Influence Maximization in Dynamic Social Networks

Honglei Zhuang*†, Yihan Sun*, Jie Tang*, Jialin Zhang‡ and Xiaoming Sun‡
*Department of Computer Science and Technology, Tsinghua University
†Department of Computer Science, University of Illinois at Urbana-Champaign
‡Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences
hzhuang3@illinois.edu, syhlalala@gmail.com, jietang@tsinghua.edu.cn, zhangjl2002@gmail.com, sunxiaoming@ict.ac.cn

***Abstract*—Social influence and influence diffusion has been widely studied in online social networks. However, most existing works on influence diffusion focus on static networks. In this paper, we study the problem of maximizing influence diffusion in a dynamic social network. Specifically, the network changes over time and the changes can be only observed by periodically probing some nodes for the update of their connections. Our goal then is to probe a subset of nodes in a social network so that the actual influence diffusion process in the network can be best uncovered with the probing nodes. We propose a novel algorithm to approximate the optimal solution. The algorithm, through probing a small portion of the network, minimizes the possible error between the observed network and the real network.**

We evaluate the proposed algorithm on both synthetic and real large networks. Experimental results show that our proposed algorithm achieves a better performance than several alternative algorithms.

I. Introduction

Social influence, the phenomenon that the actions of a user can induce her friends to behave in a similar way, is a subtle force that governs the dynamics of social networks [5]. For example, a company wants to market a new product through the effect of "word of mouth" in the social network. It wishes to find and convince a small subset of users (seed users) to adopt the product so as to trigger a large cascade of further adoptions via social influence. Fundamentally, we need to understand the influence diffusion by answering questions such as: how to select the seed users so that the total number of triggered users to adopt the product can be maximized (a.k.a. influence maximization).

Recently, with the rapid development of online social networks such as Facebook, Twitter, and Google Plus, a bulk of research has been conducted for studying the influence diffusion in social networks. For example, Richardson [14] and Kempe et al. [9] formally defined the problem of influence maximization. Chen et al. [4] presented an efficient algorithm to solve the problem. Goyal et al. [7] leveraged real propagation traces to derive more accurate influence models. However, most existing methods do not consider the temporal information. In [8], the authors presented the problem of MINTIME for influence maximization. In MINTIME, an influence spread threshold η and a budget threshold k are given, and the task is to find a subset of size at most k such that by activating it, at least η nodes are activated in expectation in the minimum possible time. Chen et al. [3], instead, explored time-critical influence maximization in social networks with a deadline constraint τ. The goal is to choose a budgeted subset to activate, which could maximize the influence spread at τ time stamps later. However, their work assumes that the dynamic network is fully observed. This is unrealistic in many real situations. For example, there are more than 3 million following relationships newly added and 3 million removed from Weibo network every day. It is difficult to provide a fully observed network at any moment.

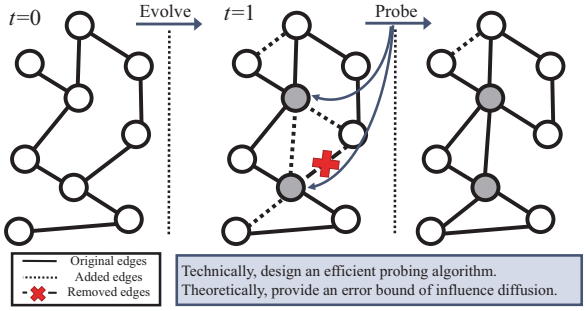


Figure 1: An example of probing influence diffusion in a dynamic social network. The left figure presents the given network at $t = 0$; the middle figure describes the process of probing in the network at $t = 1$; the right figure illustrates the partially observed network.

In this paper, we study the problem of influence maximization on dynamic social networks. An example is illustrated in Figure 1. Particularly, the scenario setting is that we are given a network G^0 at the first time stamp, but can only observe the changes of network by periodically probing a small subset of its nodes for their connections in the following time stamps. After the probing at time t, we obtain a partially observed network $\hat{G}^t$. Different probing strategies may result in different observed networks. Our ultimate objective is to study influence maximization [9] on dynamic social networks. Thus, an ideal solution for probing is that the influence diffusion in the observed network $\hat{G}^t$ is exactly the same as that in the fully observed network G^t. The challenge here is to design a probing strategy to

maintain an observed network on which the solution of influence maximization is as close to the solution on the real network as possible.

