Quantifying User Influence In Social Network Using Retweet cascade

Yifei Zhang

May 26, 2017

Abstract

Finding influential user in the social network enable the advertiser to promote their products better and help the company in making decision. Estimate user influence often base on a social network and diffusions. However, In many cases, the structure of a social network is hard to obtain, and the diffusion paths are hidden from observer due to privacy issues. The data people usually get is a cascade which is a collection of time that record when certain nodes in social network are activated. In this report, We propose an approach to estimate user influence base on the cascade. This method can detect the influential user without knowing the structure of underlying social network and the hidden path in diffusions.

1 Introduction

Since the origin of social media, big social network sites Twitter have already become the most popular places for people to get information and broadcast their ideas and it plays an increasingly important role in connecting different people[1].This makes social media become an ideal place for advertising and marketing since social media can enable information to diffuse into a large population of individuals in short time. However, to fully take advantage of the strength of social network as a marketing and advertising platform, studying user influence appears especially important. Knowing user influence enable people to gain a better understanding how information diffuse, why particular trend or a piece of information can spread fast and broad than others.This kind of knowledge can benefit company to advertise their product efficiently and can help marketer to design a successful campaign.

User influence can be applied to many areas,however there are two challenges in estimating user influence on a social network. The first is that obtaining a social network is non-trivial. As we know, social network becomes more complicated than ever.The size of a social network makes data difficult to collect.A single user in Twitter may have thousands even millions of friends and followers, which implies that in a social network one node can have thousands even

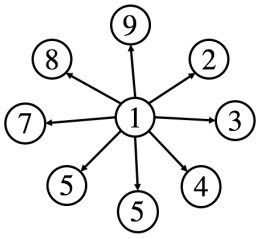


Figure 1: Observed cascade

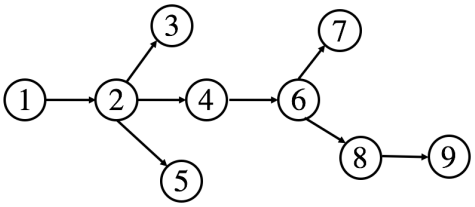


Figure 2: Hidden diffusion trace

millions in-degrees and out-degrees. Moreover, due to the limitation of public API access, sometimes data is not available for privacy issues. What people can observed is some diffusions. The current state of art, ConTinEst[2] can estimate influence of a node based on the social network which is inferred through multiple diffusions[3].However in practice, we do not have enough diffusions to do the network inference. what people can observed is one or two diffusions in the same social network. The second challenge is that the observed diffusion is a cascade. A cascade is a star-like directed graph shown in Fig.1 Every edge in this graph attach to the original node. The intermediary traces in cascade are lost. The cascade is just a sequence of time indicating when curtain nodes are activated. The number k in node u_k represents the time t_k when node is actived.Cascade is quite common in real life. In on-line news media, people can easily observe which and when a website publish a special news. However, it's hard to trace back from where this site gets the news. In marketing, we can uncover when and who buy a product or subscribe a service by checking the record. But it is hard to know who influence them to buy or subscribe. Thus, We know when and who but how and why information(be it in form of retweet, idea, news)propagates though individuals. The intermediary traces are hidden.In Twitter, we also have these two problems.The retweet can regard as a diffusion that flow on the social network which is constructed by user's friends and followers. Crawling friend list over a significant amount of user is time-consuming. What we can get is retweets on the fly. However, Due to the limitation of Twitter API, every retweeted tweet we get only shows it comes from the initial tweet. Thus, what we can observe is a retweet cascade, a star-like graph as shown in Fig. 1. However the really trace of the diffusions is like Fig.2 which is hidden from us. Simply applying a naive graph based measurement, out-degree, on a cascade,the center user will gain the most of the influence. While, as shown in Fig 2, the second user and sixth user will also gain some influence.

In this report, we solve the problems mentioned above by using proposed approach. Since the intermediate traces are hidden from us.we could not directly get user influence through a cascade.what we do is to estimate user influence by considering all possible diffusion traces that associated with an observed cascade. For each possible diffusion traces, we adopt the definition of influence

as the number of average users that a target user can diffuse information(be it tweet) to as in previous work[2]. By using the algorithm we proposed, the user influence can be quantified over all possible diffusion traces.The experiment result shows that our method can get intermediary influential user in a cascade and can obtain similar ranking quality comparing to the state of art.

