2018 18th IEEE International Conference on Communication Technology

Time-Sensitive Influence Maximization in Social Networks

Min Hu*, Qin Liu†, Hejiao Huang*, Xiaohua Jia*§
*Department of Computer Science, Harbin Institute of Technology, Shenzhen, China
†Department of Computer Science, Wuhan University, Wuhan, China
§e-mail: csjia@cityu.edu.hk

***Abstract*—A lot of people have been concerned about the problem of maximizing influence in social networks, which is aimed to find a set of nodes to get the influence spread maximized. However, the existing reasearches mainly focus on that a node influences its neighbors once without considering time and cost constraints. But in real world, people often try to influence their friends repeatedly during a time interval. Sometimes, the spread of information will cost a certain price as well. In this paper, we study the Time-sensitive Influence Maximization Problem and propose a Time and Cost constrainted Influence model with users' Online patterns (TCIO model) . In TCIO model, the selection of seed nodes is limited to the budget and each node can influence its neighbors repeatedly according to their online patterns with different probability until a given message expire time is reached. We then show that the problem is NP-hard and our model satisfies monotonicity and submodularity for influence spread. Based on this, we develop a greedy algorthm to solve the problem. To reduce the computation complexity and optimize seed node selection with cost, we propose an efficient method GMAI for approximately calculating added influence using influence weight. Our experiments show that our model is effective and practical since it takes into account time factors, and GMAI faster and more effecient than other evaluated algorithms.**

Keywords-social networks; time-sensitive influence maximization; time and cost constrainted

I. Introduction

With the rapid development of information technology, there are more and more popular online social networks, such as QQ and Facebook [1]. It provides great opportunities for marketing new products or spreading political agendas in a very short period of time. For example, to produce a new film will select some influential users from social networks, then give them some reward to ask them to arouse the attention of a lot of people by utilizing the Word-of-Mouth effect in social networks [2]. When using viral marketing, we usually try to find some people with high influence to disseminate information so that we can have more people being reached. However, there are some difficulties in solving the influence maximization problem [3]–[6] that aims to find the most influential ones. Intensive effort has been dedicated to the research of finding a small subset of influential individuals in a social network to make the influence maximized [4].

Kempe et al. [3] first formally defined the problem of maximization of influence. There are several kinds of models considered by Kempe et al. and other researchers, including the Independent Cascade (IC) and the Linear Threshold (LT) model. In addition, they proved this problem NP-hard under both the two models and proposed a greedy algorithm.

Based on the IC model and LT model, some solutions have been proposed to improve the efficiency and scalability of the algorithms to solve the problem. However, few of them take into account the factors of time. In real practice, when an enterprise publishes a new product, it often needs to spread information within a given time and total budget. Also in the process of information dissemination, whether a user will be influenced heavily depends on his online patterns. In the recent study, it's revealed that time plays an important role in information diffusion [7].

In this paper, we study time-sensitive influence maximization problem. We propose a Time and Cost constrainted Influence model with users' Online patterns called TCIO model. We consider the online patterns of users along the time sequence. We propose an algorithm that employs a greedy method using simulations to solve the time-sensitive influence maximization problem. Considering that the computational complexity of the greedy method is too high, we further propose an algorithm based on added value of influence.

This paper is organized as follows. After describing related work, in Section III, we define the Time-Sensitive Influence Maximization Problem and propose the TCIO model. Section IV presents a simulation based method to compute exact influence spread and a simple greedy algorithm under the TCIO model. In order to reduce computation complexity, we proposes an efficient method. Section V illustrates the experiment results, and Section VI concludes this paper.

II. Related Work

Domingos and Richardson [5] first raised the influence maximization as an algorithmic problem and proposed a probabilistic solution. Kempe et al. [3] first formally defined the problem of maximization of influence. In addition, they proposed a greedy algorithm, guaranteeing that the influence degree is within (1- 1/e) of the optimal influence degree.