The problem is different from the settings in existing work and poses a set of unique challenges. First, as the network changes can be only detected by probing, it might be necessary to adjust the classical influence diffusion algorithm to the dynamic settings. Second, it is important to design an efficient algorithm to select the subset of nodes to probe so as to benefit the performance of influence diffusion.

To address the above challenges, employing influence maximization as an example, we propose a novel algorithm for probing influence diffusion. We propose Maximum Gap Probing (MaxG) algorithm which aims to maximize the change of solution achieved by probing. We define the "maximum possible performance gap" formally, and provide a method to explicitly estimate its value.

We test the proposed algorithm on both synthetic and real large networks. In synthetic network, we evaluate how the proposed algorithm approximates the "true" solution to the influence maximization. In two real networks, Twitter and Coauthor, we also compare the proposed algorithm with several baseline approaches. MaxG achieves a significant better performance than the baseline algorithms.

II. Preliminaries

We first introduce the concept of probing in dynamic networks. Specifically, we assume a discrete notion of time t. The directed social network at time t is denoted by $G^t = (V^t, E^t)$, where V^t is the set of nodes and $E^t \subset V^t \times V^t$ is the set of edges included in the social network at time t. Notation $\{G^t\}_{t=0,\cdots,T}$ defines the sequence of networks over time. We suppose that the network at $t = 0$ is known, while the change of the network over time can only be observed by *probing*. When an algorithm (or a user) *probes* a node, it can see the current connections of this node. The algorithm can keep a partially observed network $\hat{G}$, by probing a fixed number of b nodes at each time stamp, and update their connections. If the given b is large (with an extreme case $b = |V|$), then the observed network should be almost the same as the fully observed network. However, if b is a small number, the design of probing strategy can be very challenging. First, we need to probe nodes that are most likely to change so as to keep the observed network as close to the real network as possible. Second, regarding specific task that will be implemented on the observed network, the probing strategy should leverage the importance of each node in the task as well.

Before we study influence maximization on dynamic networks, we briefly review the influence maximization problem on the static networks. Suppose we are given a social network $G = (V, E)$, where V is the set of nodes (users) and $E \subset V \times V$ is the set of edges. Notation $e_{ij} \in E$ represents a directed relationship from node v_i to v_j. Let each node have a binary status (active or inactive) and p_{ij} quantify the influence probability of v_j on v_i. When a node v_j is active, we say that the node has a probability p_{ij} to influence its neighbor v_i to be active[1]. For simplification, we assign uniform probability p to all the p_{ij}. Influence maximization is to find a small subset of nodes (seed nodes) in a social network that could maximize the spread of influence. In particular, it aims to find a subset S of nodes with size k and activates all nodes in the set. Hopefully these nodes can spread their influence to activate other inactive nodes. The influence spread process continues until no more inactive nodes can be activated. A good solution to influence maximization is to find the seed set S that can maximize the number of activated nodes after the influence spread process. The problem is proved to be NP-hard [9] and a number of algorithms have been proposed to approximate the optimal solutions such as [4].

Based on the classical influence maximization problem, we propose the problem of probing influence diffusion in dynamic networks. Our goal is to design a probing strategy so that the solution of influence maximization obtained based on the observed network at time stamp t can trigger the maximum influence spread on the real network at t. We define the problem as follows:

Problem 1. Influence Maximization in Dynamic Social Networks. *Let G^0 be a network at time $t = 0$. Suppose the network keeps changing over time, and the changes can be known only by probing. Our problem is to find b nodes so that if we probe their connection changes at time t and partially update the observed network, the approximate solution to the influence maximization problem with the observed network will be close to the "true" solution on the fully observed network at time t.*

Unlike the traditional influence maximization problem [14], [9] which aims to find the optimal subset on static networks, we focus on probing strategies to mostly benefit the influence maximization algorithm in dynamic networks.