There are three main contributions of this report:

- **Availability of Social Network** We address the problem of lacking the social network in estimating user influence. The proposed method can get user influence without knowing the structure of social network
- **Losing Intermediary Trace In Cascade** The proposed method does not rely on a particulate diffusion trace that generate the observed cascade.On the contrary, our method can estimate user influence through all possible diffusion traces in that cascade.
- **Algorithm** The number of possible diffusion traces are factorial increase with the size of a cascade. To efficiently compute the user influence over all possible diffusion traces, we develop a fast algorithm by exploiting linear algebra and matrix operations

The rest of this report is structured as follows. Sec.2 presents related work. Sec.3 details the proposed user influence estimation method, while Sec.4 presents how to solve the computation difficulty over huge amount of diffusion traces and the proposed algorithm is also detailed in Sec. 4 and the experiments are outlined in Sec.5. We conclude in Sec.6.

2 Related works

There has been a broad spectrum of algorithm developed to estimate user influence in social network. [4] use the number of followers ,number of rewteets, number of mentions to estimate user influence.A common approach to estimate user influence is computing the PageRank [5] over an observed social graph. TwitteRank[6] extend PageRank by taking consideration of both topical similarity between users and the structure in the social network. Another variant PageRank is temporal PageRank[7] which incorporates time information in ranking. All these method have some limitations. Calculating the social influence using naive measurements such as number of retweets, number of follower implies the hypothesis that social influence never disappear.This is not always correct, since users might retire from the active life of a social network-for example to move to another social network. [8]. While PageRank, TwiiterRank and temporal PageRank only function on the complete social network which we mentioned is hard to get in practice.

The current state of art ConTinEst use random(sampling) algorithm to approximate user influence[2] base on Continuous-Time Independent Cascade Model.By exploring the shortest path property of the independent cascade model ConTinEst converts the user influence estimation problem to neighborhood size estimation problem.To make ConTinEst work, people need to infer the social network. Such network inference requires multiple diffusions. However,in practice, we can not observed that much diffusions over the same social network.

3 Our proposal

In this section, we mainly talk about the proposed approach that estimate user influence.The problem we face is that our data is just a cascade. People do not observe the structure of a social network and the intermediary traces in the cascade are hidden.

Diffusion Scenarios Information diffuse as directed trees through the network.A diffusion scenario can be regarded as one of these tree. The root of the tree is the source where the information start to diffuse. The propagation process keep continuing by sequentially attaching new leaves to this tree over time as shown in Fig.3. Since we only observe the times when certain node are added through diffusion process,there are many possible diffusion scenarios that explain a observed cascades Fig 4.Fig.5

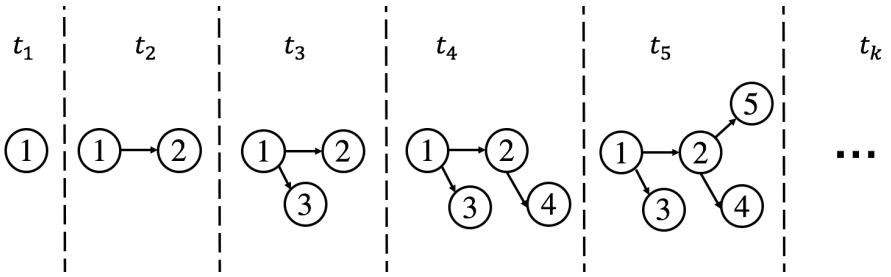


Figure 3: The diffusion process

Constructing Diffusion Scenarios through Cascade To estimate the user influence base on the given cascade. All possible diffusion scenarios need to be constructed. The concept of building diffusion scenario can show in Fig. 6. Every node in this figure represents a user. The first user u_1 who publish the initial tweet appears at t_1. Then we observed the second user u_2 at t_2. It has no option but retweet from u_1. A directed edge is drawn from u_1 to $u2$. However, when the third user u_3 come, it has two options, it can retweet either from the first user u_1 or the second u_2. Then two directed edges are drawn from u_1 to u_3 and u_2 to u_3. Thus, two diffusion scenarios G_1 and G_2 are constructed due to the different users to retweet. When the fourth user appears, it can retweet