However, the simulation based greedy algorithm is computationally expensive and cannot scale well in large social networks [8], [9]. Some afterward researchers concentrated on more scalable methods. Leskovec etc. [8] presented an optimal greedy algorithm by reducing the computation of calculating the influence spread which is named cost-effective lazy forward (CELF) algorithm. Chen et al. proposed a scalable heuristic method called LDAG for the LT model [10]. In addition, Wang et al. [11] tried to explore the underlying community structure of social networks. Liu et al. [12] put forward a greedy algorithm based on the concept of the influence spread path to improve the scalability of the algorithm.

The study of influence maximizing problem is always based on some kinds of models, among which IC, LT and voter model [13] are three representative models. Although various extensions of those models were proposed recently, none of them considers the crucial factors in real influence diffusion time constraints, total budget, repeated influence, and users' online patterns. Our proposing TCIO model embraces these essential characteristics.

III. System Model and Problem Definition

In this section, we introduce the system model and Time-sensitive Influence Maximization Problem.

TABLE I. Notation Table

Notation	Definition
G(V,E)	Online social network graph
n	$\|V\|$
l	$\|E\|$
S	Seed set
M	Online pattern set
m_{vt}	Online pattern of node v in time slot t
P	Probability of node forwarding set
p_{uv}	Probability of node v forwarding node u
t_0	Message release time
T	Message expire time
c_u	Cost of node u
B	Total budget
$\sigma(S)$	Number of nodes influenced by S with time limit T

A. System Model

We propose a Time and Cost constrainted Influence model with users' Online patterns (TCIO model). Let *G(V,E)* denote a directed social network graph, where V is the set of nodes representing users and E is the set of directed edges representing social relationship between users. Each edge is associated with a forwarding probability p_{uv} . p_{uv} is the probability that node v forwards a message from node u when v is online and u already forwarded the message. The time of a day is divided into time slots and a node can be either online or offline in a time slot according to its online pattern. m_{vt} denotes the online pattern of a node v in time slot t, where m_{vt}=0 if node v is offline in time slot t; m_{vt}=1 for online. Each node has a price when it is chosen as a seed node to spread a message. Let T be the message expire time and t_0 the message release time. Let B be the total budget for selecting seed nodes to spread a message in a social network.

B. Problem Definition

Time-sensitive Influence Maximization Problem is a problem of selecting a set S of nodes, so that in the case of a given message expire time T, total budget B, and the users' online patterns M, when nodes in S publish a message, the number of nodes that eventually receive the message is maximizd. Note that every node that has forwarded the message can influence its neighbors repeatedly according to their online patterns until its neighbors become influenced to forward the message or the expire time is reached.

Given a social network graph *G(V, E)*, message expire time T, message release time t_0 and total budget B, the problem of our concern is to find a seed set $S \subseteq V$ within the budget, so that the expected number of nodes influenced by S within T, $\sigma(S)$, is maximized.

Similar to the conventional influence maximization problem, the time-sensitive version is NP-hard, which is shown in Theorem 1.

Theorem 1. The time-sensitive influence maximization problem is NP-hard.

Proof. The conventional influence maximization problem is known to be NP-complete and can be considered as the corresponding time-sensitive problem with unlimited time bound. We argue that the time-sensitive influence maximization problem is the generalized version of the conventional influence maximization problem, which thus is NP-complete.

IV. Information Diffusion Simulation Based Algorithm

A. Simulation Based Influence Computation

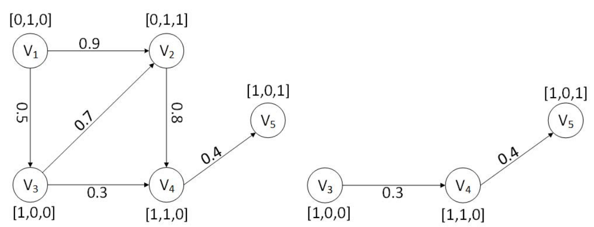
As we consider users' online patterns, in time slot t, we only need to consider the users that are online in time slot t. Let $G_t(E_t, V_t) \subseteq G(E, V)$ be the social network graph that contains only the nodes (users) that are online during slot t, $V_t \subseteq V$. Let C_t denote the set of nodes that have already forwarded the message after slot t. Our algorithm runs through all time slots from t_0 to $t_0 + T$. For an edge $(u, v) \in E_t$, there is a probability p_{uv} indicating the probability that v forwards a message posted by u. In the algorithm, we generate a random number p. If $p_{uv} \geq p$, we let node v forward the message from node u. The nodes that have received and forwarded the message will influence their online neighbors repeatedly along the time sequence. This process ends if there are no nodes that are not influenced or time T is reached.