III. Probing Algorithms

In this section, we introduce an algorithm for probing influence diffusion in dynamic social networks. For the algorithm, except the first time stamp, we do not have a full image of the network at following time stamps. Instead, we keep an observed network $\hat{G}^t$ at each time stamp t. When $t = 0$, the observed network is exactly the same as the real network, i.e. $\hat{G}^0 = G^0$. At each time stamp $t > 0$, we update the connections of probed nodes based on the observed network $\hat{G}^{t-1}$ from previous time stamp and obtain $\hat{G}^t$. Then we perform an influence maximization algorithm on the observed network and output the approximated seed

[1] As in a social network, e_{ij} usually represent v_i follows v_j, which means that v_j is possible to influence v_i.

set S^t. We utilize the degree discount heuristics algorithm proposed in [4] for influence maximization, i.e. to greedily choose the node with the maximum improvement to the influence spread and add it to the seed set, until the seed set reaches the given size k. A simple heuristic function $h_\Gamma(u)$ is defined to estimate the marginal improvement of each node u:

$$\begin{aligned} h_\Gamma(u) &= 1 + p \cdot [d_{in}(u) - r_{in}(u) - r_{out}(u) \\ &- p \cdot r_{out}(u)\,(d_{in}(u) - r_{in}\,(u))] \end{aligned} \quad (1)$$

where Γ denotes the set of nodes that have been chosen as seeds so far; $d_{in}(u)$ is the in-degree of node u; $r_{in}(u)$ and $r_{out}(u)$ are the number of nodes that have already been chosen as seeds in Γ among the predecessors and successors of u respectively. We calculate the heuristic function on $\hat{h}_\Gamma(v)$ on the observed network $\hat{G}$.

A straightforward strategy for probing is to randomly choose nodes with equal probability and probe their connection changes. However, as it treats all nodes equally, it is unlikely to minimize the loss to the performance of influence maximization. An alternative algorithm is Degree Weighted Round-Robin Probing (DegRR). It probes each node with frequency proportional to their observed degrees, e.g. the sum of their in-degrees and out-degrees. Although it considers the different importance of each node in influence maximization tasks, neighborhood of nodes with high degree does not necessarily change frequently.

Our proposed idea is to detect the maximum possible change to the solution of influence maximization so as to minimize the loss between the observed network and the real network. We formulate an objective function and propose the Maximum Gap Probing (MaxG) to optimize it.

Maximum Gap Probing (MaxG) Our intuition is to probe the nodes which are expected to bring the most change of the approximated solution on the observed network, so as to approach the performance of real solution. We define function $\hat{Q}_v(S)$ to evaluate the influence spread of seed set S on the observed network $\hat{G}$ after probing node v. We use S_o and S'_o to represent the optimal seed set for influence maximization obtained before and after probing v respectively. Note that S'_o is dependent on the probed node v and thus can also be written as a function of v, i.e. $S'_o(v)$. Then the difference of solution with and without probing v could be measured by $\left(\hat{Q}_v\left(S'_o(v)\right) - \hat{Q}_v\left(S_o\right)\right)$. We refer to the difference as "performance gap".

We need to estimate the "maximum possible performance gap" for each node. More precisely, for a given probability ϵ as "tolerance", indicating that we only consider the performance gap with the occurrence probability not lower than ϵ, we accordingly define $\beta(v)$, the maximum possible performance gap discovered (corrected) by probing v, as the maximum real value satisfying

Input: G^0, T, ϵ, b
Output: Seed set S^t at $t = 1, 2, \cdots, T$

1 $\hat{G} \leftarrow G^0$; $\forall v \in V, c_v \leftarrow 0$;
2 **for** $t = 1$ **to** T **do**
3 $\forall v \in V, c_v \leftarrow c_v + 1$;
4 **for** b *times* **do**
5 $S_o \leftarrow k$ nodes with maximum $\hat{d}_{in}(v)$;
6 $\hat{d}_{max} = \max_{u \notin S_o} \hat{d}_{in}(u)$;
7 $\hat{d}_{min} = \min_{w \in S_o} \hat{d}_{in}(w)$;
8 **foreach** $v \in V$ **do**
9 $z_v \leftarrow \sqrt{-2c_v \ln \epsilon}$;
10 **if** $v \in S$ **then**
 $\beta_v \leftarrow \min\left\{0, \hat{d}_{max} - \hat{d}_{in}(v) + z_v\right\}$ **else**
 $\beta_v \leftarrow \min\left\{0, \hat{d}_{in}(v) - z_v - \hat{d}_{min}\right\}$
11 $v^* \leftarrow \arg\max_{v \in V} \beta_v$, $c_{v^*} \leftarrow 0$;
12 Probe v^* in G^t and update $\hat{G}$;
  // Degree discount heuristics
13 $S^t \leftarrow \emptyset$;
14 **for** k *times* **do**
15 $v^* \leftarrow \arg\max_{v \in V \setminus S^t} \hat{h}_{S^t}(v)$;
16 $S^t \leftarrow S^t \cup \{v^*\}$;
17 **foreach** *neighbor* u *of* v^* **do**
18 Update $\hat{h}_{S^t}(u)$;
19 Output S^t;

