Towards Efficient Influence Maximization for Evolving Social Networks

Xiaodong Liu(✉), Xiangke Liao, Shanshan Li, and Bin Lin

National University of Defense Technology, Changsha 410073, China
{liuxiaodong,xkliao,shanshanli,binlin}@nudt.edu.cn

Abstract. Identifying the most influential individuals can provide invaluable help in developing and deploying effective viral marketing strategies. Previous studies mainly focus on designing efficient algorithms or heuristics to find top-K influential nodes on a given static social network. While, as a matter of fact, real-world social networks keep evolving over time and a recalculation upon the changed network inevitably leads to a long running time, significantly affecting the efficiency. In this paper, we observe from real-world traces that the evolution of social network follows the preferential attachment rule and the influential nodes are mainly selected from high-degree nodes. Such observations shed light on the design of IncInf, an incremental approach that can efficiently locate the top-K influential individuals in evolving social networks based on previous information instead of calculation from scratch. In particular, IncInf quantitatively analyzes the influence spread changes of nodes by localizing the impact of topology evolution to only local regions, and a pruning strategy is further proposed to effectively narrow the search space into nodes experiencing major increases or with high degrees. We carried out extensive experiments on real-world dynamic social networks including Facebook, NetHEPT, and Flickr. Experimental results demonstrate that, compared with the state-of-the-art static heuristic, IncInf achieves as much as 21× speedup in execution time while maintaining matching performance in terms of influence spread.

1 Introduction

Influence maximization (IM) is one fundamental and important problem which aims to identify a small set of influential individuals so as to develop effective viral marketing strategies in large-scale social networks [7]. As a matter of fact, real-world social networks keep evolving over time. For example, in Facebook, new people might join while old ones might withdraw, and people might make new friends with each other. Moreover, real-world social networks are evolving in a rather surprising speed; it is reported that as much as 1 million new accounts are created in Twitter every day. Such massive evolution of network topology, on the contrary, may lead to a significant transformation of the network structure, thus raising a natural need of efficient reidentification.

Existing researches on influence maximization focus mainly on developing effective and efficient algorithms on a given static social network. Although one

F. Li et al. (Eds.): APWeb 2016, Part I, LNCS 9931, pp. 232–244, 2016.
DOI: 10.1007/978-3-319-45814-4_19

could possibly run any of the static influence maximization methods, such as [5,6,12], to find the new top-K influential individuals when the network is updated, this approach has some inherent drawbacks that cannot be neglected: (1) the running time of a specific static method can be extremely long and unacceptable especially on large-scale social networks, and (2) whenever the network topology is changed, we need to recalculate the influence spreads for all the nodes which leads to very high costs. Can we quickly and efficiently identify the influential nodes in evolving social networks? Can we incrementally update the influential nodes based on previously known information instead of frequently recalculating from scratch?

Unfortunately, the rapidly and unpredictably changing topology of a dynamic social network poses several challenges in the reidentification of influential users. On one hand, the interconnections between edges in real-world social graphs are rather complicated; as a result, even one small change in topology may affect the influence spreads of a large number of nodes, not to mention the massive changes in large-scale social networks. It is very difficult to efficiently compute the changes of influence spreads for all the nodes after the evolution. On the other hand, since there are a great many nodes in large-scale social networks, how to effectively limit the range of potential influential nodes and reduce the amount of calculation as much as possible is a very challenging problem.

