Seed Set Selection in Evolving Social Networks

Shuai Xu*, Naiting Xu, Jiahao Zhang, Feiyang Li, Shasha Li
School of Software Engineering
University of Science and Technology of China
Hefei, Anhui, P.R. China
e-mail: sa615527@mail.ustc.edu.cn,{sa615237, jhcheung, mmmwhy, lisa1990}@mail.ustc.edu.cn

***Abstract*—With the popularity of social network sites, a huge volume of data has been produced. Motivated by the value of big data, a number of organizations and business companies have been putting effort into analyzing big data to increase business performance. Then many problems have been raised about social network analysis views big data, and one of key research issues is social influence analysis. Unfortunately, in this area, a large amount of research works only focused on analyzing the static social network. In detail, in this kind of network, we can use many existing efficient algorithms to select the most influential nodes which can maximize influence spread. However, the real-life network structure will change constantly over time, so a series of network snapshots will be generated. In this case, those nodes can not maximize influence spread in the following time stamps. In this paper, we propose a novel and efficient method to select seed set in evolving social network under the information of this network is only available at first time stamp. Then this seed set always makes good performance of influence diffusion during the evolution of social network.**

Experimental evaluations using both real and synthetic large dynamic networks show that our proposed algorithm achieves a better performance than the methods ignoring real-life social network evolution.

Keywords-big data; social influence analysis; influence spread; partial nodes; graph prediction

I. INTRODUCTION

Recently, with the fast development of the World Wide Web, many social network sites are becoming popular and successful, such as Twitter, Facebook, Google Plus, Skype communication network, paper reference network and so on. They evolve into a platform for people to analyze the spread of product adoption, ideas, news and relationship. Hence a huge amount of data has been produced. A considerable amount of research has been focused on analyzing those big data social networks, many problems have been raised and resolved. Specifically, social influence analysis is one of key issues. In this paper, we try to solve it, selecting the most influential nodes of size k in a static network to get a better result during the evolution of social network, which is so-called seed set selection in evolving social network.

We can give a realistic example of its application. A smart phone company develops a new smart phone, and wants to market this phone to society through social network. Due to constrained promotional budget, only a limited number of people could get one phone for free. The purpose of the company is to make those people with the phone to affect more people to buy this mobile phone. This problem is so hard that we must spend much time to research. The major challenge is that a real-life social network has a highly dynamic nature, and evolves rapidly over time. At the same time, the size of social network is generally very large with complex structure. The company wishes those initial people can maintain the great influence spread and affect more people over time.

A. Relate Work

There has been extensive research on social network analysis, in which influence diffusion and influence maximization are two key research issues. Motivated by the probabilistic soluti-on proposed by Domingos and Richardson [1], Kempe etc. [2] originally formulated influence maximization problem as a discrete optimization problem. Under Independent Cascade Model (IC) and Linear Threshold Model (LT) , they showed influence maximization problem is NP-hard, monotonic and has sub-modular properties of influence diffusion function. Later, a greedy strategy called hill-climbing was proposed with a 1-1/e approximation guarantee.

Following the work of Kempe etc. [2], a large amount of research works have been focused on influence maximization problem, then many different efficient and state-of-the-art influence maximization algorithms have been proposed. However, those proposed algorithms mainly focused on the static network. Leskovec etc. [3] studied the sub-modular properties of influence diffusion to propose a cost-effective lazy forward (CELF) which improves the simple greedy algorithm. Chen etc. [4] also improved the greedy algorithm with NewGreedyIC for the IC model. And they presented a heuristic algorithm with Degree Discount which is generally very efficient and fast in practice but finds lower quality influential nodes. This strategy aimed to find the max-degree nodes every time and the degree of its neighbor nodes should be reduced.

There are also some research works on influence diffusion and influence maximization in the evolving social network. Zhuang *et al*. [5] probed a small number of nodes in an unobserved social network at certain time stamp, however, information of full network is not available. Chen *et al*. [6] proposed a novel strategy to track a subset of nodes of size k which maximizes influence diffusion, then iteratively updates those nodes during the evolution of network.

