Layer-Prioritized Influence Maximization in Social Networks

Qianwen Zhang, Yuzhu Wu, and Jinkui Xie(✉)

Department of Computer Science and Technology, East China Normal University,
Shanghai 200062, China
zqw1005@126.com, cstxpxz@163.com, jkxie@cs.ecnu.edu.cn

Abstract. Influence maximization, first proposed by Kempe, is the problem of finding seed nodes that maximizes the number of affected nodes. However, not only influenced number, but also influence layer is a crucial element which may play an important role in viral marketing. In this paper, we design a new framework, *layer-prioritized influence maximization* (LPIM), to address the problem of influence maximization with an emphasis on influence layer. The proposed framework is mainly composed of three parts: (1) graph clustering. (2) key node selection. (3) seed node detecting. We also demonstrate the effective and efficient of our proposed framework by experiments on large collaboration networks and complexity analysis respectively.

Keywords: Social networks · Influence maximization · Layer-prioritized

1 Introduction

In recent years, various online social networks have emerged. Many social networks, such as Facebook, Google+, Flickr, Weibo, and Youtube, help strengthen individuals' relationships online, and make it easy to propagate information via word-of-mouth effect. This phenomenon has been found useful for viral marketing, social influence maximization, etc. For example, to promote a film, a company may give some free tickets to influential users, hoping they will recommend to their friends. The problem of finding individuals who can trigger maximum adoptions was defined as *Influence Maximization Problem* (IM problem) by Kempe [1], attracting a lot of research interest [2–5].

While modeling the process of influence propagation, there are two basic diffusion models: *Linear Threshold Model* (LT model) proposed by Granovetter and Schelling [6,7], and *Independent Cascade Model* (IC model) proposed by Goldenberg et al. [8,9]. Several studies aim at addressing the IM problem under these models. In Ref. [10], Kimura and Saito presented shortest-path based on IC model for finding sets of influential nodes. In Ref. [11], Chen et al. further improved the greedy algorithm, greatly decreasing the time while keeping closed influence spread. They continued to propose PMIA model in Ref. [12], maintaining good

D. Liu et al. (Eds.): ICONIP 2017, Part V, LNCS 10638, pp. 631–638, 2017.
https://doi.org/10.1007/978-3-319-70139-4_64

balance between efficiency and effectiveness, which is a popular algorithm to select seed sets. Purohit et al. proposed influence-based coarsening for networks, and obtained greatly speed-up for tackling IM problem [13]. In Ref. [14], Chen et al. selected influential individuals by exploring the community structures, improving efficiency and scalability with almost no compromise of effectiveness.

The above work mainly focus on the number of affected individuals and time consumption for solving the problem. However, besides the affected number, affected range is also a key target. Consider the following scenario as a motivating example. To issue some alert, like severe weather forecasting or disease prevention, concerned organizations may particularly inform several people, with the hope to spread widely. In this situation, it is reasonable to consider influence range besides the total number of influenced people. In this paper, we introduce influence layer as an indicator to evaluate affected range. Furthermore, to address the problem of influence maximization with an emphasis on influence layer, we design a new framework, *layer-prioritized influence maximization* (LPIM), which comprises three phase: (1) graph clustering, (2) key node selection, (3) seed subgraph(node) detecting. Since we hope the chosen individuals spread out as much as possible, we adopt spectral clustering methods in phase (1). The two remaining questions how to select key node and detect seed subgraph will be addressed in Sect. 2.

Organization. In Sect. 2, we describe the research problem, further detail the LPIM framework and associated algorithms. In Sect. 3, we present the experiments on several real datasets. We conclude the paper in Sect. 4.

2 Layer-Prioritized Influence Maximization

In this section, we first provide a review of the influence maximization problem, including its definition and popular IC model. Then we present the proposed layer-prioritized influence maximization framework and our approaches dealing with the arising issue.

2.1 Influence Maximization: Review

We consider a social network to be an influence graph $G = (V, E)$, where each vertex $v \in V$ represents a user, and $e \in E$ represents the link between them. For every edge $(i, j) \in E$, p_{ij} denotes the probability that j is activated by i through the edge after i is activated.