To well address these challenges, we investigate the dynamic characteristics exhibited during the evolution of real-world social networks. Through tests on three real-world dataset traces, Facebook [15], NetHEPT [2] and Flickr [14], we observe that, first, the growth of social network is mainly based on the preferential attachment principle [3], that is the new-coming edges prefer to attach to nodes with higher degree, which naturally leads to the "rich-get-richer" phenomena; and second, the top-K influential nodes are mainly selected from those high-degree nodes. Inspired by such observations, we know that the influence changes of some nodes will have no impact on the top-K selection, and thus can be pruned to reduce the amount of calculation. Motivated by this, we propose IncInf, an incremental method to identify the top-K influential nodes in evolving social networks instead of recalculating from scratch, thus significantly improves the efficiency and scalability to handle extraordinarily large-scale networks. To summarize, the main contributions of IncInf are as follows:

First, we design an efficient approach to quantitatively analyze the influence spread changes from topology evolution by adopting the idea of localization. A tunable parameter is provided to tradeoff between efficiency and effectiveness.

Second, we propose a pruning strategy which could effectively narrow the search space into nodes only experiencing major increases or with high degrees based on the changes of influence spread and the previous top-K information.

Third, we conduct extensive experiments on three real-world social networks. Compared with the state-of-the-art algorithm, IncInf achieves up to 21× speedup in execution time while providing matching influence spread. Moreover, IncInf provides better scalability to scale up to extraordinarily large-scale networks.

2 Preliminaries and Problem Statement

Social Network. A social network is formally defined as a directed graph $G = (V, E, P)$ where node set $V = \{v_1, v_2, \cdots, v_n\}$ denotes entities in the social network. Each node can be either active or inactive, and will switch from being inactive to being active if it is influenced by others nodes. Edge set $E \subset V \times V$ is a set of directed edges representing the relationship between different users. Take Twitter as an example. A directed edge (v_i, v_j) will be established from node v_i to v_j if v_i is followed by v_j, which indicates that v_j may be influenced by v_i. P denotes the influence probability of edges; each edge $(v_i, v_j) \in E$ is associated with an influence probability $p(v_i, v_j)$ defined by function $p : E \rightarrow [0, 1]$.

Independent Cascade (IC) Model. IC model is a diffusion model that has been well studied in [5,11,12]. Given an initial set S, the diffusion process of IC model works as follows. At step 0, only nodes in S are active, while other nodes stay in the inactive state. At step t, for each node v_i which has just switched from being inactive to being active, it has a single chance to activate each currently inactive neighbor v_j, and succeeds with a probability $p(v_i, v_j)$. If v_i succeeds, v_j will become active at step $t+1$. If v_j has multiple newly activated neighbors, their attempts in activating v_j are sequenced in an arbitrary order. Such a process runs until no more activations are possible [11]. We use $\sigma(S)$ to denote the influence spread of set S, which is defined as the expected number of active nodes at the end of influence propagation.

IM problem in Evolving Networks. An evolving network $\zeta = (G^0, G^1, \cdots, G^t)$ is defined as a sequence of network snapshots evolving over time, where $G^t = (V^t, E^t, P^t)$ is the network snapshot at time t. $\Delta G^t = (\Delta V^t, \Delta E^t, \Delta P^t)$ denotes the structural change of network G^t from time t to $t + 1$. Obviously, we have $G^{t+1} = G^t \bigcup \Delta G^t$. The influence maximization problem is defined as follows:

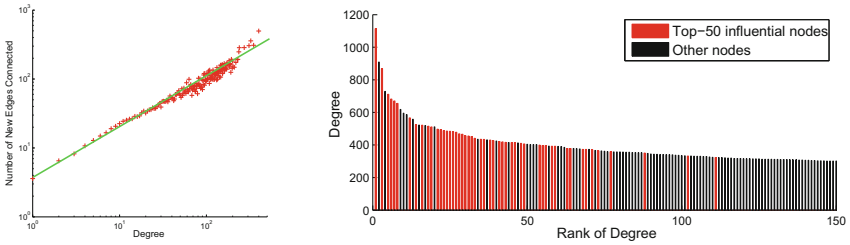
Given: Social network G^t at time t, the top-K influential nodes S^t in G^t, and the structural evolution ΔG^t of graph G^t.

Objective: Identify the influential nodes $S^{t+1} \subset V^{t+1}$ of size K in G^{t+1} at time $t + 1$, such that $\sigma(S^{t+1})$ is maximized at the end of influence diffusion.

