962	4020	4.09	5.55	11

In order to show performance of TODA algorithm, we do comparison experiment with two typical observer deployment method [5], first choose the node with high degree [15] and choose it randomly. On each model network, we compute the average nearest distance with the observer proportion from 1% to 10%, gradually accumulate 0.5%. For degree, we choose the nodes with the highest degree in the given proportion as observers, and compute the average nearest distance of other non-observers. For random, we choose nodes in the given proportion randomly 100 times, and compute the average of average nearest distance of these 100 observers sets. For TODA, we set the parameter as follow, the max generation is 1000 and the population size is 500. After running TODA, we get the observers set and its average nearest distance.

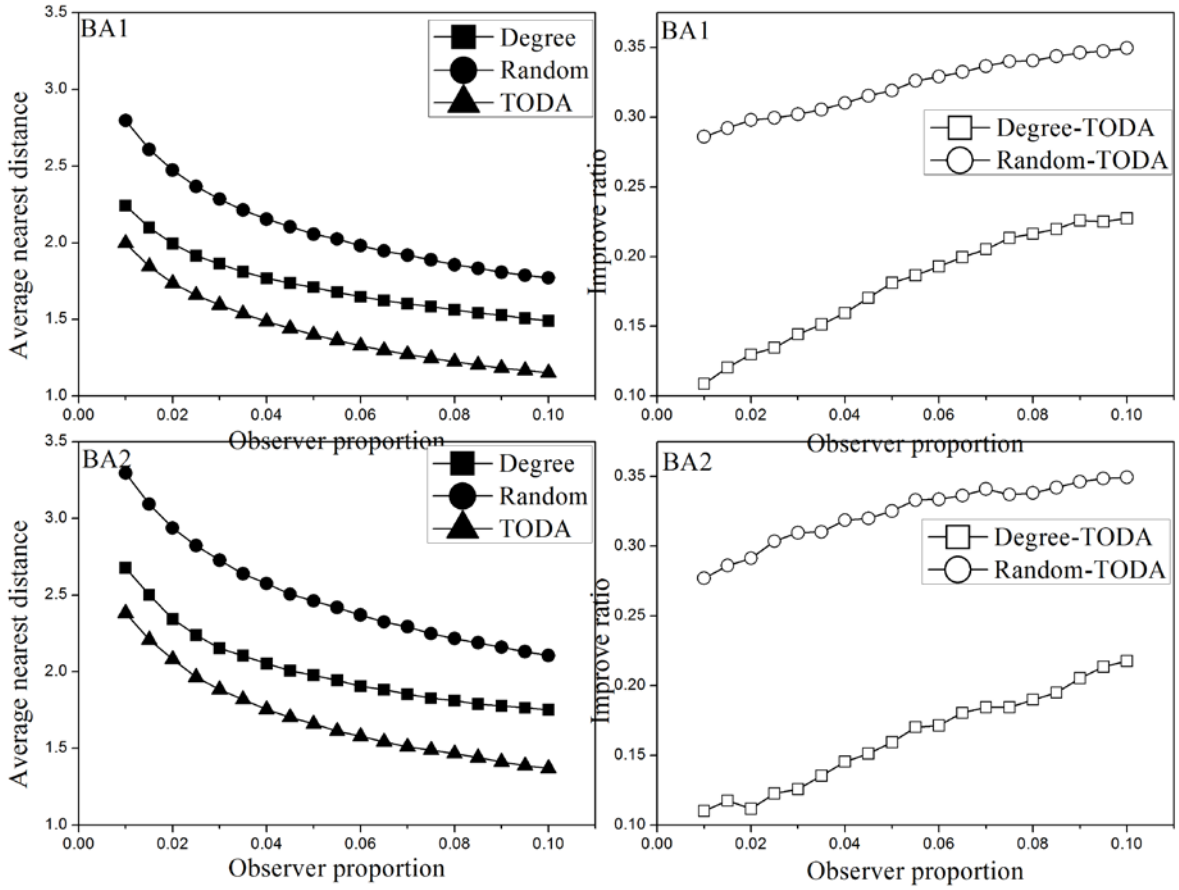


Fig. 4 Simulation experiment result on BA network

Furthermore, In order to verify the effect of TODA, we compute the improve ratio between TODA and other two methods as eq. 6 and eq. 7, in these two equations, Ir is the improve ratio between the two specified methods, And is the average nearest distance of the specified method. All the experiment results are shown in figure 4 and figure 5, figure 4 is the result of BA networks and figure 5 is the result of ER networks.