As seen in Algorithm 1, we use simulation to compute whether a node is influenced in the process of information diffusion. $G_t(E_t, V_t)$ is generated with nodes and edges in time slot t, as is shown in Fig. 1. The initial graph is an online social network. The node in the graph represents a user. The number on the edge between u and v is the forwarding probability for v to forward u's message. For instance, the number on the edge between V_1 and V_2 is 0.9, it means that V_2 has probability of 0.9 to forward V_1's message. The numbers in the brackets near the node is the online pattern of the user. For example, the online pattern of the user V_1 is [0,1,0], it means that the user is only online in the second time slot. The graph of the first time slot shows that only V_3, V_4 and V_5 are online and the edges contain offline nodes are pruned. The graph of other time slot is generated in the same way.

From Fig. 2, we can see V_1 releases a message (Suppose that if the user is offline in the present time slot, it can be forced

Algorithm 1: Simulation-Based Influence Computation

```
Input: G, t_0, T, M, P
Output: σ(S)
t ← t_0;
while t < t_0 + T do
    Generate G_t(E_t, V_t);
    //The edges between the users that have forwarded
    //the message and those online are added.
    for u ∈ C_{t-1} and v ∈ V_t and v ∉ C_{t-1} do
        add edge (u, v) to G_t(E_t, V_t);
    end
    C_t ← C_{t-1};
    for u ∈ C_t and v ∈ V_t and v ∉ C_t do
        generate a random number p between 0 and 1;
        if p ≤ p_uv then
            C_t ← C_t ∪ {v};
        end
    end
    t++;
end
return |C_{t-1}|;
```



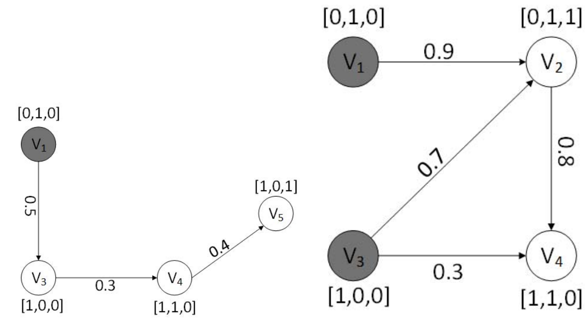
(a) Initial graph (b) The graph of the first time slot

Figure 1. The social network graphs with the users' online patterns

to go online if it needs to be a source of information) and only V_3, V_4 and V_5 are online. It is possible for V_3 to forward the message of V_1, and the probability of forwarding is 0.5, so a random number is generated. Assume that the number is 0.2, because 0.2 is less than 0.5, it means V_3 forwarding the message. Considering that V_4 is also online, we assume the random number 0.5. It means V_4 not forwarding the message for 0.5 is more than 0.3. The process of information diffusion terminates in the first time slot. Then in the second time slot, the new edges in the graph are those between the online users in the second time slot. The edges between the users that have forwarded message in the first time slot and those online in the second time slot are added. The edges that have successfully forwarded massage in the first time slot are removed. The process terminates if there are no nodes that are not influenced or time *T* is reached. When the process terminates, the number of influenced nodes is returned.

B. Simple Greedy Algorithm for Influence Maximization

We propose Algorithm 2 to simulate the time-sensitive influence diffusion along the time sequence and solve the time-



(a) The information diffusion graph of the first time slot (b) The information diffusion graph of the second time slot

Figure 2. The information diffusion graph

sensitive influence maximization problem. Note that Algorithm 2 differs from the algorithm for conventional influence maximization problem, which does not consider time factor.