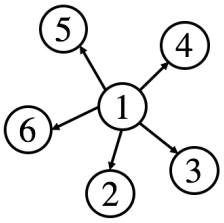


Figure 4: Observed cascade

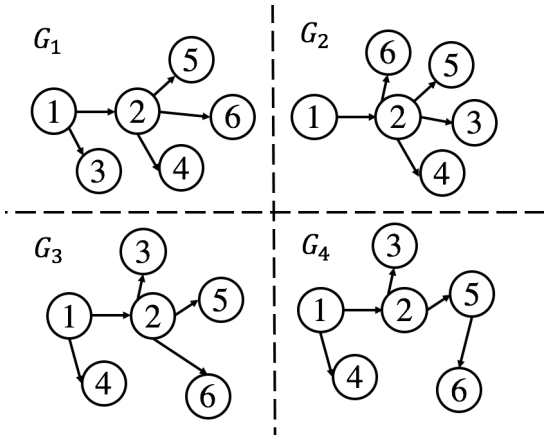


Figure 5: 4 possible diffusion scenarios associate with observed cascade

from u_1 to u_3 in any of previous diffusion scenarios. This process will continue till the last user we observe. Then All possible diffusion scenarios are built by sequentially adding users to previous diffusion scenarios.Eventually $k-1!$ number of diffusion scenarios are constructed when a cascade with the size of k is observed.

Probability of retweeting To estimate user influence. Only obtaining all the possible diffusions scenarios is not enough. Given a possible diffusion scenario G, we also need to know the probability of retweeting which corresponding to the weight of each edge in G. More specifically, let E be the collection of directed edge in diffusion scenarios G where each $e \in E$ is a directed edge $\{u_a, u_b\}$ representing u_b retweet from u_a.Every edge e correspond to a value representing the probability of retweeting. To model this probability, we made two assumptions: First, the user intent to retweet the latest tweet rather than tweet published a long time ago. This social phenomenon has been proved by many research. literatures[9][10].The influence of user is exponential decay.The second assumption is that user prefers retweet from a user who has a lot of followers. The number of followers a user has often related to its number of retweet[1]. More followers, a user has more channels he can get to broadcast his tweet to others, which increasing the likelihood of seeing by other users. [11][4]So the probability of retweeting can be express in the following equation:

$$Prob\{u_a, u_b\} = \frac{f(u_a)}{\sum_{u \in G} f(u)} e^{-r(t_b - t_a)} \tag{1}$$

This equation describe the probability that user u_b retweet from user u_a where t_a, t_b represent the time they retweet respectively. $f(u_a)$ gives the number of follower that user u_a has. $e^{-r(t_b-t_a)}$ express the probability decrease in exponential.G is one diffusion scenario which user u_b is added to.

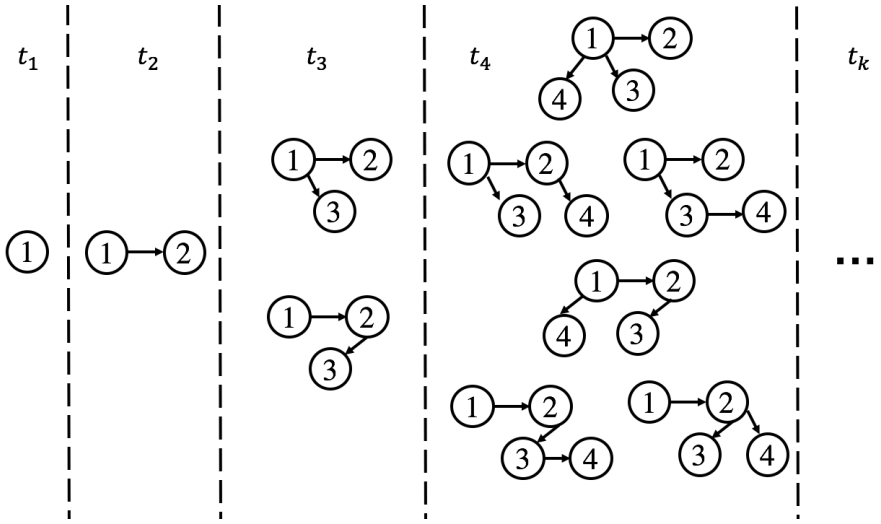


Figure 6: Construct Diffusion Scenarios

Probability of diffusion scenarios As we describe above, one diffusion is constructed by a sequence of retweeting edges. Each user u_b will retweet from another user u_a in its parent scenario with probability $Prob(u_a, u_b)$. As each rewteet represent an egde e in diffusion scenario.Thus, we can obtain the probability of a diffusion scenarios $Prob(G)$ via:

$$Prob(G) = \prod_{\{u_a,u_b\}\in E} Prob\{u_a, u_b\}$$

User influence in one diffusion Intuitively, given a diffusion scenarios G, the wider the spreading tweet, the more influential the user.We adopt the definition of influence as the number of average users that a target user can diffuse tweet to as in the previous work[2]. More precisely, let's say U is the collection of users in G. Consider a target user $u_i \in U$. This u_i can diffuse its tweet to any user $u_k \in U$ though the path $z(u_i, u_k)$ which is from u_i to u_k. The expected number of diffused user given a target user u_i can be computed as:

$$\begin{aligned}\varphi(u_i|G) &= \mathbb{E}\Big[\sum_{u_k\in U} \mathbb{I}(z(u_i, u_k|G))\Big] \\ &= \sum_{u_k\in U} \mathbb{E}\Big[\mathbb{I}\{z(u_i, u_k|G)\}\Big] \\ &= \sum_{u_k\in U} Prob\{z(u_i, u_k|G)\}\end{aligned}$$

Where $\mathbb{I}\{.\}$ is indicator function and the probability of path $z(u_i, u_k|G)$ can be obtained by the production over edges in this path z. That is:

$$Prob\{z(u_i, u_k|G)\} = \prod_{\{u_a,u_b\}\in z} Prob\{u_a, u_b\}$$

User influence over all possible diffusion scenarios Since we do not have only one diffusion scenarios.We have multiple diffusion scenarios. So the influence of target user $\varphi(u_i)$ can be express as summing out $\varphi(u_i|G)$ over all possible diffusion scenarios:

$$\varphi(u) = \sum_{G\in VG} Prob\{G\}\varphi(u|G)$$

Where VG is the collection of all possible diffusion scenarios that are generated from an observed retweet cascade. G is one possible diffusion scenario in VG. The final user influence over all possible diffusion scenarios is:

$$\varphi(u) = \sum_{G\in VG} Prob\{G\} \sum_{u\in G} \prod_{\{u_a,u_b\}\in z} \frac{f(u_a)}{\sum_{u\in G} f(u)} e^{-r(t_b-t_a)} \tag{2}$$

It is challenging to compute equation(2) directly, Particularly given that the number of diffusion scenarios is factorial with the size of diffusion cascade.For example, there 10^{137}possible ways in which a cascade of 100 events can unfold.To solve this problem, We develop an efficient algorithm in the next section.

4 Efficient Influence Estimation

The key observation here is that users u_k are sequentially added to previous diffusion scenarios at time t_k. If we ignore u_k, the structure of all previous diffusion scenarios remains unchanged. This observation allows us to explore the relation between $\varphi(u)$ in t_k and t_{k-1} and it turns out user influence can be computed incrementally. We do not need to keep track of each possible diffusion scenario. We only need to keep track of how user influence increase over time.

4.1 Derive Relation Between Sequential Diffusion scenarios

Following the description in section 3,users are sequentially attached to previous diffusion scenarios. When user u_k is attached to previous diffusion scenarios constructed in time $k-1$ with a collection of users $U = \{u_1, u_2, u_3 \cdots, u_{k-1}\}$, for all $G \in VG^{k-1}$ at time t_{k-1}, each G can generates $k-1$ new scenario G_i by adding edge $\{u_i, u_k\}$ where $G_i = G \cup \{u_i, u_k\}$.$G$ is the parent diffusion scenario

of G_i Fig. 7. So $\varphi^k(u_j)$ can be written as:

$$\begin{aligned}\varphi^k(u_j) &= \sum_{G \in VG^k} Prob(G)\varphi(u|G) \\ \varphi^k(u_j) &= \sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} Prob(G_i)\varphi(u_j|G_i)\end{aligned} \tag{3}$$