$$Ir_{Degree-TODA} = \frac{And_{Degree} - And_{TODA}}{And_{Degree}} \tag{6}$$

$$Ir_{Random-TODA} = \frac{And_{Random} - And_{TODA}}{And_{Random}} \tag{7}$$

An obvious result is shown in figure 4, on BA model networks, the average nearest distance of each observer deployment on each network is decreased with the increase of observer proportion. On each observer proportion, the average nearest distance of the observers set chosen by TODA is shorter than it chosen by random or degree. In both BA model networks, at the observer proportion 1%, the improve ratio between degree and TODA can achieve nearly 12%, and the improve ratio between random and TODA can achieve nearly 27%, at the observer proportion 10%, the improve ratio can achieve 22% and 34%. Furthermore, the improve ratio is increased with the increase of observer proportion. That is, the observers selected by TODA have a better performance than them selected by degree and random. So we can discover earlier by observers chosen by TODA after the wrong information diffusion beginning.

The experiment result of ER model networks are shown as figure 5, and we get a similar result, the performance of TODA is better than degree and random, too. We can get the conclusion that the TODA is an effective observer deployment method on timeliness. We can discover the wrong information diffusion and locate the information source earlier by the observers chosen by TODA, and the performance will be better with more observers in the network.

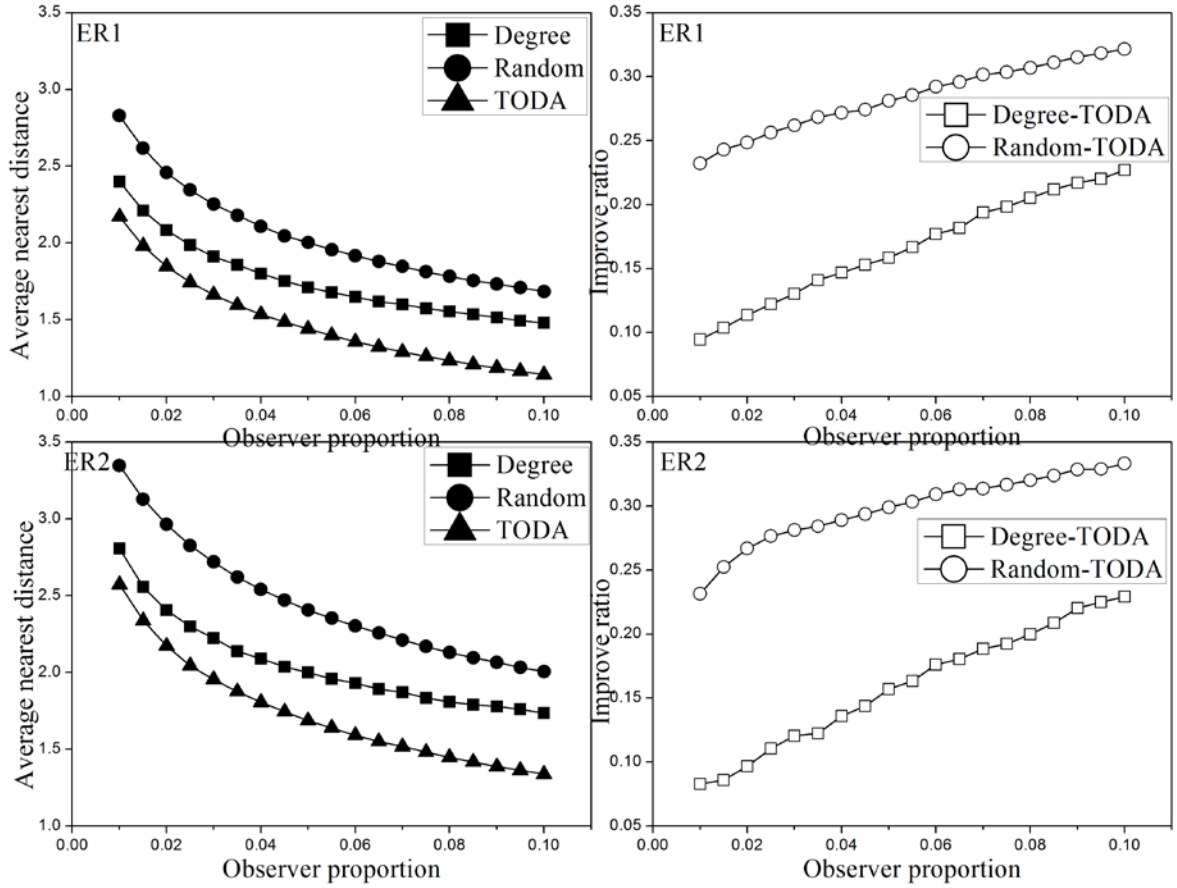


Fig. 5 Simulation experiment result on ER network

Summary

Locating the information source timely is important to control wrong information diffusion on social network. In order to improve the timeliness of locating source method based on deploying observers, in this paper, we pay attention on observer deployment method and provide a timely observer deployment algorithm to choose observers set to locate source timely. In simulation experiment, we select both ER and BA model to generate experimental networks, compared with existing observer deployment method, choose high degree nodes first be and choose randomly, the result shows our method can improve the timeless more obviously.

