Topic-aware multiple source locating with partial observation

Jiefeng Gao, Shixiang Feng

May 27, 2018

Abstract

In this report, we come up with a topic-aware source-locating method that can locate multiple sources in network with partial observation based on topic. Firstly we decide infection probability based on item topic and user interests. Then we apply Optimal Jordan Cover(OJC) algorithm to sorted out edges with low infection probability. Finally with partial observations we locate multiple diffusion sources in networks. Experiments on two different networks based on reality data have shown that our method has certain capability to locate diffusion sources.

1 Introduction

Locating the sources of observed information in social network has attracted much attention in recent years, which is crucial for controlling and preventing rumor risks. But existing work on diffusion source locating merely relies on network structure and analysis and doesn't fully use the topic information which is an important component for source locating. Meanwhile, there are some excellent algorithm about source-locating in abstract network that not actually consider topic information. The combination of both is promising and natural. In this report, we will create two network based on citation data and twitter data. We can learn some property from network: user's interests for different topics and infection probability between two users.Then we will apply OJC algorithm to see the source-locating effect.

2 Topic-aware SI Model

Different from the common SI model, the infected probability on each edge would change to according to different topics in the topic-aware SI model. [1] For each edge $(u, v) \in E$ and each item i, $p^i_{u,v}$ represents the infected probability between user u and user v on item i. More specifically, for each item i spreading on network, we view it as a topic distribution $\gamma^z_i = p(z|i)$ for each topic $z \in [1, K]$ with $\sum_{z=1}^{K} \gamma^z_i = 1$. And every user in the network has their different interests for each topic, let p^z_u denote the distribution of interests for each topic z and $\sum_{z=1}^{K} p^z_u = 1$. Then we construct the infected probability $p^i_{u,v}$ for item i based on twofold: 1)the interest similarity of neighbor users; 2)the user's enthusiasm for this item.

In the topic-aware SI model, each node in the network is in one of two states − Susceptible(S) or Infected(I). Once a node is infected, it will stay infected forever. With the diffused item i, an infected node u tries to infect each of its susceptible neighbors v independently with probability $p^i_{u,v}$, as shown in Eq.1,

$$p^i_{u,v} = \sum_z p(z|u, i)p^z_v \tag{1}$$

where $p(z|u, i)$ is the following logistic selection function

$$p(z|u, i) = \frac{exp(p^z_u + \gamma^z_i)}{1 + exp(p^z_u + \gamma^z_i)} \tag{2}$$

3 The Optimal Jordan Cover Algorithm

This algorithm refers to an AAAI paper.[2]

We assume the network is represented by an undirected graph g. Denote by $E(g)$ the set of edges and $V(g)$ the set of nodes in graph g. We further assume a heterogeneous SI model for diffusion. In this model, each node has two possible states: susceptible (S), infected (I). Time is slotted. At the beginning of each time slot, each infected node (say node u) attempts to infect its neighbor (say node v) with probability q_{uv} , independently across edges. We call q_{uv} the infection probability of edge (u, v). We call r_u the recovery probability. We further assume $q_{uv} \in (0, 1]$ for all edges $(u, v) \in E(g)$ and $r_v \in [0, 1]$ for all nodes $v \in V(g)$.

We assume the epidemic diffusion starts from m sources in the network. Denote by $s_1, s_2, ..., s_m$ the sources and S the set of sources, i.e., $S = s_1, s_2, ..., s_m$. We assume m is a independent constant. Finally, we assume that a partial snapshot of the network state at time slot t is given, with an unknown observation time t. In the snapshot, each infected node reports its state with probability $\theta_v \in (0, 1)$, independent of other nodes. If a node reports its state, we call it an observed node. Denote by I' the set of observed infected nodes. Based on I' , the source localization problem is to find S that solves the following maximum likelihood (ML) problem:

$$W* = \arg\min_{W \subset V(g)} Pr(S = W|I') \tag{3}$$

However, this is a difficult problem on non-tree networks. So instead of solving the ML problem, it's desirable to take asymptotic perfect detection.