Algorithm 1: Maximum Gap Probing

$$P\left[\hat{Q}_v\left(S'_o(v)\right) - \hat{Q}_v\left(S_o\right) \geq \beta(v)\right] \leq \epsilon$$

And the objective of our algorithm is to maximize the possible correction by probing v which maximizes $\beta(v)$.

There are several optional instantiations for $\hat{Q}_v(S)$. Since it is intractable to compute the expected influence spread for a given seed set, we instead sum up the in-degree of nodes in seed set S on the observed network $\hat{G}$. Without specifying the in-degree distribution of each node v, the precise value of $\beta(v)$ is still hard to calculate. However, we can employ Azuma-Hoeffding inequality to provide an estimate of $\beta(v)$, as shown in the following part. The pseudo-code of MaxG algorithm is presented in Algorithm 1, where the estimation of $\beta(v)$ is shown in Line 8-10.

Estimation of $\beta(v)$ We derive the estimation of "maximum possible performance gap" $\beta(v)$ below. Note that we omit the superscript t for simplicity.

$\hat{Q}_v(S)$ indicates the sum of in-degree of nodes in S on the observed network $\hat{G}$ after probing v. Thus S_o and S'_o are the k nodes with maximum in-degrees on network $\hat{G}$ before and after probing v respectively. Note that the performance gap $\left(\hat{Q}_v(S'_o(v)) - \hat{Q}_v(S_o)\right)$ is non-zero only when S_o is different from $S'_o(v)$. We consider two situations when the probed node v is in or not in S_o.

If we probe $v \in S_o$, and find that $\hat{d}_{in}(v)$ is still higher than $\max_{u \notin S_o} \hat{d}_{in}(u)$, then the performance gap

will be 0; otherwise v will not appear in $S'_o(v)$. Instead, $u^* = \arg\max_{u \notin S_o} \hat{d}_{in}(u)$ will be included in $S'_o(v)$. Thus, the performance gap before and after probing v would be $\min\left\{0, \hat{d}_{in}(u^*) - d_{in}(v)\right\}$. Similarly, if we probe $v \notin S_o$, the performance gap before and after probing v would be $\min\left\{0, d_{in}(v) - \hat{d}_{in}(w^*)\right\}$, where $w^* = \arg\min_{w \in S_o} \hat{d}_{in}(w)$.

We do not specify how the network evolves over time. However, two conditions below are usually satisfied on most dynamic social networks. First, the difference of in-degree of a node v between two consecutive time stamps is no larger than 1. With sufficiently fine granularity of time, this condition can be satisfied. Second, each node v's in-degree is relatively stable so that $\mathbb{E}\left[d_{in}^{t+1}(v) | d_{in}^{t}(v)\right] = d_{in}^{t}(v)$. Thus the in-degree of each node v is a martingale.

Suppose $(t - c_v)$ is the latest time stamp when v is probed. We apply Azuma-Hoeffding inequality for martingale:

$$P\left(d_{in}^{t}(v) - d_{in}^{t-c_v}(v) \leq -z\right) \leq \exp\left(\frac{-z^2}{2c_v}\right) \qquad (2)$$

Note that according to the definition of c_v, v is probed at time stamp $t - c_v$, and thus we have $\hat{d}_{in}^{t-c_v}(v) = d_{in}^{t-c_v}(v)$. Since $P\left(d_{in}^{t}(v) - \hat{d}_{in}^{t}(v) \leq -z\right) \leq P\left(d_{in}^{t}(v) - \hat{d}_{in}^{t-c_v}(v) \leq -z\right)$. Let the probability in Equation 2 bounded by ϵ. Then with a probability greater than $(1 - \epsilon)$ we have $d_{in}^{t}(v) > \hat{d}_{in}^{t}(v) - z_v$, where

$$z_v = \sqrt{-2c_v \ln \epsilon}$$

Similarly, an upper bound of $d_{in}^{t}(v)$ can be obtained. We again omit the superscript t. The value of $\beta(v)$ at t can then be written as

$$\beta(v) = \begin{cases} \min\left\{0, \hat{d}_{in}(v) - z_v - \min_{w \in S} \hat{d}_{in}(w)\right\}, & v \notin S_o; \\ \min\left\{0, \max_{u \notin S} \hat{d}_{in}(u) - \hat{d}_{in}(v) + z_v\right\}, & v \in S_o. \end{cases}$$

IV. Experimental Results

In this section, we first introduce the data sets we used in our experiments, then describe the experiment setup. Finally we present the experimental results and several analysis.