References

[1] Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network[C]// ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2003:137-146.

[2] Dong W, Zhang W, Tan C W. Rooting out the rumor culprit from suspects[C]// IEEE International Symposium on Information Theory. IEEE, 2013:2671-2675.

[3] Shah D, Zaman T. Rumors in a Network: Who's the Culprit?[J]. Information Theory IEEE Transactions on, 2011, 57(8):5163-5181.

[4] Shah D, Zaman T. Detecting sources of computer viruses in networks: theory and experiment[J]. Acm Sigmetrics Performance Evaluation Review, 2010, 38(1):203-214.

[5] Pinto P C, Patrick T, Martin V. Locating the source of diffusion in large-scale networks.[J]. Phys.rev.lett, 2012, 109(6):1-5.

[6] Prakash, B.A, Vreeken J, Faloutsos C. Spotting Culprits in Epidemics: How Many and Which Ones?[C]// IEEE International Conference on Data Mining. IEEE, 2012:11-20.

[7] Zhu K, Chen Z, Ying L. Locating the contagion source in networks with partial timestamps[J]. Data Mining & Knowledge Discovery, 2015:1-32.

[8] Zhang Y B, Zhang X Z, Zhang B. Observer deployment method for locating the information source in social network[J]. Journal of Software, 2014, 25(12), 2837-2851.

[9] Zhang X, Zhang Y, Lv T, et al. Identification of efficient observers for locating spreading source in complex networks[J]. Physica A Statistical Mechanics & Its Applications, 2016(442), 100-109.

[10] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique[J]. Pattern Recognition, 2000, 33(9):1455-1465.

[11] Erdős, P, Rényi, A. On the evolution of random graphs[J]. Publication of the Mathematical Institute of the Hungarian Academy Ofences, 1960, 38(1):17--61.

[12] Barabasi A L, Albert R. Emergence of Scaling in Random Networks[J]. Science, 1999, 286(5439):509-512.

[13] Dipl.-Math. Oec. A L, Friedl D M B, Heidemann J. A Critical Review of Centrality Measures in Social Networks[J]. Business & Information Systems Engineering, 2010, 2(6):371-385.

[14] Ernesto E, Rodríguez-Velázquez J A. Subgraph centrality in complex networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 71(5):122-133.

[15] Freeman L C. Centrality in social networks conceptual clarification[J]. Social Networks, 1978, 1(3):215-239.