The Independent Cascade model is a popular diffusion model used to model the influence propagation. Given a seed set S, the IC model works as follows. Let S_n be the set of vertices that are activated in the nth round, with $S_0 = S$. In the $n + 1$ th round, each newly activated vertex v_i may activate its neighbor v_j which is not yet activated with an independent probability of p_{ij}. This process is repeated until S_n is empty. Note that each activated vertex only has one chance to activate its neighbors. Use $\sigma(S)$ to denote the expected number of active vertices when the process finishes, which we call *influence spread*.

Let X represents a set of vertices, $f(X)$ denotes the vertices activated by X. During the random process of propagation in the IC model, given a seed set S, the influence spread can be written as follows:

$$\begin{cases} I^0(S) = \emptyset, \quad I^1(S) = S, \\ I^{n+2}(S) = I^{n+1}(S) \cup f(I^{n+1}(S) \backslash I^n(S)) \quad \text{for } n \geq 1. \end{cases}$$

where $I^n(S)$ denotes the set of activated vertices at nth round by the seed set S. The expected influence spread $\sigma(S)$ is $|I^\infty(S)|$. Based on the above definition, we formalize the IM problem as below:

Problem 1 (IM Problem). For a graph $G = (V, E)$ and a positive integer k, compute the seed set

$$S^* = \underset{S \subset V \wedge |S|=k}{\arg\max} \ \sigma(S).$$

2.2 Layer-Prioritized Influence Maximization Framework

Original algorithms of influence maximization target at finding individuals that maximize the influence spread. However, the chosen individuals may gather under specific situation. For example, two individuals have a common social circle, and both of them are influential among their friends. Some previous work may choose both of them, because of their high-impact, while ignoring the distribution of the chosen seeds. Based on the above observation, we define *influence layer* to evaluate the range of influenced area as below:

Definition 1 (influence layer). *Given seed nodes S, the influence layer $\phi(S)$ refers to the sum of length of distinct longest path influenced by each seed node under influence cascade model.*

Our proposed LPIM framework, emphasizing on influence layer, aims at finding seeds with appropriate distribution and less compromise of influence spread. Figure 1 shows the overview of the LPIM framework, including three phases: (1) graph clustering. Since we want the chosen individuals spread out as much as possible, we adopt spectral clustering methods to obtain subgraphs; (2) key node selection. For each subgraph, we select an influential node; (3) seed subgraph detecting. Among all subgraphs who have different status in social graph, we detect the most influential subgraphs, whose quantity equals to the number of seed nodes defined in IM problem.

Key Node Selection. Given subgraphs, phase (2) of PLIM is to select an influential node for each subgraph. But how to identify the high-impact node in subgraph is a problem. In this part, we give two strategies of selecting key node. One is to regard the node with highest degree as key node. Another strategy is based on the observation of influence propagation, we consider the node who have close connection with others to be influential nodes. Specifically, we adopt the thought of random walk, choose a few nodes to be initial points, and let them propagate influence for T steps. By repeating the process, the node who has been affected most times is considered to be influential.

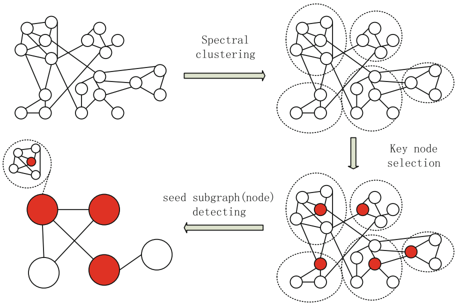


Fig. 1. Overview of the LPIM framework

Formally speaking, since each vertex has a chance to influence its neighbor under IC model, we define simulate influence probability $q_{ij}(t)$ from jth vertex to ith vertex at tth step as

$$q_{ij}(t) = \begin{cases} 1 & \text{if } \mathrm{rand}() \leq p_{ji} \\ 0 & \text{otherwise} \end{cases},$$

then we denote by $x_i(t)$ whether the ith vertex is newly influenced at step t as