A. Data Sets

We utilize a synthetic dynamic network and two real dynamic networks to verify the effect of our proposed algorithms. See Table I for statistics of the data sets.

Synthetic. We generate the synthetic dynamic networks similar to a dynamic graph model borrowing the idea of preferential attachment [2]. We first establish a random network as the network G^0 at time stamp $t = 0$. It is generated according to Erdös-Rényi model with the number of nodes as 500 and the probability to construct an edge for each pair of nodes as 0.05. Then for each of the following time stamp, we uniformly choose 100 edges and change their heads respectively to 100 nodes stochastically chosen with probability proportional to their in-degree. We generate 200 time stamps for a synthetic dynamic network.

Table I: Details of data sets

Data set	#(Nodes that have ever appeared)	#(Edges that have ever appeared)[4]	Time stamps
Synthetic	500	12,475	200
Twitter	18,089,810	21,097,569	10
Coauthor	1,629,217	2,623,832	27

Twitter. We crawled the follow links between 18,089,810 users from Twitter[2] at 10 different time stamps during October to December 2010. The average time interval between two consecutive time stamps is about one week. There are 21,097,569 directed edges that once appeared in any of all the 10 network snapshots. We treat the network at the first time stamp as known, and regard the following 9 as unknown snapshots of the dynamic network.

Coauthor. We construct a dynamic coauthor network from ArnetMiner[3]. We collected 1,768,776 publications published during 1986 to 2011 by 1,629,217 authors. We regard each year as a time stamp and there are 27 time stamps in total. At each time stamp, we create an edge between two authors if they have coauthored at least one paper in the most recent 3 years (current year included). We convert the undirected coauthor network into directed by regarding each undirected edge as two symmetric directed edges.

B. Experiment Setup

Comparison methods. We conduct a comparison between the following algorithms.

- *Random Probing (Rand).* Randomly choose b nodes to probe with uniform probability at each time stamp.
- *Enumerating Probing (Enum).* Probe each node one by one in a circular order with equal frequency.
- *Degree Weighted Probing (Deg).* Randomly chooses b nodes independently with probability proportional to their observed degree $\hat{d}^t(v)$ at time stamp t.
- *Degree Weighted Round-Robin Probing (DegRR).* Deterministic version of Deg strategy.
- *Maximum Gap Probing (MaxG).* The algorithm proposed in Section III.

Besides, we also perform influence maximization on fully observed network G^t to provide an estimation of the upper bound of the performance achieved by each probing algorithm. We denote it as "BEST".

Evaluation measures. To evaluate the performance of probing algorithms, we employ the seed set S^t output by

[2]https://twitter.com

[3]http://arnetminer.org

[4]Since we have multiple synthetic networks in the experiments, we give the expected number of edge in a synthetic dynamic network instead.

Table II: Average expected number of activated nodes

Data Set	b	Rand	Enum	Deg	DegRR	MaxG	BEST
Synthetic	1	13.83	13.55	13.78	14.30	**14.79**	15.95
	5	15.07	15.33	15.09	15.40	**15.60**	
Twitter	100	987.74	987.62	988.41	1001.47	**1005.12**	1011.15
	500	987.45	987.67	988.36	1006.38	**1010.61**	
Coauthor	100	20.34	20.82	28.67	38.94	**45.51**	91.51
	500	20.35	22.93	44.27	56.68	**61.74**	

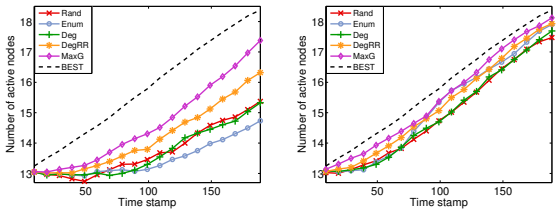
the influence maximization algorithm obtained from the observed network $\hat{G}^t$ after probing, and calculate the expected number of nodes activated by the seed set on the real network G^t. We then measure the algorithm's performance by the average expected number of activated nodes over all the time stamps. In our experiments, the expected number of activated nodes at each time stamp is approximated by 2000 rounds of simulations. For Synthetic data set, we repeat all the experiments on 5 different synthetic dynamic networks and take the average value to obtain a more accurate result.