3 Observations of Social Network Evolution

3.1 Preferential Attachment Rule

Understanding the pattern of the network topology evolution is of primary importance to design efficient influence maximization algorithms for evolving social networks. We first study the preferential attachment rule [3], or in other words, the "rich-get-richer" rule [8], which postulates that when a new node joins the network, it creates a number of edges, where the destination node of each edge is chosen proportional to the destination's degree. This means that new edges are more likely to connect to nodes with high degree than ones with



(a) Preferential attachment (b) Relation between influence and degree

Fig. 1. Network topology evolution pattern on Facebook.

low degree. This is reasonable in reality; Lady Gaga gains 30,000 new followers on average every day which can never image for any common individual. The results on the Facebook dataset are demonstrated in Fig. 1(a) where the x axis is the degree of different nodes and the y axis is the average number of new edges attached to nodes of different degree. Note that both the x and y axis are in log scale. From Fig. 1(a) we can see that the degree of users in Facebook is linearly correlated with the number of new links created. This suggests that high-degree nodes get super-preferential treatment. Consequently, the influence spread change should be considerably great for the influential nodes, while there may be only small or even no change for ordinary people.

3.2 Relation Between Influence and Degree

Examining the relation between the influence and the degree of node can help us understand the effect of degree changing on the influence spread of nodes. For this reason, we run the static MixGreedy algorithm [5] on the final graph and identify the top-50 influential nodes. The results on the Facebook dataset are illustrated in Fig. 1(b) where the x axis is the rank of degrees of different nodes (we only show the top 150). Obviously, all the selected influential nodes have a large degree. In particular, among the 50 nodes, 48 nodes rank in top 100 of the whole 61,096 nodes in terms of degree, and the other two nodes rank 102 and 111 respectively. While on the NetHEPT and Flickr datasets, the top-50 influential nodes are selected from the top 1.79 % and 0.84 % nodes in degree, respectively. This demonstrates that the top-K influential nodes are mainly selected from those with large degrees.

4 IncInf Design

In this section, we present the detailed design of IncInf, an incremental approach to solve the influence maximization problem on dynamic social networks. The main idea of IncInf is to take full use of the valuable information that is inherent in the network structural evolution and previous influential nodes, so as to substantially narrow the search space of influential nodes. In this way IncInf is able

to incrementally identify the top-K influential nodes S^{t+1} of G^{t+1} at time $t+1$ based on the previous influential nodes S^t at time t and the structural change ΔG^t from G^t to G^{t+1}, thus significantly reduces the computation complexity and improves the efficiency.

4.1 Basic Operations of Topology Evolution

The evolution of social network, when reflected into its underlying graph, can be summarized into six categories, which are inserting or removing a node, introducing or deleting an edge, and increasing or decreasing the influence probability of an edge. We denote the six types of topology change as $addNode$, $removeNode$, $addEdge$, $removeEdge$, $addWeight$, $decWeight$. The detailed descriptions and their effects on influence spread are shown in Table 1.

Table 1. Details of six types of basic operation

Operation	Description	Impact on influence spread
$addNode(u)$	add a new node u into the current network	the influence spread of u is set to 1
$removeNode(u)$	delete an existing node u from the network	the influence spread of u is set to 0
$addEdge(u, v, w)$	introduce a new edge (u, v) with $p(u, v) = w$	the influence spread of all the nodes that can reach u may be increased
$removeEdge(u, v)$	remove an existing edge (u, v) from the network	the influence spread of all the nodes that can reach u may be decreased
$addWeight(u, v, \Delta w)$	increase $p(u, v)$ by Δw	the influence spread of all the nodes that can reach u may be increased
$decWeight(u, v, \Delta w)$	reduce $p(u, v)$ by Δw	the influence spread of all the nodes that can reach u may be decreased

It should be noted that only after the $addNode$ operation can node u establish links ($addEdge$) or sever links ($removeEdge$) with other nodes, and node u can only be removed when all its associated edges are deleted. Moreover, the weight operation can be equivalently decomposed into two edge operations. For example, $addWeight(u, v, \Delta w)$ can be divided into $removeEdge(u, v)$ and $addEdge(u, v, w + \Delta w)$, supposing the previous weight of edge (u, v) is w.