$$x_i(t) = \begin{cases} 1 & \text{if } \sum\limits_j q_{ij}(t)x_j(t-1) \geq 1 \\ 0 & \text{otherwise} \end{cases},$$

with the initial value where $x_i(0) = 1$ if ith vertex chosen to be initial point. Then, we have

$$Y = \sum_{t=1}^{T} Q(t) \cdot X(t-1), \tag{1}$$

in which

- $Q(t) = \{q_{ij}(t)\}_{n \times n}$ is a matrix consisting of simulate influence probability at tth step,
- $X(t)$ is the column vector consisting of each $x_i(t)$, viz. $X(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^{\mathrm{T}}$.
- Y is the column vector consisting of each y_i, which represent how many times the ith vertex has been influenced during the whole process.

As mentioned, we repeat the process several times, then regard the node with biggest y_i as key node.

Seed Subgraph Detecting. In phase (3), we aim at detecting important subgraph among all clusters in phase (1). Since finding nodes with as much influence as possible is one of our purpose, we decide using PMIA [12] which is a fast and popular algorithm among existing influence maximization algorithms to identify seed subgraphs. The problem is how to define the weight between each subgraph which we consider as a vertex.

Notice in phase (2), we have selected key node for each subgraph, the procedure of influence propagation between each subgraph could be treated as the procedure between each key node of subgraph from the overall framework.

For a shortest path $SP = <u = a_1, a_2, \cdots, a_d = v>$, we define propagation probability, $P(SP_{u,v})$ as

$$P(SP_{u,v}) = \prod_{i=1}^{d-1} p(a_i, a_i + 1)$$

For two subgraphs c_i and c_j, let v_i and v_j to denote chosen seed node respectively. For an edge connecting subgraphs with vertices $u_1 \in c_i$ and $u_2 \in c_j$, we define propagation probability of the effective path, $P(EP_{v_i,v_j})$ as

$$P(EP_{v_i,v_j}) = P(SP_{v_i,u_1}) \cdot p(u_1, u_2) \cdot P(SP_{u_2,v_j}) \tag{2}$$

Figure 2 shows how we define the weight between two subgraphs. Since we have selected the key node of each subgraph, we convert the subgraph to a tree with the chosen node as root. According to the form of (2), we further calculate the propagation probability of the effective path, e_1 and e_2 in this example. Then we abstract the relationship between two subgraphs as average propagation probability of all effective paths between them.

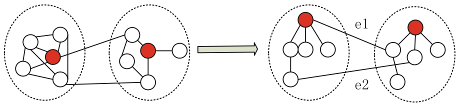


Fig. 2. Subgraph transform

Then we use PMIA algorithm on the new graph, whose vertices are key nodes we selected in phase (2). Once we detect the seed subgraphs, key nodes of these subgraphs are final seed nodes we target at.

Based on three phases, the algorithm for our proposed LPIM framework is shown in Algorithm 1.

Algorithm 1. LPIM(G, k)

```
Input:    Graph of a social network $G(V, E)$; number of seeds set k
Output:    k seeds
$C_s = \{c_1, c_2, \cdots, c_m\} \leftarrow$ spectral clustering
for each $c_i \in C_s$ do
    $v_i \leftarrow$ key node selection
    $c'_i \leftarrow$ converted rooted tree of $c_i$
end for
update weight between each pair of $v_i$
$V' = \{v_1, v_2, \cdots, v_m\}$
$S \leftarrow$ PMIA on $G'(V', E')$
return  $S$;
```

3 Experiments

We conduct experiments on several real-life networks. Our experiments aim at illustrating the effectiveness of our proposed LPIM.

Datasets. We use three real social network datasets. Two collaboration networks NetHEPT and NetPHY are obtained from arXiv.org in the High Energy Physics Theory and Physics domains respectively. Another dataset is DBLP, which is an academic collaboration network. In these datasets, each vertex in the network represents an author, and an edge between a pair of vertices represents their co-authorship. These datasets are commonly used in the literature of influence maximization [1,11,15,16].