3.1 Definition and Steps

The algorithm defines several conceptions:

Define hop-distance between a node v and a node set W to be the minimum hop-distance between node v and any node in W:

$$d(v, W) \triangleq \min_{u \in W} d(v, u) \tag{4}$$

Define the infection eccentricity of node set W to be the maximum hop-distance from an infected node in I' to set W:

$$e(W, I') = \max_{v \in I'} d(v, W) \tag{5}$$

Denote K to be the candidate set for sources and define m-Jordan-cover (m-JC) to be the set W*(K, I',m) such that:

$$W * (K, I', m) = \min_{W \in W||W|=m, W \subset K} e(W, I') \tag{6}$$

That is, m-JC is the set of m nodes in K with the minimum infection eccentricity.

With all these conceptions, we can give the steps of algorithm:

Step1:Candidate Selection:

1. Set a positive threshold Y.
2. Pick a node set K which includes Y infected observed nodes and several unobserved nodes.
3. Merge I' and K as $K+$
4. Calculate the induced graph g of $K+$
5. Calculate the connected subgraph of g as $g-$. If g is not connected,we select a random node in each component, randomly pick one selected node and add the shortest pathes from this node to all other selected nodes to form a connected $g-$.

Step2:Jordan Cover:

1. For any m combination of nodes in K, compute the infection eccentricity of node set as defined before on subgraph $g-$

It should be noted that the algorithm should satisfy some limitation. If we denote the average node degree as u, the lower bound on the infection probability as q and the lower bound on the reporting probability as θ, the limitaion is:

1. $uq\theta = \Omega(logn)$

2. $\limsup_{n->\infty} \frac{Y}{uq\theta} < 1$
3. $\liminf_{n->\infty} \frac{Y}{uq\theta} > 0$

Especially, condition 1 is the doctrine we use to filter the network so that OJC works properly. Edges with infection probability that lower than threshold will be discarded, and nodes without any edges also be discarded. As network structure may change every filtering, we repeat this process until no more edges are discarded.

4 Introduction of Data and Data Parsing

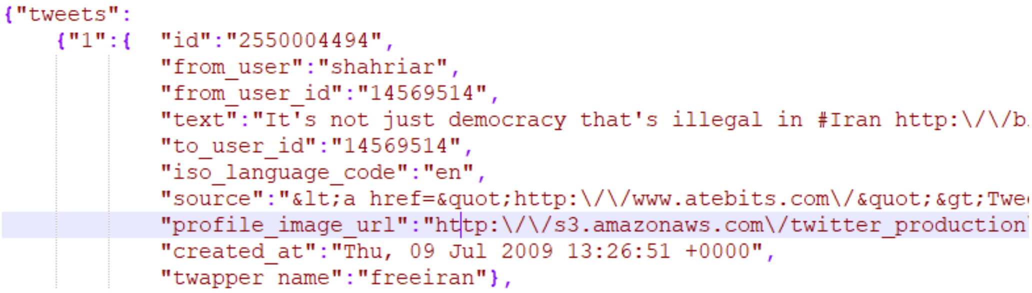
4.1 Introduction of Data

We will evaluate our algorithms on two real-world datasets. The first dataset is Twitter data, which is crawled from Twitter website and selected several topics for testing(16693 nodes and 85189 edges). There are six topics in the data. While the second dataset is Citation data, which is crawled from the Internet on several specific topics containing link and text information(1467 nodes and 5358 edges).

4.2 Data Parsing

4.2.1 Create network

Taking Twitter dataset as an example.The format of Twitter is shown below. So we can know the user and the Twitter he or she sent, and we can also know the user's friend from the data. We create the network using the dataset, where each node represents a user and the link represents that there is relationship between two users.



```
{"tweets":
    {"1":{  "id":"2550004494",
            "from_user":"shahriar",
            "from_user_id":"14569514",
            "text":"It's not just democracy that's illegal in #Iran http:\/\/b
            "to_user_id":"14569514",
            "iso_language_code":"en",
            "source":"&lt;a href=&quot;http:\/\/www.atebits.com\/&quot;&gt;Twe
            "profile_image_url":"http:\/\/s3.amazonaws.com\/twitter_production
            "created_at":"Thu, 09 Jul 2009 13:26:51 +0000",
            "twapper name":"freeiran"},
```

Figure 1: *data format*

4.2.2 Count topic distribution

From the introduction of topic-aware SI model we can know that to propagate a item we should know the topic distribution of the item, in other words, we should know the probability of the item i belonging to a topic z, which is denoted by $\gamma_i^z = p(z|i)$.