In order to guarantee that the algorithm has a lower bound of approximation ratio of (1 - 1/e), the TCIO model should have the properties of monotonicity and submodularity.

Theorem 2. The influence function $\sigma(S)$ is monotone and submodular under the TCIO model.

Proof. For a given function $f : 2^v \rightarrow R$ is said monotone if $f(S) \leq$ f$(S^{'})$, $\forall S \subseteq S^{'}$, and submodular if
$f(S \cup \{v\}) - f(S) \geq f(S^{'} \cup \{v\}) - f(S^{'})$, $\forall S \subseteq S^{'}, v \in V$.
$\sigma(S)$ is the number of influenced nodes by *S* within *T* time slots over the determined graph *G(V,E)*. It is equal to the number of nodes reachable from *S* by at least one path. It is easy to see that

$$\sigma(S) \leq \sigma(S \cup \{u\}). \tag{1}$$

Therefore $\sigma(S)$ is monotone.
We have

$$\sigma(S \cup \{u\}) - \sigma(S) = \sigma(S \cup \{u\}) \backslash \sigma(S) = \sigma(\{u\}) \backslash \sigma(S). \tag{2}$$

Since $\sigma(S)$ is monotone and $\sigma(S) \leq \sigma(S^{'})$, $\forall S \subseteq S^{'}$,

$$\sigma(\{u\}) \backslash \sigma(S) \geq \sigma(\{u\}) \backslash \sigma(S^{'}), \forall S \subseteq S^{'}. \tag{3}$$

Thus,

$$\sigma(S \cup \{u\}) - \sigma(S) \geq \sigma(S^{'} \cup \{u\}) - \sigma(S^{'}), \forall S \subseteq S^{'}. \tag{4}$$

Therefore, $\sigma(S)$ is submodular.
In summary, $\sigma(S)$ is both monotone and submodular under the TCIO model.

Because $\sigma(S)$ is monotone and submodular by Theorem 2, the greedy strategy in Algorithm 2 can achieve approximation ratio of (1 - 1/e) according to [14].

Algorithm 2 (SimpleGreedy algorithm) is based on mountain climbing greedy algorithm. In order to reduce the computation error, we perform R iterations and use the average

Algorithm 2: Simple Greedy Algorithm for Influence Maximization

Input: G, S, t_0, T, M, P, R
Output: S
$S = \emptyset$;
while $c \leq B$ **do**
  $SS \leftarrow V - S$;
  for *every* $u \in SS$ **do**
    $S^{'} \leftarrow S \cup u, u \in SS$;
    while *the number R of iterations is not reached* **do**
      calculate $\sigma(S^{'})$ the nodes influenced by seed nodes using Algorithm 1;
    end
    calculate the average $A_u\, of\, \sigma(S^{'})$ of R iterations;
  end
  $v \leftarrow arg\, max_u(A_u)$;
  $c \leftarrow c + c_v$;
  $S \leftarrow S \cup \{v\}$;
end
return S;

influence to determine which node should be selected as a seed node. R is a given number of simulated information diffusion times. Algorithm 2 stops when the total budget is reached.

C. Greedy on Maximal Added Influence

Algorithm 1 is a simulation-based influence computation under the TCIO model. We then propose Algorithm 2, which is a simple greedy algorithm to solve the time-sensitive influence maximization problem. It doesn't have a performance guarantee in the worst case. We further design an improved algorithm based on maximal added influence.

Algorithm 3 (GMAI algorithm) uses influence weight to calculate added influence in the information diffusion process. For the influence function is monotonous and submodular [14], the algorithm repeatedly adds the node that generates the largest added influence to the seed set S. We can choose seed node according to its approximate added influence gain.