Experiment setup. We use Independent Cascading Model [9] with uniform probabilities $p = 0.01$ in our experiments. In Synthetic data set, we set the size of seed set $k = 30$. In Twitter and Coauthor data set we set $k = 100$. At each time stamp, we allow the algorithm to probe b nodes. For Synthetic data set, we set $b = 1$ and $b = 5$ respectively, while in the other two data sets we set $b = 100$ and $b = 500$. For MaxG algorithm, we set the tolerance probability $\epsilon = 0.01$. We will show in our following experiments that the performance of MaxG algorithm remains unsensitive to ϵ in a very wide range.

C. Performance and Analysis

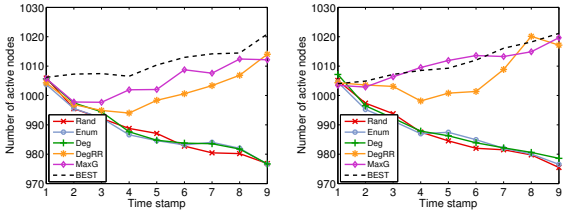
Performance Comparison. We test Rand, Enum, Deg, DegRR, and MaxG algorithms on all the three data sets. Table II illustrates the average expected number of activated nodes of different algorithms on all the data sets.

As shown in Figure 2(a), on Synthetic data set, our proposed algorithm MaxG significantly outperforms baselines algorithms (z-test, $\alpha = 0.05$) when the probing budget b is limited to be 1. The error rate of Rand algorithm is 13%, while our MaxG algorithm can achieve error rate of 7%. We also verify the effect of our algorithms on two large real data sets, Twitter and Coauthor (Cf. Figure 3 and 4). It is clear that our proposed algorithm MaxG still significantly outperforms other baseline algorithms such as Rand (sign-test, $\alpha = 0.05$). In Twitter data set, the error rate achieved by MaxG algorithm is only $\frac{1}{50}$ with respect to the error rate achieved by Rand. In Coauthor data set, our MaxG algorithm improves the expected influence spread by 203% compared to Rand baseline with $b = 500$. Note that on Figure 3(b), on some time stamps the performance of MaxG and DegRR algorithms is even better than "BEST". It is because the degree discount heuristics we employ for influence maximization on real network provides an estimation of the optimal solution.



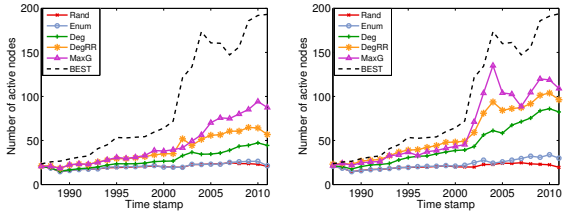
(a) $b = 1$ (b) $b = 5$

Figure 2: Results of Synthetic data set



(a) $b = 100$ (b) $b = 500$

Figure 3: Results of Twitter data set



(a) $b = 100$ (b) $b = 500$

Figure 4: Results of Coauthor data set

Accuracy of observed networks. We also conduct a study on how similar the observed network is to the real network. More precisely, we use the Jaccard similarity between the edge sets of the observed network and the real network as the metric to measure the accuracy of the observed network. We plot the Jaccard similarity on Synthetic data set ($b = 5$) in Figure 5. It can be observed that DegRR and MaxG are capable of constructing an observed network which is more close to the real network than Rand and Deg. This result to some extent explains why they can achieve better performance. However, Enum also achieves similar accuracy, which implies that our task is more challenging than simply recovering the real network.

In-depth analysis of MaxG algorithm. We conduct an experiment to explore the correlation between the performance of MaxG and the tolerance probability ϵ. We set $\epsilon = e^{-0.5}, e^{-1}, \ldots, e^{-8}$ respectively and perform MaxG algorithm on the synthetic data set, with $b = 1$. We plot the curve of performance on Figure 6. It can be observed

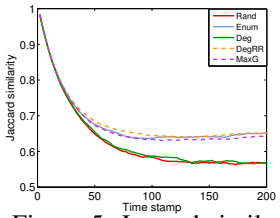


Figure 5: Jaccard similarity between observed and real networks. Tested on synthetic data set ($b = 1$).