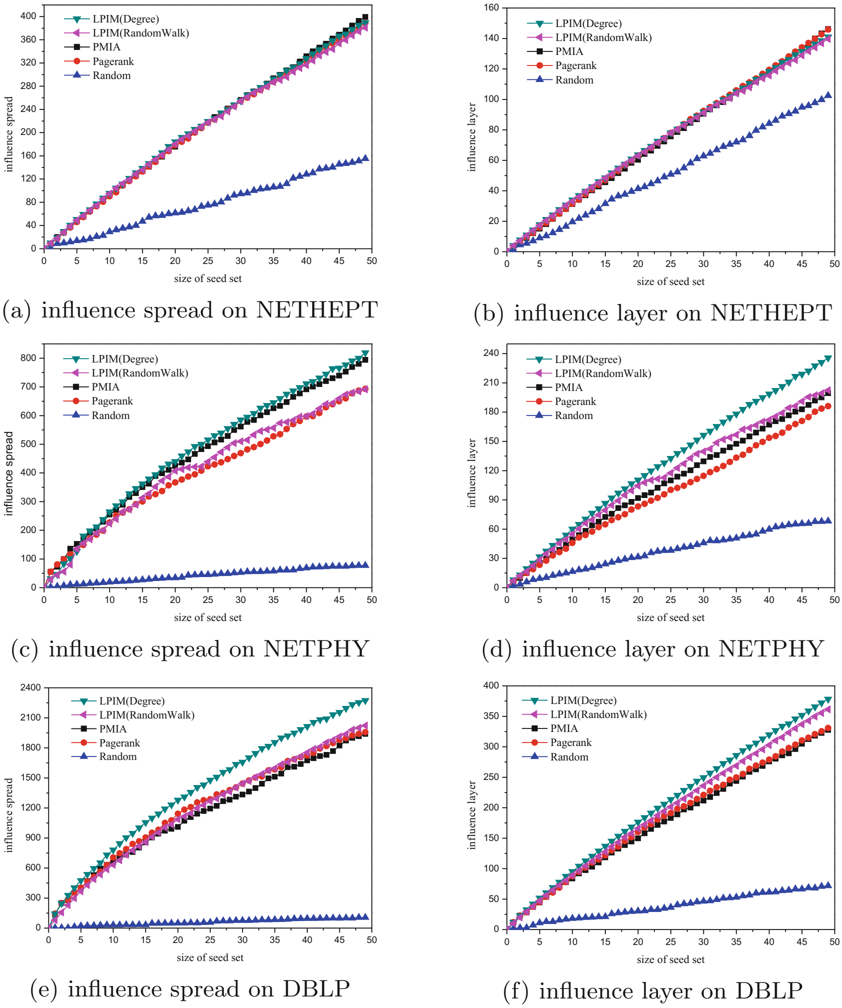
Since our algorithm base on general IC model in which weighted cascade model is usually adopted to obtain the probabilities [1]. We set $p_{ij} = 1/d(j)$ for an edge e_{ij}, where $d(j)$ is the in-degree of jth vertex.

We compare our LPIM algorithm with several other methods. LPIM(Degree) is our algorithm in which selects highest degree node in phase (2) for LPIM model. LPIM(RandomWalk) is our algorithm with another strategy of choosing key node. PMIA [12], proposed by Chen et al. is a heuristic algorithm which is a fast and popular solution to influence maximization problem. Pagerank is a well known algorithm for ranking web pages [17]. We also adopt Random algorithm that selects k random vertices in the graph as a baseline comparison.

Figure 3(a) shows the experimental results on NETHEPT, The x-axis indicates number of seed nodes and y-axis indicates influence spread, namely the number of affected nodes. In Fig. 3(b), the y-axis indicates influence layer. For graph NETHEPT, different algorithms other than random method have similar performance, which may result from the structure of graph. Figure 3(c) and (d) show the result on NETPHY dataset. It is clearly that LPIM(Degree) perform best among these methods on both influence spread and influence layer. LPIM(RandomWalk) is slightly better than PMIA with regard to influence layer, but simulated influence spread is worse than LPIM(Degree) and PMIA method. The results on DBLP dataset is similar to the NETPHY dataset.