4.2 Influence Spread Changes

As discussed above, whenever an edge (u, v) is introduced into or removed from the social network, the influence spread of all the nodes that can reach node u

may be changed. However, as a matter of fact, the real-world social networks exhibit small-world network characteristics and the connections between nodes are highly complicated. As a result, even one small change in topology, such as an edge addition or removal, may affect the influence spread of a large number of nodes, thus introducing massive recalculations. In order to reduce the amount of computation, we design an approach to efficiently calculate the changes on the influence spread of nodes which adopts the localization idea [6].

The main idea of localization is to use the local region of each node to approximate its overall influence spread. In particular, we use the maximum influence path to approximate the influence spread from node u to v. Here the maximum influence path $MIP(u, v, G)$ from node u to v in graph G is defined as the path with the maximum influence probability among all the paths from node u to v, and can be formally described as follows:

$$MIP(u, v, G) = \arg \max_{p \in P(u,v,G)} \{prob(p)\} \tag{1}$$

where $prob(p)$ denotes the propagation probability of path p and $P(u, v, G)$ denotes all the paths from node u to v in graph G. For a given path $p = \{u_1, u_2, \cdots, u_m\}$, the propagation probability of path p is defined as follows:

$$prob(p) = \prod_{i=1}^{m-1} p(u_i, u_{i+1}) \tag{2}$$

Moreover, an influence threshold θ is set to tradeoff between accuracy and efficiency. During the propagation process, we only consider paths whose influence probability are larger than θ while ignoring those with smaller probability. By doing this, the influence is effectively restricted to the local region of each node.

Similarly, in our proposal we localize the impact of topology changes on influence spread into local regions, and thus reduce the amount of computation. Among six types of topology change, *addNode* (or *removeNode*) is the most straightforward since it simply sets the influence spread of the node to 1 (or 0); *addWeight*, *decWeight* as well as *removeEdge* are methodologically similar to *addEdge*. Consequently, in the following we take *addEdge* as an example to show which nodes' influence spread need to be updated and how to determine those changes when a new edge is added into the graph.

Consider the case when a new edge $e = (u, v, w)$ is introduced between two existing node u and v. We denote the graph before and after such a topology change as G^t and $G^{t'}$, and the current seed set is S. The detailed algorithm is described in Algorithm 1. According to the principle of localization [6], if the propagation probability w is smaller than the specified threshold θ, or not bigger than the probability of $MIP(u, v, G^t)$, edge e can be simply neglected and there is no need to update any node's influence spread (lines 1–3). Otherwise, the newly-added edge e would become the $MIP(u, v, G^{t'})$. As a result, each node i whose maximum influence path to u has an influence probability larger than θ is likely to experience a rise in terms of influence spread (line 4) because node i may influence more nodes through the new edge e. So, we then check the probability

Algorithm 1. Edge addition

Input: a new edge $e = (u, v, w)$, graph G^t.
Output: The influence spread changes of nodes in $G^{t'}$.