B. Our Contribution

In summary, for selecting top-k most influential nodes in a social network, current most of exiting algorithms are only adapted for a static network. As the network evolves over time,

978-1-5090-6352-9/17/$31.00 ©2017 IEEE

those selected nodes will make poor performance of influence spread. Thus most influential nodes selection in an evolving network is a key and urgent research issue.

We propose a novel and efficient method to solve this problem. Firstly, we make a definition for this problem. Obviously, we can only acquire the network at $t = 0$, while the evolution of the network is not known at other time stamps. Thus we propose graph prediction algorithm based on common neighbor to predict the evolution of network. However the size of real-life social network is generally large, so a huge amount of time must be spent on prediction for all nodes in network. Thus we propose an efficient algorithm to select some nodes to predict the evolution of network. By using the prediction algorithm, a sequence of network snapshots can be generated in the following time stamps. Finally, A strategy is proposed to select top-k most influential nodes from those network snapshots.

We have conducted extensive experiments to validate our method on both real-life and synthetic social networks. Finally, experimental results show that our method has good performance and conclusion is correct.

C. Paper Organization

Section II introduces the concept of dynamic social network and defines the problem of seed set selection in an evolving social network. Section III introduces the strategy that finding the most influential nodes under IC model. Section IV shows our experiments result on two real-word and one synthetic dynamic social networks with time stamps. Section V shows the conclusions and future work of this paper.

D. Notations

We assume that the time t is discrete. Then a static directed social network at time t can be denoted as $G^t = (V^t, E^t)$, where V^t is a node set of size n and $E^t \subseteq V^t \times V^t$ is an edge set of size m. Namely $e(v,w)$ is the edge for v to w, $p(v,w)$ is the weight of $e(v,w)$. For each node $v \in V^t$ in G^t, we can denote the in-degree and out-degree as $d_G^-(v)$ and $d_G^+(v)$ respectively. A static network can evolve and produce a sequence of networks: $\{G^t\}_{t=0,\ldots,T}$, where T is the period of the evolving network. Furthermore, $N(v)$ is the neighbors set of node v, $h_t(v,w)$ is similarity value between v and w at time t, $H(v,w)$ is the value of $\sum_{t=1}^{T} h_t(v,w)$, $P(v,w)$ is final prediction value between v and w. $CN_t(v,w)$ is the set of common neighbors between v and w at time t, $CB_t(v,w)$ is closeness between the common neighbors of v and w at time t, $CD_t(v)$ is change degree of node v at time t, $Tw(t)$ is time weight at time t and $AN_t(v,w)$ is the set of v, w and their redefined common neighbors at time t.

II. PRELIMINARIES

In this section, we firstly introduce the definition of the evolving directed social network, independent cascade model and influence maximization problem. Then we introduce two previous algorithms, which will be used in our paper.

A. Social Network Evolution

An evolving directed social network at time t is defined as a graph $G^t = (V^t, E^t)$, where V^t is a node set and E^t is an edge set of between every pair of nodes in G^t at time t. For example, in a friendship network, each node represents a person. An edge $e(v,w)$ represents that v is a friend of w. Everyone will make new friends and lose friends for different reasons over time. Thus the network structure will change constantly, then a sequence of network snapshots will be generated and denoted by $\{G^t\}_{t=0,\ldots,T}$ where T is the period of the evolving network, and we assume that T is finite. This network is called the evolving network.

B. Independent Cascade Model.(IC)

Given a static directed social network $G^t = (V^t, E^t)$ at time t. For each $v \in V^t$, it is only one of two status which are active and inactive. If at time $t = t_1$, w is inactive , and $N(w)$ are the neighbors of w. Then for each $u \in N(w)$, if u is active, u has a single chance to activate w. It will succeed with probability $P(u,w)$. If all $u \in N(w)$ fail, w is still inactive. Otherwise, w will be active at next time $t+1$. This method has been employed for this purpose [2,3,4,7,8].