According to the definition of the user influence in one diffusion scenario, the

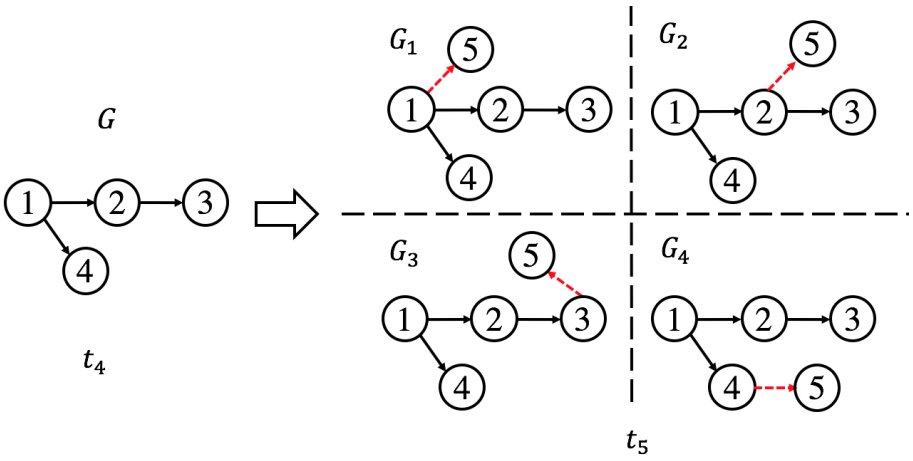


Figure 7: A diffusion scenario with 4 users at time t_4 will generate 4 new diffusion scenarios at t_5 by attach u_5 to each user in t_4

influence of user u_j in scenario G_i at time t_k can be written as:

$$\varphi(u_j|G_i) = \sum_{\substack{u_v \in G_i \\ v>j}} Prob\{z(u_j, u_v|G_i)\}$$

$\sum_{u_v \in G_i; v>j} Prob\{z(u_j, u_v|G_i)\}$ can be decomposed into two parts. The first part the probability of all the paths from u_j to other users except u_k and the second part is the probability of path from u_j tou_k

$$\varphi(u_j|G_i) = \sum_{\substack{u_v \in G_i \\ v>j \\ v \neq k}} Prob\{z(u_j, u_v|G_i)\} + Prob\{z(u_j, u_k|G_i)\}$$

We make use of the fact that G_i without edge $\{u_i, u_k\}$ is equal to its parent scenario G. So $\sum_{\substack{u_v \in G_i \\ v>j \\ v \neq k}} Prob\{z(u_j, u_v|G_i)\}$ is actually equal to $\sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\}$ then

$$\varphi(u_j|G_i) = \sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_k|G)\} + Prob\{z(u_j, u_v|G_i)\} \tag{4}$$

By combining equation (4), equation(3) is written as:

$$\varphi^k(u_j) = \sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} Prob(G_i) \left[\sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\} + Prob\{z(u_j, u_k|G_i)\} \right]$$

$$= \underbrace{\sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} \left[Prob\{G_i\} \sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\} \right]}_{A} + \underbrace{\sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} \left[Prob(G_i) Prob\{z(u_j, u_k|G_i)\} \right]}_{B}$$

For part A, what we interested in is that how A relate to $\varphi(u_j)$ in previous timestamps t_{k-1}. we utilize the fact again that $G_i = G \cup \{u_i, u_k\}$. Thus $Prob\{G_i\} = Prob\{G\}Prob\{u_i, u_k\}$. Then A can be expressed as:

$$A = \sum_{G \in VG^{k-1}} \sum_{i=0}^{k-1} Prob\{G\} Prob\{u_i, u_k\} \sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\}$$

$$= \sum_{G \in VG^{k-1}} Prob\{G\} \sum_{i=1}^{k-1} Prob\{u_i, u_k\} \sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\}$$

The next importance fact is that $Prob\{u_i, u_k\}$ describe the probability that u_k retweet from any previous user u_i where $u_i \in \{u_1, u_2, \cdots, u_{k-1}\}$. Thus $\sum_{i=1}^{k-1} Prob\{u_i, u_k\}$ need to normalize to 1.