1: **if** $w < \theta$ or $w \leq prob(MIP(u, v, G^t))$ **then**
2: return;
3: **end if**
4: **for** each node i with $prob(MIP(i, u, G^t)) > \theta$ **do**
5: **for** each node j with $prob(MIP(v, j, G^t)) > \theta$ **do**
6: **if** $prob(MIP(i, j, G^t)) < \theta$ and $prob(MIP(i, j, G^{t'})) > \theta$ **then**
7: $deltaInf[i] += prob(MIP(i, j, G^{t'})) \times (1 - prob(j, S))$
8: **end if**
9: **if** $prob(MIP(i, j, G^t)) > \theta$ and $prob(MIP(i, j, G^{t'})) > \theta$ **then**
10: $deltaInf[i] += (prob(MIP(i, j, G^{t'})) - prob(MIP(i, j, G^t))) \times (1 - prob(j, S))$
11: **end if**
12: **end for**
13: **end for**

of the maximum influence path from i to v and its successors in G^t and $G^{t'}$. Based on the two probabilities, we divide the problem into two small cases:

The first case is when the probability of maximum influence path from i to j in G^t is smaller than θ while that in $G^{t'}$ is larger than θ (lines 5–6). Here j denotes the node whose probability of $MIP(v, j, G^t)$ is larger than θ. In such a case, node i build a new path to j through the new edge e which increases the influence spread of i by $prob(MIP(i, j, G^{t'})) \times (1 - prob(j, S))$ (line 7). Here $prob(j, S)$ is the probability of that node j is influenced by the current seed set S, which is defined as follows:

$$prob(j, S) = \begin{cases} 1, & \text{if } j \in S \\ 1 - \prod_{w \in n(j)} 1 - prob(w, S) \cdot p(w, j), & \text{if } j \notin S \end{cases}$$

Here $n(j)$ denotes the in-neighbour set of j.

The second case is when the probability of maximum influence path from i to j is larger than θ in both G^t and $G^{t'}$ (lines 9–11). In this case, the influence increase of node i is $(prob(MIP(i, j, G^{t'})) - prob(MIP(i, j, G^t))) \times (1 - prob(j, S))$.

We treat the network dynamics from G^t to G^{t+1} as a finite change stream $c_1, c_2, \cdots, c_i, \cdots$ where each change c_i is one of the six topology changes we described above. When all the changes in the change stream are processed, we can obtain the influence spread change for all the nodes.

4.3 Potential Top-K Influential Users Identification

From the preferential attachment rule, we know that the influence spread changes of those high-degree nodes should be much greater than the ordinary nodes. Moreover, according to the power-law distribution, such high-degree nodes only account for a small part of the whole nodes. Consequently we can pick out nodes only experiencing major increases or with high degrees because these nodes are of great potential to become the top-K influential nodes in G^{t+1}. Then we only calculate the actual influence spread for these selected nodes while ignoring the others. In this way, a large percent of nodes are pruned and the search space is largely narrowed. It should be noted that a smart pruning strategy is of key importance since a poor selection might either affect the efficiency or reduce the accuracy in terms of influence spread. We describe the details of our pruning strategy as follows:

(1) In the ith iteration, if the influence spread of the previous influential node S_i^t increases in G^{t+1}, the chosen nodes are those with a larger influence spread change than $deltaInf[S_i^t]$;

In most cases, the influential nodes will attract a good many of new nodes and establish new links. Thus, their influence spreads will increase drastically. In such a case, the nodes whose influence spread changes are smaller than the influential nodes are completely impossible to become the most influential node in G^{t+1}. Therefore, when the influence spread of the previous influential nodes increase, we only select those whose influence spread changes are larger than the influential nodes in G^t. According to the preferential attachment rule, such a pruning method can greatly narrow the search space and reduce the amount of computation.

(2) In the ith iteration, if the influence spread of the previous influential node S_i^t decreases in G^{t+1}, in addition to item (1), the nodes are further selected to hold a sufficiently large degree or experience a sufficiently great increase. In order to formally define "large degree" and "great increase", here we set an threshold η to tradeoff between running time and influence spread. Here the nodes with sufficiently large degrees (or great increase) are defined as the set of node v_j whose degree (or degree increase ratio) is among the top η percent of all nodes in G^{t+1}. The degree increase ration of v_j is defined as $degree_j^{t+1}/degree_j^t$ where $degree_j^t$ denotes the degree of node v_j in graph G^t.