C. Influence Maximization Problem

Given a static directed social network $G^t = (V^t, E^t)$ at time t and an integer k, influence maximization problem aims to find a subset of nodes of size k. We use $S \subseteq V^t$ to denote this subset that will maximize the influence spread. This problem has been proved to be NP-hard [2].

D. Monte Carlo Simulation Methods

Monte Carlo Simulation is an efficient and simple algorithm for estimating influence spread and conduct on the static social network snapshots. Given an initial node set of size k, it runs the process for thousands of times and reports the average influence spread for the initial active node set. This value is the expected influence spread. We will use this method in our proposed algorithms.

E. Degree Discount Algorithm (DDA)

Degree Discount algorithm proposed in [4] has several advantages such as: it can be implemented easily, it is very fast and scalable. The general idea is as follows. Let $N(w)$ are the neighbors of vertex w. For each $v \in N(w)$ which is one of nodes selected, we should not count the edge $e(w,v)$ for w's out-degree, when considering w as a new seed depending on its out-degree. This algorithm can be used to select seed set of size k in a static social network and those seed set can maximize the influence diffusion. Thus we will use this algorithm in our proposed algorithms.

Problem definition: Seed Set Selection in Evolving Social Networks.

Let G^0 be a network snapshot at time $t = 0$. At each time stamp $t > 0$, the network structure will change, and the changes can not be known. Thus at time $t = T$, a sequence of network snapshots as $\{G^t\}_{t=1,\ldots,T}$ will be generated. The problem aims to select seed set of size k in G^0 under the information of G^0 is only available.Let $S \subseteq V$ denotes this seed set which will obtain good influence spread in $\{G^t\}_{t=1,\ldots,T}$.

III. PROPOSED METHODS

In this section, we introduce three algorithms for selecting the most influential node set.

A. Selecting The Nodes To Predict

Intuitively, the size of real online network is generally large, so a huge amount of time is spent on prediction for all nodes in network. In generally, most nodes change very little and bring the least change of the solution on the networks at following time stamps. Thus we could select partial nodes to predict networks rather than using all nodes.

A naive method for selection is to choose some nodes with equal probability. However, it is almost impossible to minimize the difference of influence maximization between the real network and the predictive network.

Our strategy is to formulate an objective function in Equation (1). This strategy represents the probability of minimizing the loss between the real network and the predictive network. We propose the Maximum Gap Selecting (MGS) algorithm (Algorithm 1) to optimize our objective function, where the MGS algorithm is adapted from the MaxG algorithm proposed in [5]. In detail, MaxG algorithm updates graph snapshots by probing some nodes, then selects seed set. Based on the strategy for probing, we propose MGS algorithm.

1) Maximum Gap Selecting(MGS): Suppose we are given two graph snapshots of an evolving social network at time $t=0$ and $t=1$, they are denoted as G_0 and G_1 respectively. Assume node $v \in G_0 \cap G_1$, G is defined as the network by updating the connections of node v in G_0 according to G_1. We select optimal seed set S_0 and S by using Degree Discount Algorithm(DDA) in G_0 and G respectively, then the difference of solution could be measured by $PG_v = \sigma_v(S) - \sigma_v(S_0)$, where PG_v is called as "performance gap" for node v, $\sigma_v(S)$ indicates the influence spread of seed set S.

For each $v \in G_0$, the PG_v should be estimated, then we select satisfaction nodes based on PG_v. The condition is defined in Equation (1).

$$Pr[\sigma_v(S) - \sigma_v(S_0) \geq \kappa(v)] \leq \alpha \quad (1)$$

Where α is the probability showing that only PG_v is satisfied when the occurrence probability is not less than α, and $\kappa(v)$ is accordingly defined. Our MGS algorithm is to select those nodes which can maximize $\kappa(v)$.

Several optional methods can estimate the $\sigma_v(S)$, in this paper, we use the sum of the in-degree in seed set S to estimate it. While the precise value of $\kappa(v)$ is hard to calculate without specifying the in-degree distribution of each node v. Thus we should estimate the value of $\kappa(v)$, then Azuma-Hoeffding inequality can be used and shown in the following part.