It's a problem belonging to text classification. First, we use Word2Vec method to learn how to use a vector to represents a word. The dimension of a word is 200 in our work. Second, we pick the top 5 word from each Twitter according to the importance of the word using ti-idf method. Then, we merge the vectors of the top 5 words, and using the 1000-dimension vector to train the machine learning model. The output of the model is the probability of a item i belonging to a topic z.

4.2.3 Count user interests

We should also know user's interests for each topics, which is denoted by p_u^z. Intuitively, we count the user's interests according to the proportion of a topic z in the all Twitter the user sent. However, some users sent too few Twitter(some even only sent one twitter), so this method is not credible enough. To solve this problem, we use the method below to count the interests.

For the users who sent much Twitter, we regard the proportion of a topic z in the all Twitter the user sent as the users' interests for z. While for the users who sent few Twitter, we use different method. We count the user's interests not only according to the Twitter he sent, but also according to the network we create. We calculate the user's interests according to his neighbor's interests, because we can say that the user's interest is similar to his follower or the users he follows. After calculate the interests using the two method, we averaged the two interests and regard it as the interests for the user who sent few Twitter.

5 Simulation

In this part, we will show the OJC source locating effect on two networks that based on real databases. The first database is citation data crawled from the Internet on several specific topics and the other is twitter data, including user and Twitter user posted.

5.1 Network based on Citation data

The network's initial property are listed:

nodes number	1297
edges number	5358
observation probability	0.689
total time slot	1

Denote x as the proportion of nodes contained in K that selected from unobserved nodes and y denotes the proportion of nodes contained in K that selected from observed infected nodes. We set $x = y$ and changes them from 0.5 to 0.8, step 0.1. Considering the scale of the computation and computer capability, we only choose sources to be one or two, and total time slot is set to 1 as listed. To assess the accuracy about location, we borrow the conception of eccentricity in [2]. Combinations with minimal average eccentricity to real sources are recorded.

Result for one source:

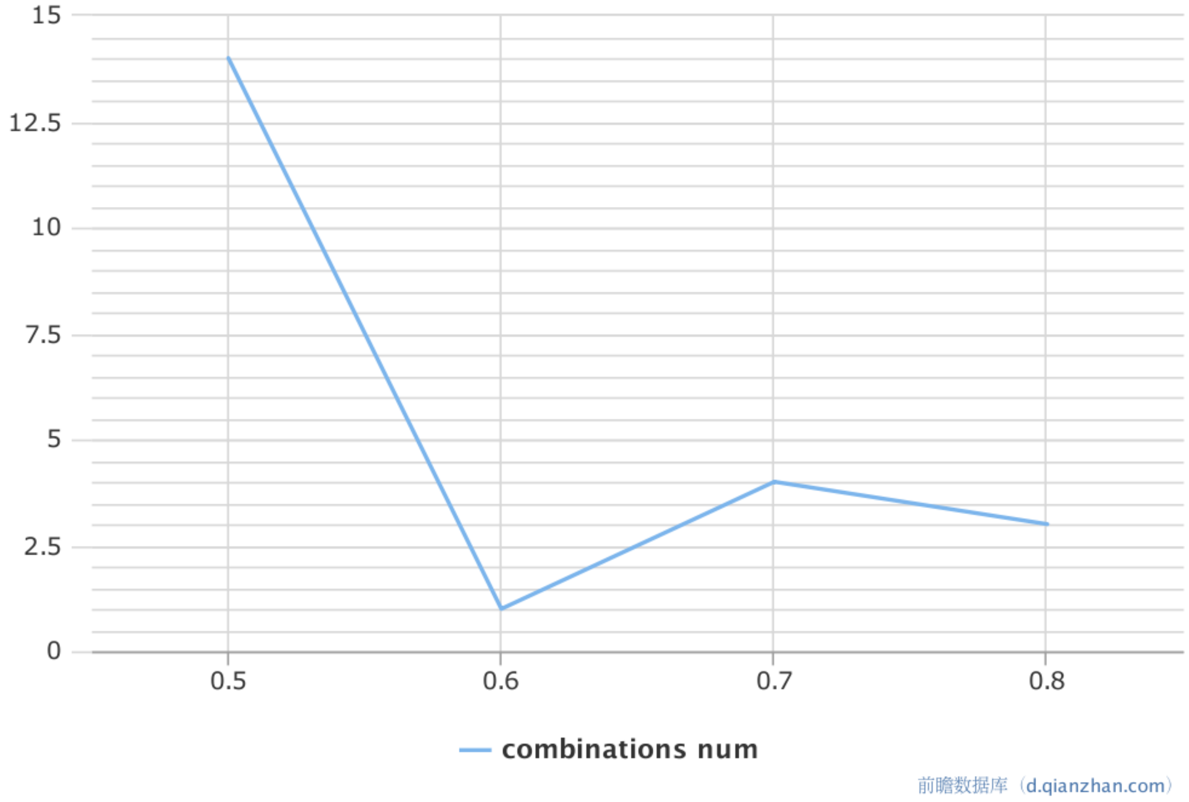


Figure 2: *one source:combination num*

Result for two sources:

It's coincidental that both figures is horizontal lines. In following analysis, we will pay less attention to these two as their particularity.

5.2 Network based on Twitter data

The network's initial property are listed:

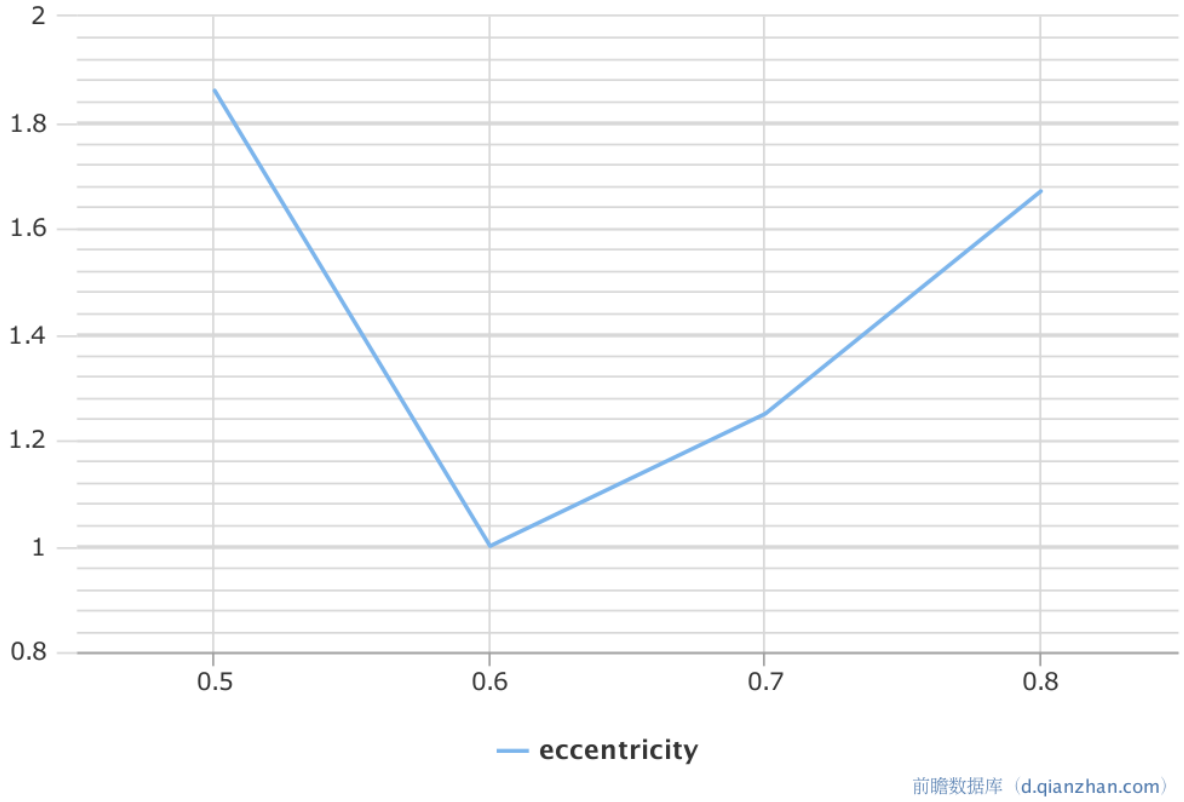


Figure 3: *one source:eccentricity*

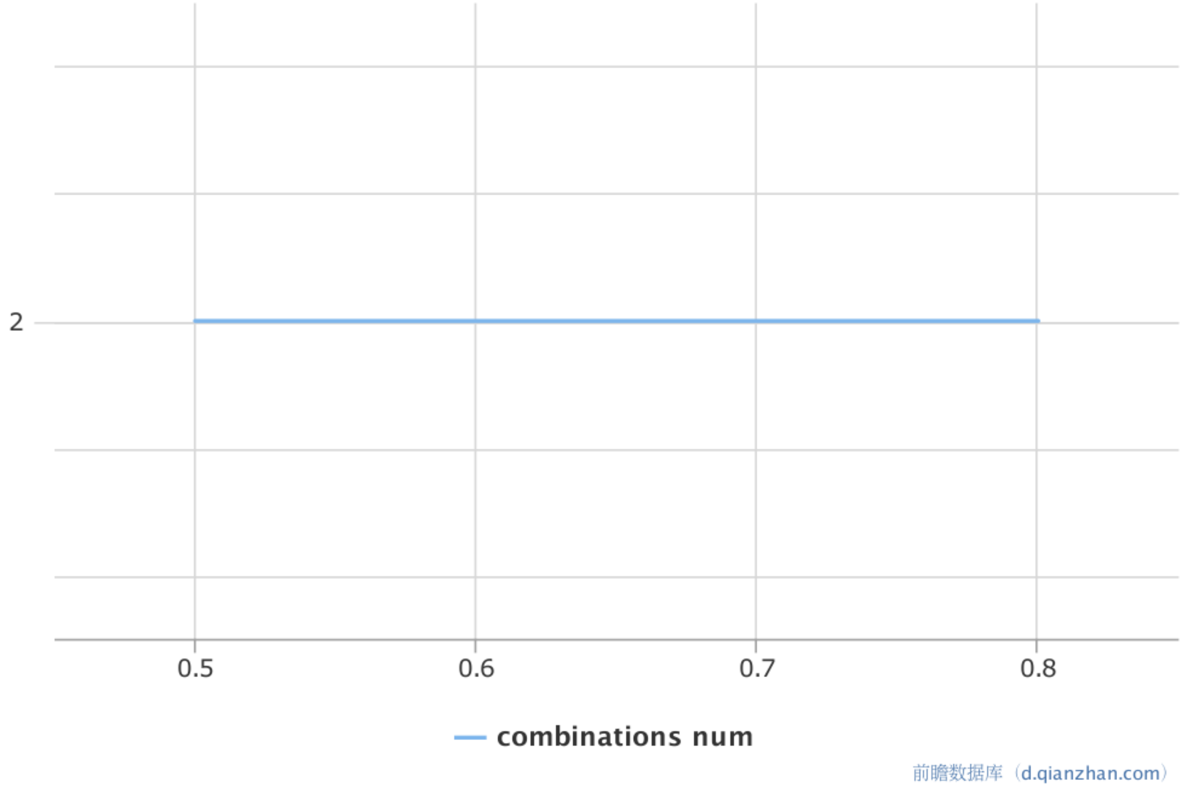


Figure 4: *two sources:combination num*