$\sigma(S)$ is the expected number of influenced nodes by seed node set *S*. We can compute added influence by using $\sigma(S \cup u) - \sigma(S)$. Every time we select a seed node, we compute current added influence. It takes lots of time and space in large social networks. Considering $\sigma(S \cup u) - \sigma(S)$ is no larger than $\sigma(\{u\})$, we use simulation result to approximate the added influence.

$$\sigma(S \cup \{u\}) - \sigma(S) \approx \sigma(\{u\})\alpha, \tag{5}$$

$$\alpha = \frac{\sum\limits_{(u,v) \in E} \sigma(\{v\}) p_{uv} \prod\limits_{w \in S} (1 - p(w, v, t))}{\sum\limits_{(u,v) \in E} \sigma(\{v\}) p_{uv}}, \tag{6}$$

where *k* is the number of seed nodes, p_{uv} is the probability that *v* forwards a message from *u*, *p(w,v,t)* the probability that *v* forwards a message from the seed node w before time slot *t*. *p(w,v,t)* can be computed in Algorithm 1, which records how the neighbors of a node are influenced by the seed nodes. The main idea of this equation is that the greater the possibility that the neighbors of a node are influenced by the seed nodes in the process of disseminating information, the greater the discount on the added influence of this node.

Since $\sigma(S)$ is submodular, in our formulas, $\sigma(S \cup \{u\}) - \sigma(S) \leq \sigma(\emptyset \cup \{u\}) - \sigma(\emptyset) = \sigma(\{u\})$. α is the influence weight computed by formula 6 to approximate the added influence.

The influence $\sigma(\{u\})$ of every node *u* is computed by the simulation using Algorithm 1. We approximate the added influence of a node one by one in every node's simulation. Algorithm 3 is proposed with this added influence spread approximation. Algorithm 3 first computes every node's influence by using Algorithm 1 and records them all. Then it can select the first seed node which has the most influence as the first seed node. After this, it chooses seed node one by one utilizing the formula 5 and 6 till the total budget is reached. We have made some optimizations that we select the biggest of the ratios of the added influence to the cost of every node in order to get the best performance cost-effective seed nodes within the budget.

Algorithm 3: Greedy on Maximal Added Influence

Input: G, t_0, T, B, M, P
Output: S
S=$\emptyset$;
for *every* $v \in V$ **do**
  calculate $\sigma(\{v\})$ using Algorithm 1;
end
$u \leftarrow arg\ max_v \sigma(\{v\})$;
$S \leftarrow S \cup \{u\}$;
while $c \leq B$ **do**
  for *every* $v \in V \backslash S$ **do**
    compute v's added value $\mu(v)$ according to equation 6;
  end
  $u \leftarrow arg\ max_v(\mu(v)/c_v)$;
  $c \leftarrow c + c_u$;
  $S \leftarrow S \cup \{u\}$;
end
return S;

V. Experiments

TABLE II. Basic Information of Real Social Networks Dataset

Networks	Email	Wiki-vote
Number of nodes	1K	7K
Number of edges	10K	103K
Average clustering	0.2202	0.1409

A. Experimental Setup

Datasets. The real-world social networks are used in the experiments, which are also widely used in previous work on influence maximization. The basic information about them are listed in Table II. The first is a network that the members of Virgili University Rovira send mails, called Arenas/email [15]. The second one (Wiki-vote [16]) is a Wikipedia voting network where nodes represent wikipedia users and an edge from node i to j represents that user i voted on user j.

Evaluated Methods. The experimental study is to verify the capability of our methods for time-sensitive influence maximization problems. The methods proposed in this paper are based on the greedy algorithm framework under our TCIO model. Also we approximately calculate the added influence to select the seed nodes. The following methods are evaluated.

- SimpleGreedy: For time-sensitive influence maximization problem, we use simulation to calculate influence spread $\sigma(S)$ by 20000 times of simulations.
- GMAI Algorithm: It uses approximate added influence spread to select seed nodes, combination of Algorithm 1 and Algorithm 3.
- Random: It simply selects k random vertices in the graph, which is also evaluated in [3].
- MaxDegree: It selects k nodes in non-increasing order of nodes out-degree, which is a simple heuristic algorithm.
- SingleDiscount:It discounts the degree of each neighbor of a newly selected seed by one.