2) Estimation of $\kappa(v)$: When we choose node v from G_0 to predict, there are two situations that v is in S_0 or not.

Indeed we can deduce the value of $\kappa(v)$ which is borrowed from [5] (Section III in [5]), then write it as follow:

If $v \in S_0$, $\kappa(v) = \max\{0, \max_{a \notin S} d_G^-(a) - d_G^-(v) + \beta_v\}$

If $v \notin S_0$, $\kappa(v) = \max\{0, d_G^-(v) - \min_{b \in S} d_G^-(b) + \beta_v\}$

Where $\beta_v = \sqrt{-2\theta_v \ln \alpha}$ and $\theta_v = t$

Algorithm 1 MGS procedure

Input: G^t, K, I, α
Output: Node set P^t

```
1:  initialize SN^t = φ; G = G^t; ∀ v ∈ V, θ_v = t;
2:  for k =1; k <= K; k++ do
3:      for i =1; i <= I; i++ do
4:          SP=SP ∪ argmax_{v∈V\SP} d_G^-(v);
5:      end for
6:      dn = d_G^-(SP_I);
7:      dm = argmax_{v∈V\SP} d_G^-(v);
8:      for all v∈V do
9:          β_v = sqrt(-2θ_v ln α);
10:         if v∈SP then
11:             κ_v = max{dm - d_G^-(v) + β_v, 0};
12:         else
13:             κ_v = max{d_G^-(v) - dn + β_v, 0};
14:     end for
15:     v* = argmax_{v∈V} κ_v;
16:     θ_{v*} = 0, SP = φ;
17:     P^t = P^t ∪ {v*};
18: end for
19: return P^t;
```

The pseudo-code of MGS algorithm is shown in Algorithm 1,where G^t is a graph snapshot at time t, K is the size of P^t, and we set α=0.01.

Initially, we choose I nodes with maximum in-degree (Lines 3-5). Then we calculate the minimum in-degree of node set SP (Line 6), and the maximum in-degree of difference sets of V and SP (Line 7). Next, in Lines 8-13, we can calculate the estimation of $\kappa(v)$. Finally,the node set of size k can be attained (Lines 15-17).

B. Graph Prediction Based On Common-Neighbors

1) Link Prediction: Suppose that we have known a static network $G^t = (V^t, E^t)$ at time t, and it will evolve over time by adding new nodes, adding new edges, losing nodes and losing edges. Then a sequence of graph snapshots will be produced, and denoted as $\{G^t\}_{t=0,\ldots,T}$. The object of link prediction is to calculate the occurrence probabilities of edges between each pair of nodes in network. We propose the method which is adapted from the strategy proposed in [9]. Note our method is based on directed graph, while [9] is based on undirected graph.

2) Time Weight: Suppose that we have known three graph snapshots respectively denoted as G_0, G_1, G_2. Now the prediction of G_3 can be known using several methods. The existing methods usually predict G_3 based on G_2 without consi-dering the existence time or the frequency of links. In figure 1, the occurrence probability of $e(v_1,v_2)$ in G_3 is relatively low only if G_3 is considered, otherwise if considering the structures of G_0, G_1 and G_3, the contact probability of $e(v_1,v_2)$ in G_3 is relatively high. Consequently, the time weight is defined as $Tw(t)$ in Equation (2) for giving

prominence to the importance of network structures at previous time stamps.

$$Tw(t) = e^{-\psi(T-t)} \tag{2}$$

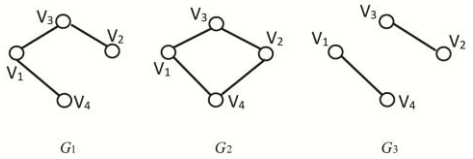


Figure 1. An example of time weight.