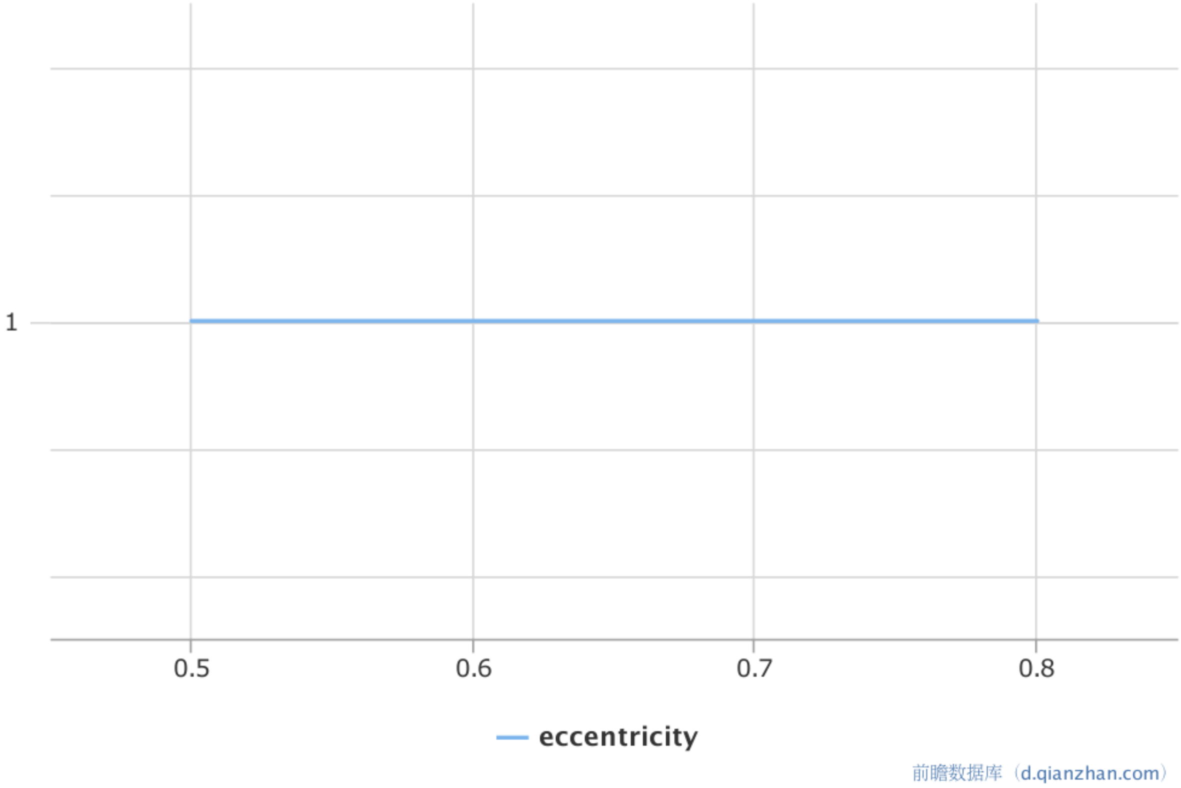
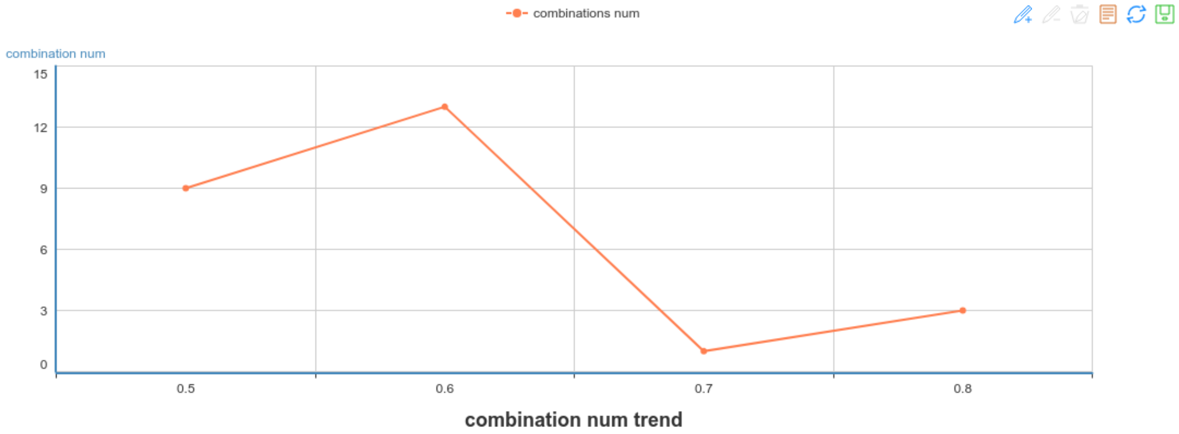
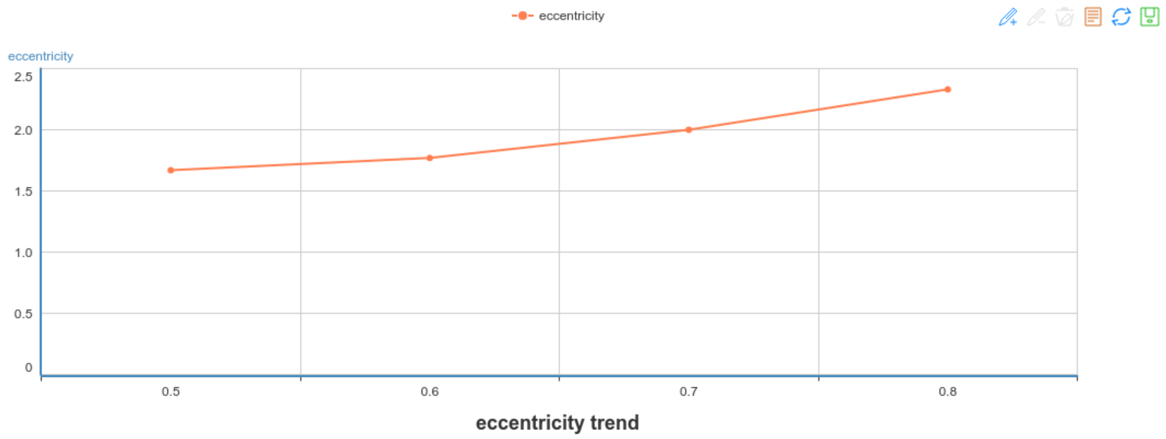