It should be noted that although the case the influence spread of a previous influential node decreases during the evolution rarely happens, we consider it here for completeness. In this case, the amount of nodes satisfying item (1) is relatively large which leads to mass computation and need to be further pruned. In order to select only the most potential nodes, we additionally select the nodes with large degree or large increase because a node with small degree has only very low probability to become an influential node in reality. Consequently, the search space is strictly circumscribed and the computational complexity is greatly reduced.

After the potential nodes are selected, we calculate the actual influence spread of these nodes in G^{t+1} and select the one with the maximum influence spread in

Algorithm 2. IncInf

Input: G^t, S^t, and G^{t+1}.
Output: the top-K influential nodes S^{t+1} in G^{t+1}.

```
1: Initialize S^{t+1} = ∅;
2: for i = 1 to K do
3:    for each topology change c_j from G^t to G^{t+1} do
4:       calculate the influence spread change deltaInf[·];
5:    end for
6:    select a set of potential nodes as pn according to pruning strategy;
7:    for each node v_l ∈ pn do
8:       calculate the marginal influence spread σ_{S^{t+1}}(v_j);
9:    end for
10:   select v_max = arg max_{v_j ∈ pn} (σ_{S^{t+1}}(v_j));
11:   S^{t+1} = S^{t+1} ∪ v_max;
12: end for
```

each iteration. Algorithm 2 outlines the design of our proposed algorithm IncInf. IncInf iterates for K round (line 2) and in each round select one node providing the maximum marginal influence spread. Lines 3–5 calculate the influence spread change of each node caused by the topology evolution. Nodes with great potential to become top-K influential are selected (line 6) and their influence spread are computed in G^{t+1} (lines 7–9). Then the node providing the maximal marginal gain will be selected and added to the S^{t+1} (lines 10–11).

5 Experiments

5.1 Experimental Setup

We use three real-world social networks Facebook [15], NetHEPT [2], and Flickr [14], and each dataset includes multiple snapshots of different time stamps. Table 2 summarizes the statistical information of these datasets. We compare our algorithm with four static algorithms: **MixGreedy**, **ESMCE**, **MIA** and **Random**. MixGreedy is an improved greedy algorithm proposed by Chen et al. in [5]. ESMCE is a power-law exponent supervised estimation approach proposed in [12]. MIA is a heuristic that uses local arborescence structures of each

Table 2. Summary information of the real-world social networks

Datasets	Nodes			Edges		
	Initial Number	Final Number	Growth	Initial Number	Final Number	Growth
Facebook	12,364	61,096	394 %	73,912	905,665	1125 %
NetHEPT	5,802	29,555	409 %	57,765	352,807	511 %
Flickr	1,620,392	2,570,535	58.6 %	17,034,807	33,140,018	94.5 %

node to approximate the influence propagation [6]. Random is a basic heuristic that randomly selects K nodes from the whole datasets.

The propagation probability of the IC model is selected randomly from 0.1, 0.01, and 0.001 for each network snapshot, and we run simulations 10000 times and take the average influence spread. The pruning threshold η is set to 5 % and we aim to find the top 50 influential nodes from each dataset. The experiments are conducted on a PC with Intel Core i7 920 CPU @2.67 GHz and 6 GB RAM.