3) Change Degree of CN: The algorithm based on common neighbor (CN) has several advantages, such as simplicity, efficiency and good performance. However, most of the proposed algorithms based on CN only consider the CN within one hop. In figure 2, there is not CN between node v_1 and v_2 if considering CN within one hop, but the contact probability of $e(v_1,v_2)$ is not small. In this paper, in order to improve the prediction accuracy, we redefine the concept of CN that considering CN within two hops. Thus there are two CNs between node v_1 and v_2 , respectively, node v_3 and v_4.

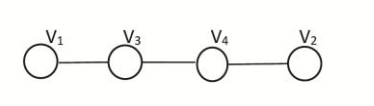


Figure 2. An example of CN.

For a pair of nodes (v,w), we can acquire their CN set denoted as $CN_t(v,w)$ at time t. Thus a sequence of CN sets can be denoted as $\{CN_t(v,w)\}_{t=0,\ldots,T}$ in the period of the evolving network. Intuitively,$CN_t(v,w)$ will change constantly over time. Thus the change degree of CN should be considered, since it has an impact on the link prediction. Then the impact is defined $CD_t(cn)$ in Equation (3), where cn is a CN, $ed_{i-1,i}$ is the Euclidean distance of cn between time t-1 and t. In detail, for each $v \in cn$, Euclidean distance indicates the absolute difference of its degree (in-degree and out-degree included) at time $t=i$ and $t=i$-1.

$$CD_t(cn) = \frac{t}{\sum_{i=2}^{t} ed_{i-1,i}} \tag{3}$$

4) Closeness Between CNs: In figure 3, there are the same CNs between node v_1 and v_2 in graph G and G' , respectively, node v_3, v_4 and v_5. However, the relationship of CN is more intimate or complex than CN'. So the probability of establishing a link between node v_1 and v_2 in CN is relatively higher. For example, suppose two people are not friends but have many mutual friends, if relationship of those mutual friends is very intimate or close, they will have great opportunity to meet and become friends.

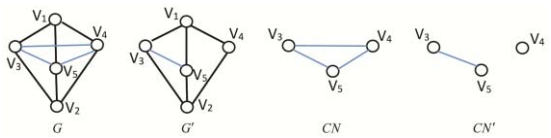


Figure 3. An example of closeness between CNs.

Based on the above idea, the closeness between CNs at time t is defined $CB_t(v,w)$ in Equation(4), Where ACN_t can be denoted as $ACN_t = \{e(v,w) \mid v,w \in CN_t(v,w)\}$, and the size of ACN_t is $|ACN_t|$.

$$CB_t(v,w) = \ln(|ACN_t|) \tag{4}$$

5) Graph Prediction Based on CN: In this section, we will describe the graph prediction algorithm.

Algorithm 2 Graph Prediction Based on CN

Input: $G^t{}_{t=0,\ldots,T}$, P^t
Output: G^{T+1}

```
1:  G^{T+1} ⩽ G^T
2:  procedure GPBaseCN(G^t_{t=0,...,T}, P^t)
3:      for all v ∈ P^t do
4:          PN_t(v) = N(v)
5:              for all w ∈ PN_t(v) do
6:                  P_{T+1}(v, w) = LPBaseCN(v, w)
7:                  Update G^{T+1}
8:      return G^{T+1}
9:  procedure LPBaseCN(v, w)
10:     for t = 1; t <= T; t++ do
11:         CN_t(v, w) = N(v) ∩ N(w)
12:         sumWt = Σ_{u ∈ CN_t(v,w)} CD_t(u)
13:         cb = CB_t(v, w)
14:         h_t(v, w) = cb . sumWt
15:     end for
16:     H_{T+1}(v, w) = Σ_{t=1}^{T} Tw(t) . h_t(v, w)
17:     P_{T+1}(v, w) = Σ_{a,b ∈ AN_T(v,w)} H_{T+1}(a, b)
18:     return P_{T+1}(v, w)
```

The Graph Prediction Based On CN is called GPBaseCN and Link Prediction Based On CN is called LPBaseCN. They are detailed in Algorithm 2. Moreover, for each $v \in P^t$ where P^t are probed nodes chosen by using Algorithm 1, we can attain neighbors of v within two hops (Lines 3-4). Then Line 6 calculates the probability of establishing a link between v and w using LPBaseCN and updating predicted graph (Line 7). In LPBaseCN procedure, we can calculate CN,closeness between CNs, change of CN's degree, time weight and similarity value between two nodes (Lines 10-15). Line 17 calculates the occurrence probability of $e(v,w)$ at time $t=T+1$, and finally returns it (Line 18).