Figure 5: *two sources:eccentricity*

nodes number	16693
edges number	85189
observation probability	0.705
total time slot	2

Denote x as the proportion of nodes contained in K that selected from unobserved nodes and y denotes the proportion of nodes contained in K that selected from observed infected nodes. We set $x = y$ and changes them from 0.5 to 0.8, step 0.1. Considering the scale of the computation and computer capability, we only choose one source, and total time slot is set to 2 as listed. To assess the accuracy about location, we borrow the conception of eccentricity in [2]. Combinations with minimal average eccentricity to real sources are recorded.





5.3 Analysis

We can see from figures that as node picking proportion increase, candidate combination number decrease. It may be that sufficient nodes ensure combinations that is fairly close to real one, which is few but recorded. However, too many nodes may also result in interference as sources are too few, which results to increasing of eccentricity.

Limited by computer capability, we cannot carry out large scale simulation. In our expectation, the proper way to decrease the candidate combinations number and eccentricity is to increase the sources and time slot. As scale increases, accidental error will surely decrease.

Whatever, the average eccentricity shows that all combinations are close to real sources, which may indicates that the algorithm will work better in large scale situation.

6 Conclusion

In this report, we studied the problem of detecting multiple sources with partial observation on a topic-aware network. A topic-aware model and the OJC algorithm are combined in our work. Limited by computer performance, we only simulate with one source and two. However, simulation resultes are acceptable as the scale is so small. The work will be promising in large scale topic-aware source locating.

Reference

[1] Wenyu Zang et al. “Topic-aware Source Locating in Social Networks”. In: *WWW ’15 Companion Proceedings of the 24th International Conference on World Wide Web* (), pp. 141–142.

[2] Kai Zhu, Zhen Chen, and Lei Ying. “Catch’Em All: Locating Multiple Diffusion Sources in Networks with Partial Observations”. In: *the Thirty-First AAAI Conference on Artificial Intelligence* ().