Based on the experimental results, we find LPIM(Degree) is better than LPIM(RandomWalk). The phenomenon may be explained as that when we use the idea of random walk to select key node, top p nodes perform stable while we only choose one node for each subgraph. So we adopt selecting key node with highest degree in phase (2), which is better and faster.

Time Complexity. In this part, we present an analysis of the time complexity of the algorithm LPIM. Given a network with n vertices and m edges, we first adopt a spectral clustering algorithm CASP [18] for graph clustering, whose time complexity is $O(p^3)+O(nlogn)$, where p is the number of clusters and $(p << n)$. While selecting key node of each subgraph by choosing the node with highest degree, time complexity is $O(n)$. In phase(3), the time complexity of calculating the weight between connected subgraphs is $O(mlogm)$ whereas using PMIA to choose k influential vertices on construct graph cost $O(kp^2logp)$. So, the total time complexity is within $O(nlogn) + O(mlogm) + O(p^3)$.



(a) influence spread on NETHEPT

(b) influence layer on NETHEPT

(c) influence spread on NETPHY

(d) influence layer on NETPHY

(e) influence spread on DBLP

(f) influence layer on DBLP

Fig. 3. Experimental results

4 Conclusion

In this paper, we have proposed the LPIM framework to solve the IM problem emphasizing on influence layer which we defined to evaluate the range of influence propagation. Experiments showed our algorithm performed well on both influence layer and influence spread compared to other methods.

For further work, to study the influence graph more realistically, we are interested at designing a clustering algorithm to obtain better clusters. Furthermore, we also intend to study the strategy of choosing seed nodes whose influence propagation could achieve maximal influence layer.

References

1. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD, pp. 137–146 (2003)
2. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 306–311. Springer, Heidelberg (2007). doi:10.1007/978-3-540-77105-0_31
3. Tang, J., Sun, J., Wang, C., Yang, Z.: Social influence analysis in large-scale networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 807–816 (2009)
4. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: A data-based approach to social influence maximization **5**(1) (2011)
5. Aslay, C., Barbieri, N., Bonchi, F., Baeza-Yates, R.: Online topic-aware influence maximization queries. In: International Conference on Extending Database Technology (2014)
6. Granovetter, M.: Threshold models of collective behavior. Am. J. Sociol. **83**(6), 1420–1443 (1978)
7. Schelling, T.: Micromotives and Macrobehavior. Norton, New York (1978)
8. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: a complex systems look at the underlying process of word-of-mouth. Mark. Lett. **12**(3), 211–223 (2001)
9. Goldenberg, J., Libai, B.: Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata. Acad. Mark. Sci. Rev. (2001)
10. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 259–271. Springer, Heidelberg (2006). doi:10.1007/11871637_27
11. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: KDD, pp. 199–208 (2009)
12. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD, pp. 1029–1038 (2010)
13. Purohit, M., Prakash, B.A., Kang, C., Zhang, Y., Subrahmanian, V.S.: Fast influence-based coarsening for large networks. In: Proceedings of the 20th ACM SIGKDD, pp. 1296–1305 (2014)
14. Chen, Y., Zhu, W., Peng, W., Lee, W., Lee, S.: CIM: community-based influence maximization in social networks. ACM Trans. Intell. Syst. Technol. **5**(2), 25 (2014)
15. Lei, S., Maniu, S., Mo, L., Cheng, R., Senellart, P.: Online influence maximization. In: Proceedings of the 20th ACM SIGKDD, pp. 645–654 (2014)
16. Zhang, Q., Huang, C.-C., Xie, J.: Influence spread evaluation and propagation rebuilding. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) ICONIP 2016. LNCS, vol. 9948, pp. 481–490. Springer, Cham (2016). doi:10.1007/978-3-319-46672-9_54
17. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In: International Conference on World Wide Web, pp. 107–117 (1998)
18. Zhu, M., Meng, F., Zhou, Y., Yuan, G.: An approximate spectral clustering for community detection based on coarsening networks. Int. J. Adv. Comput. Technol. **4**(4), 235–243 (2012)