C. Selecting The Most Influential Nodes

The algorithm is used to select the most influential nodes in a series of network snapshots.

Given a static social network G^t at time t, some nodes can be selected by using algorithm 1 (Line 3), then a prediction graph G^{t+1} can be produced based on G^t by using algorithm 2 (Line 4). Thus a sequence of prediction graphs $\{G^t\}_{t=1,\ldots,T}$ can be produced. Then a sequence of most influenced nodes can be produced by using Degree Discount Algorithm (DDA) on $\{G^t\}_{t=0,\ldots,T}$ (Lines 5 and 7). Finally, a sequence of seed sets are found in $\{G^t\}_{t=0,\ldots,T}$ (Line 8), then the seed set S_k of size k is

selected (Line 9) by finding the node with most highest number of occurrences from $\{S^t\}_{t=0,...,T}$.

Algorithm 3 Top-k Most Influential Nodes Selecting

Input: G^0 , T, k , K , I , α

Output: top-k node set S_k

1: initialize $S_k = \phi$;
2: **for** $t = 0$; $t <$ T; t ++ **do**
3: $P^t = \text{MGS}(G^t, K, I, \alpha)$;
4: $G^{t+1} = \text{GPBaseCN}(G^i{}_{i=0,...,t}, P^t)$;
5: $S^{t+1} = \text{DDA}(G^{t+1}, k)$;
6: **end for**
7: $S^0 = \text{DDA}(G^0, k)$;
8: obtain the set sequence $S^0, S^1, ..., S^T$;
9: select top-k nodes S_k with the highest number of occurrences ;
10: **return** S_k ;

IV. Experimental Results

In this section, we first describe the data sets used in our extensive experiments, then describe the experiment setup. Finally our experimental results are plotted by line graphs, what's more, detailed analysis is conducted.

A. Data Sets

We conduct extensive experiments to validate our proposed method on two real-life and one synthetic social networks.

1) Synthetic network: According to the idea of preferential Attachment [10], we can study a dynamic graph model, then generate a synthetic social network. Firstly, a random network G^0 with 1,000 nodes can be generated based on Erdös-Rényi model and the probability of establishing a link between each pair of nodes is 0.02. Then for each of time stamps, at first we randomly choose 200 edges with equal probability, then we change their head respectively to 200 nodes which are randomly chosen with probability proportional to their in-degree. Finally 10 time stamps for synthetic network are generated by this method.

2) Real-life networks: We choose wisely two real-life dynamic social networks with time stamps from the Koblenz Network Collection[1]. After selection, Facebook[2] and YouTube[3] are suitable. In detail, Facebook has 63,731 vertices and 1,634,070 edges, YouTube has 3,223,585 vertices and 18,750,748 edges. Moreover, we count degree sequence, then calculate probability of each degree in Facebook and YouTube. Thus in figure 4, the left figure shows the degree distribution of Facebook and the right figure is another.

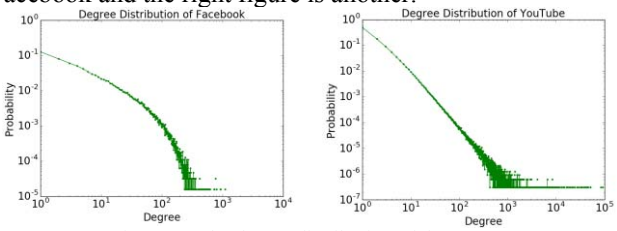


Figure 4. The degree distribution of data sets

B. Experiment Setup

We use Independent Cascading Model (IC) [2] with a probability p in [0,1] of each edge e. Then we set the propagation probability $p(e) = 1/d_G^-(v)$ where v is the node which e points to and $d_G^-(v)$ denotes the in-degree of v. This method of set propagation probability is widely adopted in [11,12,13,14].