To reduce the expensive computation of $\sigma(S)$, we apply a restriction to limit the length of path no larger than T in every time slot , which are not or less related to our problem. And the forwarding probability p_{uv} between nodes is randomly generated by using (0,1) uniform distribution.

As for the cost allocation method of nodes, we count the outdegree d_v of each node v and the maximum value M=$max_v(d_v)$. We use two distributions in our experiment.

- Uniform distribution: the cost c_v of any node v in a network is randomly generated from an interval $(d_v/M, 2d_v/M)$ according to uniform distribution .
- Normal distribution: the cost c_v of any node v in the network is generated randomly from the expectation $\mu = d/M$ and the variance $\sigma^2 = d_v/(6 * M)$ according to normal distribution.

The process of getting online patterns: First, download the distribution of Internet time of Internet users on the PC platform from the Baidu statistics from an interval 2017.09-2017.12 [17]. We set the maximum number of people online to 40%, and the percentages of the number of people online in other time slots are determined, as is shown in Fig. 3:

All algorithms are implemented in JAVA and compiled with java version "1.8.0_144" on a Linux server with an 8-core Intel(R) Xeon(R) 2.0 GHz CPU and 32 GB memory.

B. Experimental Results

We demonstrate the experimental results on real world social networks datasets in this section.

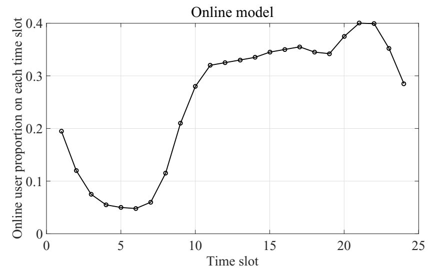
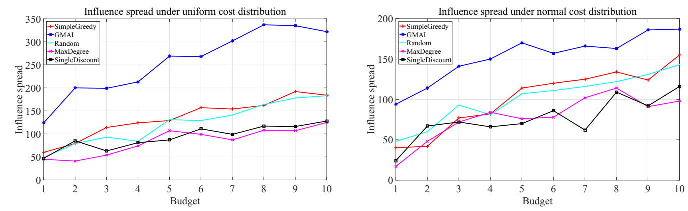
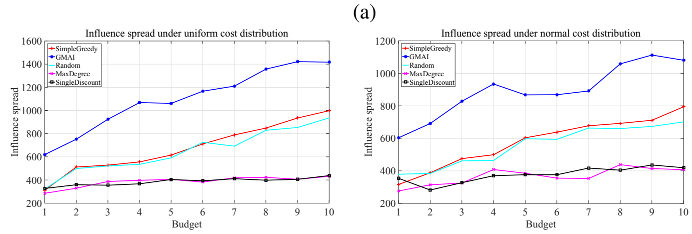


Figure 3. Online pattern.



(a)



(b)

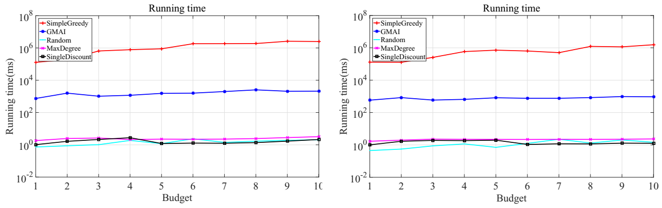
Figure 4. The influence spread of the evaluated algorithms when the budget changes under uniform and normal cost distributions (t_0=6, T=3). (a) Eamil. (b) Wiki-vote.

1) Influence Spread: We are unable to get the result in polynomial time, for the time-sensitive influence maximization problem is NP-hard. After obtaining the seed nodes, we apply 20, 000 simulations and use the average influenced nodes as the influence spread of the seed set.

Fig. 4 illustrates the results of influence spread on the datasets with T=3, t_0=6 for different budgets. It is obvious that influence spread increases as budget increases on every dataset. GMAI method achieves better influence spread than the other evaluated algorithms no matter what cost distribution it is, which verifies our proposed methods effective to solve the time-sensitive tnfluence maximization problem. For different t_0 and T, the results maintain the same because the only difference is that the different number of people online leads to different influence spread, which are not included in this paper due to the limited space.