$$A = \sum_{G \in VG^{k-1}} Prob\{G\} \sum_{\substack{u_v \in G \\ v>j}} Prob\{z(u_j, u_v|G)\}$$

$$A = \sum_{G \in VG^{k-1}} Prob\{G\} \varphi(u_j|G)$$

Finally, A fits in the definition of user influence over all diffusion scenarios at t_{k-1}, that is:

$$A = \varphi^{k-1}(u_j)$$

For part B, based on the description in section 3, we observed the only difference between diffusion scenarios in t_k and t_{k-1} is where u_k place, which indicates the change of influence for each previous user from t_{k-1} to t_k solely relay on u_k. So far, we already separate A which is user influence $\varphi^{k-1}(u_j)$in t_{k-1} from user influence $\varphi^k(u_j)$in t_k. Therefore, the rest of the part B is actually the influence that user u_k contribute when it is added at t_k. If we define $M_{jk} = B = \sum_{G \in VG^k} Prob(G) Prob\{z(u_j, u_k|G)\}$ as M_{jk} which is the influence that u_j exerts on u_k.

$$\varphi^k(u_j) = \varphi^{k-1}(u_j) + M_{jk}$$

So the influence of u_j at t_k when u_k is added can be obtained through the influence of u_j at $tk-1$ and the influence that u_k contribute to u_j. Then the $\varphi^k(u_j)$ can be calculated by summing out the influence that u_j over all previous users $(u_1, u_2, \cdot, u_{k-1}, u_k)$

$$\varphi^k(u_j) = \sum_{i=1}^{k} M_{ji} \tag{5}$$

The only problem left is to compute M_{jk} which is part B. The basic idea here is also to derive the relation between t_k and $t_k - 1$ by using the observation we have discussed.

$$B = M_{jk} = \sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} Prob(G_i) Prob\{z(u_j, u_k|G_i)\} = \sum_{G \in VG^k} Prob\{G\} Prob\{z(u_j, u_k|G)\} \tag{6}$$

We know that $G_i = G \cup \{u_i, u_k\}$ thus $Prob\{G_i\} = Prob\{G_i\} Prob\{u_i, u_k\}$ and $Prob\{z(u_j, u_k|G_i)\} = Prob\{z(u_j, u_i|G)\} Prob\{u_i, u_k\}$ so:

$$\begin{aligned}
M_{jk} &= \sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} \Big[Prob(G) Prob\{u_i, u_k\} \Big] \Big[Prob\{z(u_j, u_i|G)\} Prob\{u_i, u_k\} \Big] \\
&= \sum_{G \in VG^{k-1}} \sum_{i=1}^{k-1} \Big[Prob^2\{u_i, u_k\} Prob(G) Prob\{z(u_j, u_i|G)\} \Big] \\
&= \sum_{i=1}^{k-1} \Big[Prob^2\{u_i, u_k\} \sum_{G \in VG^{k-1}} Prob(G) Prob\{z(u_j, u_i|G)\} \Big]
\end{aligned}$$

By comparing the final and the original form of B:

$$\begin{aligned}
B &= \sum_{G \in VG^k} Prob(G) Prob\{z(u_j, u_k|G)\} \\
&= \sum_{i=1}^{k-1} \Big[Prob^2\{u_i, u_k\} \sum_{G \in VG^{k-1}} Prob(G) Prob\{z(u_j, u_i|G)\} \Big] \\
M_{jk} &= \sum_{i=1}^{k-1} \Big[Prob^2\{u_i, u_k\} M_{ji} \Big]
\end{aligned}$$

Through this equation we can observed that M_{jk} which is the influence u_j over u_k can be recursively calculated using $Prob\{u_i, u_k\}$ and M_{ji} which is influence u_j over u_i where u_i is the user that u_k attach to. Any user in a diffusion can not have the influence on any previous user because there is no directed path pointing back to the previous user. Therefore $M_{jk}(j > k)$ equals zero and the influence on the user itself set to one in default.