5.2 Efficiency Study

The time costs of different algorithms are illustrated in Fig. 2 where we record the total time cost for each snapshot of the three datasets. The experimental results show that the time costs of our algorithm on each snapshot are obviously less than those of static algorithms. Obviously, MixGreedy takes the longest time among four kinds of influence maximization algorithms. It takes MixGreedy more than as much as 6 h to identify the top 50 influential nodes on the final NetHEPT dataset, while the time is even longer on the larger dataset Facebook. Moreover, MixGreedy is not feasible to run on the largest dataset Flickr due to the unbearably long running time. ESMCE, benefiting from its sampling estimation method, runs much faster than MixGreedy, but it still takes as much as 3511 s on average to run on the five snapshots of Flickr. Compared with two greedy algorithms, the heuristic MIA performs much better. It only takes MIA 23.8 s to run on the final Facebook graph. When running on the Flickr dataset with as much as 2.5 M nodes and 33 M edges, however, its speedup is far from satisfactory, since it still needs more than 45 min to finish. While our proposed algorithm, IncInf, outperforms all the static algorithms in terms of efficiency. In particular, IncInf is almost four orders of magnitude faster than the MixGreedy algorithm on the Facebook dataset. While compared with the MIA heuristic, the speedup of IncInf is 8.41× and 6.94× on the Facebook and NetHEPT datasets, respectively; What's more, when applied on the largest dataset Flickr, IncInf can achieve as much as 20.65× speedup on average. This is because IncInf only

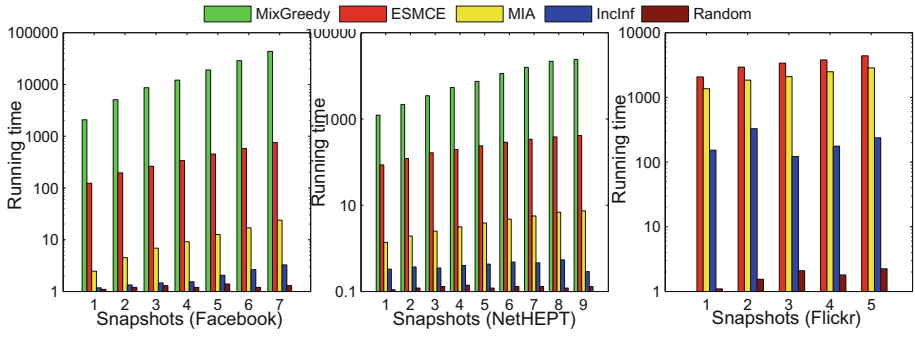


Fig. 2. The time costs of different algorithms on three real-world datasets.

computes the incremental influence spread changes and adaptively identifies the influential nodes based on the previous influential nodes and the current influence spread changes. The experimental results clearly validate the efficiency advantage of our incremental algorithm IncInf. Without doubt, Random runs the fast among all the algorithms. However, as we will show in Sect. 5.3, its accuracy is much worse and unacceptable when developing real-world viral marketing strategies.

5.3 Effectiveness Study

Figure 3 shows the experimental results. MixGreedy outperforms all the other algorithms in terms of influence spread. However, the efficiency issue limits its application to large-scale dataset such as Flickr. The performance of ESMCE, MIA and IncInf almost match MixGreedy on the Facebook dataset, while on NetHEPT, the gaps become larger but remain acceptable (only 3.4 %, 4.7 % and 5.1 % lower than MixGreedy on average). When applied to the Flickr dataset, ESMCE performs the best since ESMCE strictly control the error threshold by iterative sampling. Compared with MIA, IncInf shows very close performance and is only 2.87 % lower on average of all five snapshots, which demonstrates the effectiveness of our proposal. Random, as the baseline heuristic, clearly performs the worst on all the graphs. The influence spread of Random is only 15.6 %, 12.1 % and 10.9 % of that of IncInf on Facebook, NetHEPT and Flickr, respectively.

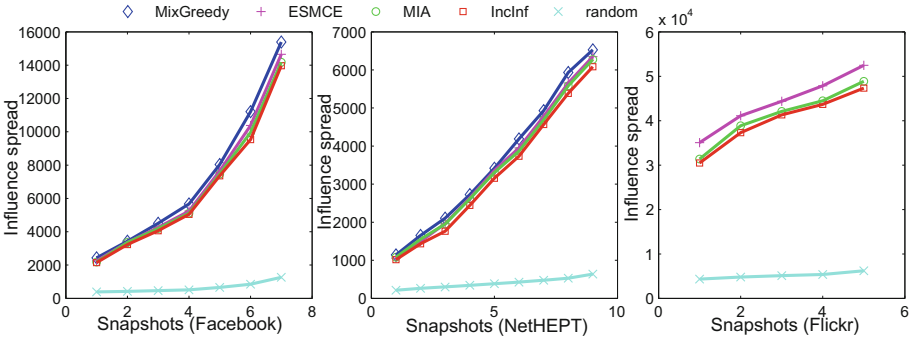


Fig. 3. The influence spread of different algorithms on three datasets.