SimpleGreedy algorithm is a greedy method based on simulation selecting the node with max influence as the seed node. MaxDegree and SingleDiscount achieve almost the same influence spread but less than SimpleGreedy. And the Random results in poor performance while the budget is small, but it can obtain better results than MaxDegree and SingleDiscount sometimes, because maybe two heads are better than one.

2) Running Time: Fig. 5 shows the running time of the evaluated algorithms with T=3, t_0=6 on the Email dataset



(a) Uniform distribution (b) Normal distribution

Figure 5. The running time of the evaluated algorithms when the budget changes under different cost distributions (t_0=6, T=3).

under different cost distributions. The running time of GMAI increases as budget increases but still less than SimpleGreedy, while Random, MaxDegree and SingleDiscount run faster.

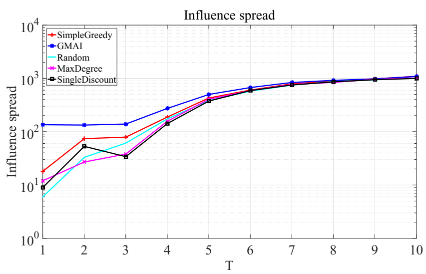


Figure 6. The influence spread of the evaluated algorithms when T changes under uniform cost distributions (t_0=6, B=1).

3) Effect of T on Influence Spread: To investigate the impact of T on the influence spread, Fig. 6 shows the influence spread returned for different values of T with B=1, t_0=6 on the Email dataset under uniform cost distribution. Different time delays have significant differences on the effect of seed set maximization. When T is small, it is obvious that GMAI has better influence spread. Due to the properity of repeated influence of our TCIO model, if T is big enough, great influence spread can also be obtained by other algorithms. But in real world, the spread of information is usually carried under certain time constraint. We believe that time constraints play an important role in the maximization of influence.

VI. Conclusion

In this paper, we proposed the Time and Cost constrainted Influence model with users' Online patterns (TCIO model) in social networks. Conventional models did not consider time delay, cost and users' online patterns in real world. Our model considers all these factor in the process of information diffusion. Because of the monotonicity and submodularity of our model, we proposed a greedy algorithm, which achieves (1 - 1/e) approximation ratio. Since this greedy algorithm is computationally expensive and is not scalable for large social networks. We further proposed GMAI Algorithm to solve the problem more efficiently. Experimental results on the real datasets show that GMAI runs faster than the SimpleGreedy algorithm. GMAI Algorithm also achieves a better influence diffusion than other evaluated algorithms.

Acknowledgment

This work is financially supported by National Science and Technology Major Project under Grant No.2016YFB0800804 and No.2017YFB0803002, National Natural Science Foundation of China under Grant No.61672195 and No.61732022.

References

[1] Masahiro Kimura and Kazumi Saito. Tractable models for information diffusion in social networks. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 259–271. Springer, 2006.

[2] Vijay Mahajan, Eitan Muller, and Frank M Bass. New product diffusion models in marketing: A review and directions for research. In *Diffusion of technologies and social behavior*, pages 125–177. Springer, 1991.

[3] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[4] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming*, pages 1127–1138, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[5] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

[6] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011.

[7] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[8] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[9] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[10] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 88–97. IEEE, 2010.

[11] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1039–1048. ACM, 2010.

[12] Bo Liu, Gao Cong, Dong Xu, and Yifeng Zeng. Time constrained influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 439–448. IEEE, 2012.

[13] Peter Clifford and Aidan Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.

[14] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsłi. *Mathematical Programming*, 14(1):265–294, 1978.

[15] Alex Arenas Website. Arenas/email for social network. http://deim.urv.cat/∼alexandre.arenas/data/welcome.htm. Alex Arenas Website.

[16] Stanford Large Network Dataset Collection. Wiki-vote for social network. http://snap.stanford.edu/data/. Stanford Large Network Dataset Collection.

[17] Baidu statistics. online-time for social network. http://tongji.baidu.com/data/hourx. Baidu statistics.