For verifying our proposed method, a comparison is made between the following algorithms.

1) Best：A sequence of graph snapshots are fully known, then for each $\{G^t\}_{t=1,...,T}$, the most influential nodes S^t could be selected using Degree Discount Algorithm and influence spread of S^t is estimated by Monte Carlo Simulation. So the influence spread is maximum at each time stamp.

2) Baseline：A graph snapshot G^0 is only known and the most influential nodes S could be chose by Degree Discount Algorithm. Then S still be treated as the most influential nodes at following time stamps.

3) Ours：The algorithm proposed in Section Ⅲ.

C. Performance and Analysis

1) Analysis of MGS algorithm: We test MGS, Rand and AllNP on 4 Amazon network snapshots[4], which is batted on March 2, March 12, May 5 and Jun 1, 2003. Note Rand algorithm is used to choose b nodes to predict network snapshot at next time stamp, and AllNP algorithm is used to predict network snapshot using all nodes. While AllNP run in $\Theta(n^T)$, a huge amount of time is spent in large network. So we extract a subgraph with 1,000 nodes from Amazon network snapshots and choose a subset of nodes of size 50. As shown in Figure 5 , it can be observed that influence spread of MGS is higher than Rand (20.1%) and lower than AllNP (8.85%). Even though MGS is a little worse than AllNP,it is much (665) faster than AllNP in Table 1.

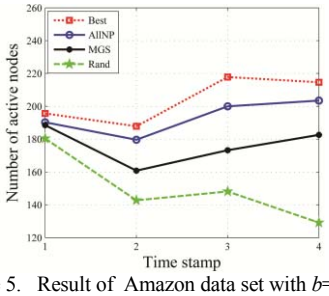


Figure 5. Result of Amazon data set with b=400

[1] http://konect.uni-koblenz.de/networks/
[2] http://konect.uni-koblenz.de/networks/facebook-wosn-links
[3] http://konect.uni-koblenz.de/networks/youtube-u-growth
[4] http://snap.stanford.edu/data/

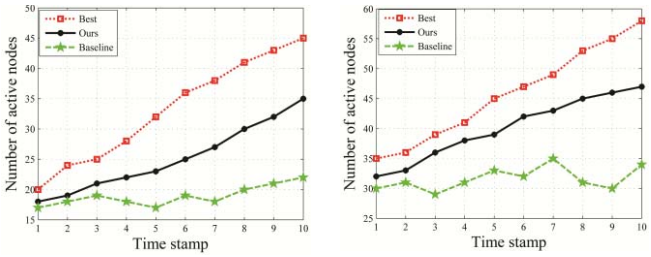
TABLE I. Running Time

Algorithm	Time
Best	2964 ms
ALLNP	56508557 ms
MGS	84880 ms
Rand	65556 ms

2) Performance Comparison: We respectively test Best, Ours and Baseline on all the three data sets. As shown in Figure 6,

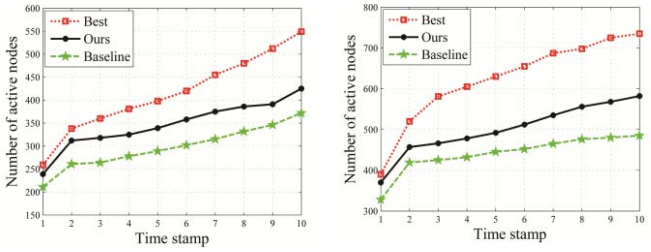
for Synthetic data set, we set the size of top-k most influential nodes set $k=20$ and $k=40$ respectively, then set the window size to 10. For Facebook (Cf. Figure 7) data set, we set $k=50$ and $k=100$ respectively, then set the window size to 10. For YouTube (Cf. Figure 8) data set we set $k=50$ and $k=100$ respectively, then set the window size to 20.