6 Related Work

Although Influence maximization on static networks has attracted a lot of attentions [5,6,13], studies on dynamic social networks still remains largely unexplored to date. Habiba et al. [10] propose a dynamic social network model which

is different from ours. In their proposal, the network keeps evolving during the process of influence propagation, and their goal is to find the top-K influential nodes over such a dynamic network. Chen et al. [4] extend the IC model to incorporate the time delay aspect of influence diffusion among individuals in social networks, and consider time-critical influence maximization, in which one wants to maximize influence spread within a given deadline. While in [9], the authors consider a continuous time formulation of the influence maximization problem in which information or influence can spread at different rates across different edges. Charu Aggarwal et al. [1] try to discover influential nodes in dynamic social networks and they design a stochastic approach to determine the information flow authorities with the use of a globally forward approach and a locally backward approach.

7 Conclusion

In this paper, we consider the influence maximization problem in evolving social networks, and propose an incremental algorithm, IncInf, to efficiently identify top-K influential nodes. Taking advantage of the structural evolution of networks and previous information on individual nodes, IncInf substantially reduces the search space and adaptively selects influential nodes in an incremental way. Extensive experiments demonstrate that IncInf significantly reduces the execution time of state-of-the-art static influence maximization algorithm while maintaining satisfying accuracy in terms of influence spread.

Acknowledgments. This research was supported by NSFC under grant NO. 61402511. The authors would like to thank the anonymous reviewers for their helpful comments.

References

1. Aggarwal, C., Lin, S., Yu, P.: On influential node discovery in dynamic social networks. In: SDM, pp. 636–647. SIAM, California, USA (2012)
2. ArXiv NetHEPT dataset (2003). http://www.cs.cornell.edu/projects/kddcup/datasets.html. Accessed 18 June 2013
3. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
4. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. In: AAAI, Toronto, Canada (2012)
5. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: SIGKDD, pp. 199–208. ACM, Paris, France (2009)
6. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: SIGKDD, pp. 1029–1038. USA (2010)
7. Domingos, P., Richardson, M.: Mining the network value of customers. In: SIGKDD, pp. 57–66. ACM, San Francisco, CA, USA (2001)
8. Easley, D., Kleinberg, J.: Power laws and rich-get-richer phenomena. In: Networks, Crowds, and Markets: Reasoning about a Highly Connected World (2010)

9. Gomez-Rodriguez, M., Scholkopf, B.: Influence maximization in continuous time diffusion networks. In: ICML. IEEE, Edinburgh (2012)
10. Habiba, T.B.W., Berger-Wolf, T.Y.: Maximizing the extent of spread in a dynamic network. DIMACS TR: 2007–20 (2007)
11. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: SIGKDD, pp. 137–146. ACM, Washington, D.C., USA (2003)
12. Liu, X., Li, S., Liao, X., Wang, L., Wu, Q.: In-time estimation for influence maximization in large-scale social networks. In: SNS, pp. 1–6, Switzerland (2012)
13. Liu, X., Li, M., Li, S., Peng, S., Liao, X., Lu, X.: IMGPU: GPU-accelerated influence maximization in large-scale social networks. In: TPDS, pp. 136–145 (2014)
14. Mislove, A., Koppula, H.S., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Growth of the Flickr social network. In: SNS, pp. 25–30, USA (2008)
15. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in Facebook. In: SNS, pp. 37–42, Spain (2009)