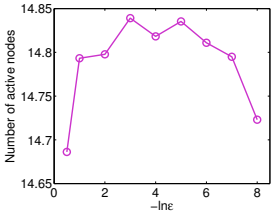


Figure 6: Tolerance probability ϵ vs. MaxG performance. Tested on synthetic data set ($b = 1$).

that when ϵ is between $[e^{-7}, e^{-1}]$, the performance of MaxG algorithm is very stable. When the tolerance probability is set to a value smaller than e^{-8} or larger than $e^{-0.5}$, the performance drops slightly, but is still better than other baselines. Thus we can choose an ϵ between $[e^{-7}, e^{-1}]$ to guarantee the performance of MaxG algorithm.

V. Related work

There has been a number of pieces of research on social influence analysis. Tang et al. [16] investigated how to measure the topic-level social influence and Tang et al. [17] studied the problem of conformity influence. [10] studied how to learn the influence probability parameters in social networks. Myers et al. [12] modeled external influence into the information diffusion process. However, few of them addressed the problem that social networks were actually varying. Prakash et al. [13] extended the Susceptible-Infected-Susceptible (SIS) model to dynamic networks and derived the epidemic threshold. But they did not provide an algorithm to maximize the influence.

There was extensive research on influence maximization. [9], [4] studied algorithms for influence maximization problem. Few of them focused on dynamic networks. [15], [6] studied several variations to optimize submodular functions, which are related to influence maximization on dynamic networks, but they did not consider to probe the real network.

Dynamic graph analysis is also another related work. Leskovec et al. [11] studied the microscopic evolution of social networks. Bahmani et al. [1] studied the computation of PageRank on evolving graphs.

VI. Conclusion

In this paper, we study the novel problem of influence maximization in dynamic social networks. We formally define the problem as probing nodes in an unobserved network and propose the Maximum Gap Probing (MaxG) algorithm. We use one synthetic network and two real networks to validate the effectiveness of the proposed method. This research has several interesting implications. First, the proposed probing algorithm can be directly used to guide online marketing decisions in the social networks. Second, the proposed probing algorithm is general and can be applied to other applications on streaming graphs, e.g., calculation of the PageRank scores on evolving graphs.

Acknowledgements. Honglei Zhuang is supported by U.S. National Science Foundation grants CNS-0931975, IIS-1017362, IIS-1320617 and the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and W911NF-11-2-0086 (Cyber-Security). Jie Tang and Yihan Sun are supported by the Natural Science Foundation of China (No. 61222212, No. 61073073), a special fund for Fast Sharing of Science Paper in Net Era by CSTD, and a research fund supported by Huawei Inc. Jialin Zhang and Xiaoming Sun are supported in part by the Natural Science Foundation of China Grant (No. 61170062, 61222202).

References

[1] B. Bahmani, R. Kumar, M. Mahdian, and E. Upfal. Pagerank on an evolving graph. In *KDD*, pages 24–32, 2012.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI*, 2012.

[4] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, pages 199–207, 2009.

[5] N. E. Friedkin. *A Structural Theory of Social Influence*. Cambridge University Press, 1998.

[6] D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, pages 333–345, 2010.

[7] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. *Proc. VLDB Endow.*, 5(1):73–84, 2012.

[8] A. Goyal, F. Bonchi, L. V. S. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 2(1), 2012.

[9] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[10] M. Kimura, K. Saito, K. Ohara, and H. Motoda. Learning information diffusion model in a social network for predicting influence of nodes. *Intell. Data Anal.*, 15(4):633–652, 2011.

[11] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD*, pages 462–470, 2008.

[12] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, pages 33–41, 2012.

[13] B. A. Prakash, H. Tong, N. Valler, M. Faloutsos, and C. Faloutsos. Virus propagation on time-varying networks: Theory and immunization algorithms. In *ECML/PKDD*, pages 99–114, 2010.

[14] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[15] M. J. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, pages 1577–1584, 2008.

[16] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816, 2009.

[17] J. Tang, S. Wu, and J. Sun. Confluence: Conformity influence in large social networks. In *KDD*, pages 347–355, 2013.