$$M_{jk} = \begin{cases} 1, k = j \\ 0, j > k \\ \sum_{i=1}^{k-1} M_{ji} P_{ik} \end{cases} \tag{7}$$

where P_{ik} is $Prob^2\{u_i, u_k\}$

4.2 Overall algorithm

So far, we have achieved an efficient method to keep track user influence over time with all possible diffusion graphs using a set of M_{jk}. $\varphi^k(u_j)$ can be computed by maintaining two matrices P_{ij} and M_{ij}. The element of these matrices can be obtained via equation (1),(7)

Algorithm 1 Calculate Matrix M_{ji}

Require: Matrix P_{ji}, Number of user k
Ensure: M_{ji}

```
for i = 1 to k do
  for j = 1 to k do
    if j = i then
      M_ji = 1
    end if
    if j > i then
      M_ji = 0
    else
      for l = 1 to i − 1 do
        M_ji ← M_jl * P_li
      end for
    end if
  end for
end for
```

4.3 Fast user influence estimation with matrix operation

This algorithm which describes in the previous section can also be written in matrix processing, The element of the matrix M_{ij} can be obtained by the following steps:

$$\underbrace{\begin{bmatrix} 1 & P_{12} & P_{13} & P_{14} & \cdots & P_{1i} \\ 0 & 1 & P_{23} & P_{24} & \cdots & P_{2i} \\ 0 & 0 & 1 & P_{34} & \cdots & P_{3i} \\ 0 & 0 & 0 & 1 & \cdots & P_{4i} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{P_{ji}} \underbrace{\begin{bmatrix} 1 & M_{12} & M_{13} & M_{14} & \cdots & M_{1i} \\ 0 & 1 & M_{23} & M_{24} & \cdots & M_{2i} \\ 0 & 0 & 1 & M_{34} & \cdots & M_{3i} \\ 0 & 0 & 0 & 1 & \cdots & M_{4i} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{M_{ji}}$$

$$
\begin{aligned}
&M_{11} = 1 && k = 1 \\
&M_{12} = P_{12}M_{11} && k = 2 \quad [P_{12}]\,[M_{11}] = [M_{12}] \\
&\left.\begin{aligned} M_{13} &= P_{13}M_{11} + P_{23}M_{12} \\ M_{23} &= P_{13}M_{21} + P_{23}M_{22} \end{aligned}\right\} && k = 3 \quad \begin{bmatrix} 1 & M_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{13} \\ P_{23} \end{bmatrix} = \begin{bmatrix} M_{13} \\ M_{23} \end{bmatrix} \\
&\left.\begin{aligned} M_{14} &= P_{14}M_{11} + P_{24}M_{12} + P_{34}M_{13} \\ M_{24} &= P_{14}M_{21} + P_{24}M_{22} + P_{34}M_{23} \\ M_{34} &= P_{14}M_{31} + P_{24}M_{32} + P_{34}M_{33} \end{aligned}\right\} && k = 4 \quad \begin{bmatrix} 1 & M_{12} & M_{13} \\ 0 & 1 & M_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{13} \\ P_{13} \\ P_{13} \end{bmatrix} = \begin{bmatrix} M_{13} \\ M_{13} \\ M_{13} \end{bmatrix}
\end{aligned}
$$

According to the equation(7), each kth colored column vector of matrix M_{ji} can be computed by sub-matrix with size of $k-1 \times k-1$ in M_{ji} multiplying kth column vector with same color in P_{ji}.Therefore, given matrix P_{ji}, from $k = 1$, Mij can be computed incrementally follow the process above.

5 Experiments and results

We do experiments both on synthetic data and real world data (retweet cascade).The result shows that our approach can give the similar result comparing to the state of art and our approach can detect the intermediary influential user.