The results on influence diffusion of the selected top-k most influential nodes for each time stamp in three data sets are shown in Figure 6, 7, 8. Then we can easily find that the performance of our method is better than Baseline and sightly worse than Best over all data sets. In detail, our proposed method results in better influence diffusion (12.5%) compared with Baseline and worse influence diffusion (9.4%) compared with Best averaged over all data sets. Note that on Figure 8 , three curves all fluctuate up and down below the y axis. It is because the network structure changes in the following time stamps, for example, new edges are added or old edges are lost.



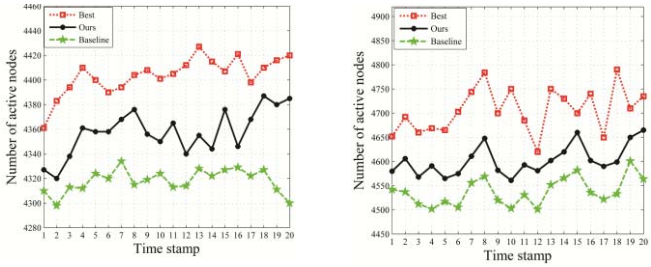
(a) k =20 (b) k =40

Figure 6. Results of Synthetic data set



(a) k =50 (b) k =100

Figure 7. Results of Facebook data set



(a) k =50 (b) k =100

Figure 8. Results of YouTube data set

V. Conclusion And Future Work

In this paper, we study a novel and challenging problem, i.e. Top-k Most Influential Nodes Selecting problem. While the problem has been widely researched in a static network, it will be much more difficult and challenging when considering the network structure rapidly changing and evolving. This paper proposes an efficient approach to select top-k most influential nodes when a graph snapshot is only known in evolving social network. Experimental evaluations show that our proposed method results in better influence diffusion compared with Baseline method which ignores network evolution and only considers one graph snapshot. We may also consider machine leaning approach for predicting the social network evolution to get more accurate results.

References

[1] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In Proc. the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 61–70, 2002.

[2] D. Kempel, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. ACM SIGKDD, pp. 137-146,2003.

[3] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos,J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In KDD, pages 420-429, 2007.

[4] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In KDD, pages 199-208, 2009.

[5] H. Zhuang, Y. Sun, J. Tang, J. Zhang and X. Sun. Influence maximization in dynamic social networks. In ICDM, pages 1313-1318, 2013.

[6] X. Chen, G. Song, X. He, and K. Xie. On infuential nodes tracking in dynamic social networks. In SDM, pages 613-621, 2015.

[7] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng. StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization. In CIKM, pages 509-518, 2013.

[8] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In KDD, pages 1039-1048, 2010.

[9] Yao L, Wang L, Pan L, et al. Link Prediction Based on Common-Neighbors for Dynamic Social Network. Procedia Computer Science, 2016. 83:82-89.

[10] A.-L. Baraba´ si and R. Albert. Emergence of scaling in random networks. Science, 286(5439):509 – 512, 1999.

[11] W. Chen, C. Wang and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social network. In KDD, pages 1029-1038, 2010.

[12] C. Wang, W.Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social network. Data Min. Knowl. Discov. 25(3):545-576, 2012.

[13] A. Goyal, F. Bonchi, and L.V.S. Lakshmanan. A data-based approach to social influence maximization. PVLDB, 5(1):73-84,2011.

[14] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social network. In ICDM, pages 918-923, 2012.

[15] Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68 – 73.

[16] O.Liu, K.L.Man, W.Chong. Social network analysis using big data. In IMECS, vol 2, 2016.

[17] Ohsaka N, Akiba T, Yoshida Y, et al. Dynamic influence analysis in evolving networks[J]. Proceedings of the VLDB Endowment, 2016, 9(12):1077-1088.

[18] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal samlping algorithms for viral marketing in billion-scale networks. In SIGMOD, pages 695-710, 2016.