Synthetic network data We use the synthetic data that ConTinEst[2] provide.This data is generated from Kronecker network[12] with the periphery matrix$[0.9, 0.5; 0.5, 0.3]$ which mimic the information diffusion traces in the real social network. Then we assign a probability for every directed edge in this network. We use exponential distribution ($Prob(u_a, u_b) = \lambda e^{-\lambda(t_a - t_b)}$) to generate these probabilities and set λ randomly.For every single user in this network, we use ConTinEst to get its influence. We also use the same distribution to compute the matrixPij in our method and then estimate user influence. To compare the result between two approaches, we normalize both of them in the range$(0, 1)$ as shown in the Fig. 8,Fig. 9. We explore the ranking quality by using Kendall's τ and Spearman's ρ rank correlation coefficient and the linear relationship by using Pearson correlation coefficient. The result are shown in Table 1.We can see we get similar result with ConTinEst[2]

Table 1:

p	Spearman	Kendall	Pearson
Time	0.7856	0.7013	0.8462

Real world data The retweet cascade data were collect by [13](via Twitter API) published between 2014-05-29 and 2014-12-26 which mentions YouTube videos. We first apply our algorithm to a single cascade to get user influence over one cascade.Then we use multiple cascades(12000)associated with one video to estimate influence by summing out user influence in each cascade. The result

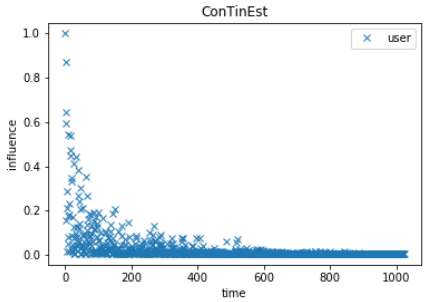


Figure 8: ConTinEst

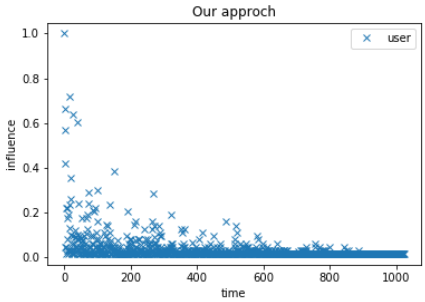
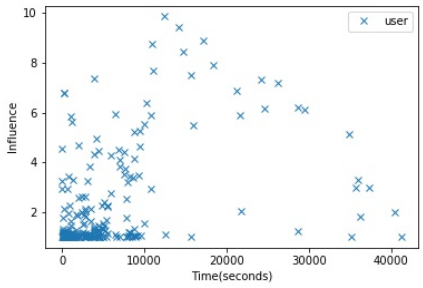
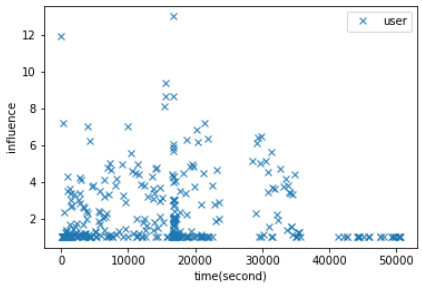


Figure 9: Our approach



(a) User influence in one cascade



(b) user influence in one multiple

Figure 10: User influence in one cascade and multiple cascade

Fig.10 shows that apart from the first user the user later on also gain influence, which indicating our algorithm can find the intermediary influential user.

6 Conclusion

We proposed a new method to calculate user influence through a cascade and give an efficient user influence estimation algorithm. Our approach gets the similar result compared to the previous state and finds intermediary influential user while do not rely on the structure of a social network and the hidden diffusion traces.

Reference

[1] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 591–600.

[2] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *NIPS '13: Advances in Neural Information Processing Systems*, 2013.

[3] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *ICML '11: Proceedings of the 28th International Conference on Machine Learning*, 2011.

[4] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy." 2010.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[6] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 261–270.

[7] P. Rozenshtein and A. Gionis, "Temporal pagerank," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 674–689.

[8] P. Feng, "Measuring user influence on twitter using modified k-shell decomposition," 2011.

[9] R. Crane and D. Sornette, "Robust dynamic classes revealed by measuring the response function of a social system," *Proceedings of the National Academy of Sciences*, vol. 105, no. 41, pp. 15 649–15 653, 2008.

[10] S. Mishra, M.-A. Rizoiu, and L. Xie, "Feature driven and point process approaches for popularity prediction," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. New York, NY, USA: ACM, 2016, pp. 1069–1078. [Online]. Available: http://doi.acm.org/10.1145/2983323.2983812

[11] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 65–74.

[12] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[13] S. Mishra, M.-A. Rizoiu, and L. Xie, "Feature driven and point process approaches for popularity prediction," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1069–1078.