ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
WCSE_2019_SUMMER
Hong Kong, 15-17 June 2019, pp. 807-814
doi: 10.18178/wcse.2019.06.120

Distance-Aware Influence Maximization Algorithm based on Random Walk

Yuwei Wang[1,a,+], Ling Chen[1,2,b]

[1]Department of Computer Science, Yangzhou University, Yangzhou,225009,China
[2]State Key Lab of Novel Software Tech, Nanjing University, Nanjing, 210093, China

Abstract. In this paper, we investigate the distance-aware influence maximization problem on the independent cascade(IC) model. A random walk based algorithm is presented to find the seed set to maximizing the influence for a distance aware query. Random walk method is used to perform path sampling to simulate the influence propagation process. Based on the result using random walk method, greedy method is used to select the seed set. Our experimental results show that the algorithm can reasonably select the seed set to maximize the influence propagation.

Keywords: Influence maximization, Distance-aware, Random walk, Greedy method

1. Introduction

In recent years, with the development of mobile Internet, the mining of social networks has been attached more and more importance. It is important to study the spreading of the influence by some influential users in social networks which are called “seeds”. Kempe[1] et al. first defined the problem of influence maximization, that is to detect *k* seeds such that the number of users finally being influenced is maximized under a certain propagation model. Independent cascade model (IC) and linear threshold model (LT) are two classical models.

With the emergence of some devices that can record the geographical location, users' check-in information can be recorded. Therefore, it is more and more necessary and important to use geographic information in social networks analysis. In recent years, Li et al. first studied the location-aware influence maximization, In [2], each user has a location in the two-dimensional space. Given a query region *R*, it aims to find the seed node *S* with the size of *k* that has the greatest influence in region *R*, namely, the seed set *S* can finally activate the largest number of users in region *R*. In the location-aware influence maximization problem [3] proposed by Xiao Li et al., multiple check-in locations of users are used, and geographical preference in a query region *R* is taken into account in the calculation of influence. But in both models, choosing the right size area is not easy ignoring the importance of the user and the target location. In [4], the distance between the user and the target promotion location is considered, and the MIA model is expanded to calculate the influence. In order to reduce the nodes to be evaluated, three pruning strategies are proposed.

2. Concept and Definitions

In this work, we investigate IM on independent cascade model (IC). Given the network $G=<V,E>$, each node $v \in V$ has a geographic location (*x, y*),where *x* and *y* represent the two coordinates of *v* respectively. Given a target node *q*, and a function $d: V \times q \to R$ assigns a distance weight to each node, so that the node

+ Corresponding author. Tel.: +86-183-6282-6387; fax: +86-514-87887937.
E-mail address: [a] 18362826387@126.com[b]yzulchen@163.com

closer to q has a greater weight. In this paper, we define $d(v,q) = ce^{-\alpha d'(v,q)}$ which is a widely used decay function [5]. In the function, α denotes the weight decay speed, c is the maximum weight that a node can achieve, and $d'(v,q)$ is the Euclidean distance between u and q.

For a seed set S, its influence propagation $I(S)$ is defined as the expected value of the number of nodes activated during propagation. Influence propagation of S is calculated by the following formula:

$$I(S) = \sum_{v \in V} F_V(v,S) \tag{1}$$

Here, $F_V(v,S)$ is the probability that the seed set S activates v under the IC model.

Definition 1 (Distance-aware influence spread) Given the geographical social network $G = (V,E)$ and a location q in a two-dimensional space, for a group of seed nodes S, distance aware influence spread $I_q(S)$ is denoted as:

$$I_q(S) = \sum_{v \in V} F_V(v,S).d(v,q) \tag{2}$$

where $d(v,q)$ is the distance weight of v.

Given a geographic social network G, a query location q and an integer k, the influence maximization based on distance-aware is to find a set of seed set S^* with k nodes, so that it has the maximum influence spread in G. That is:

$$S^* = \arg\max_{S \in V, |S|=k} I_q(S) \tag{3}$$

If the parameter in $d(v,q) = ce^{-\alpha d'(v,q)}$ is set as $c = 1$ and $\alpha = 0$, the above distance-aware influence maximization becomes the traditional influence maximization problem. Since the influence maximization problem is *NP-hard*, the above distance-aware influence maximization problem is also *NP-hard*.

3. Path Sampling based on a Random Walk

In (1), $F_V(v,S)$ is the probability that the seed set S activates v under the IC model. $F_V(v,S)$ can be calculated by the following iteration:

$$F_V(v,S) = \begin{cases} \sum_{w \in N(v)} p_{vw}.F_V(w,S) & v \notin S \\ 1 & v \in S \end{cases} \tag{4}$$

where $N(v)$ is the set of neighbor nodes of v.

In this paper, the method of random walk is used to compute $F_V(v,S)$ by simulating the process of influence propagation. In the method, it is assumed that a particle starts from v and performs random walk according to the propagation probability of each edge to simulate the reverse process of influence from S to v. Each node passing on the path is likely to activate v, and the summation of the probabilities of such nodes in S is $F_V(v,S)$.

Definition 2 We define $f_v(v,S,t)$ as the probability that a particle starts from v and arrives at a certain node in S at step t, then for a node v that is not in S:

$$F_V(v,S) = \sum_{t=1}^{\infty} f(v,S,t). \tag{5}$$

In (5), we need to sum up $f_v(v,S,t)$ over an infinite number of steps t. We notice that t is actually the length of the path of random walk. Since the probability of a long path is very small, we can limit the length of the path within a range L so that the error of the result is less than the given ε.

We use random walk to calculate $f_v(v,S,t)$. Assume that a random walking particle starting from v reaches node u_t at step t, we define:

$$g(t) = \begin{cases} 1 & u_t \in S \\ 0 & u_t \notin S \end{cases}.$$

Then $f_v(v,S,t) = E[g(t)]$, $F_V^L(v,S) = \sum_{t=1}^{L} f(v,S,t) = \sum_{t=1}^{L} E[g(t)]$.

Here, $E[g(t)]$ can be calculated by Monte-Carlo method using multiple random walks. Let R be the repeat times for random walks. Assume in the r-th random walk, the particles starting from v reaches node u_t^r at step t, we define:

$$g^r(t)=\begin{cases}1 & u_t^r \in S\\ 0 & u_t^r \notin S\end{cases}$$

Then $E[g(t)]$ can be estimated by $\frac{1}{R}\sum_{r=1}^{R} g^r(t)$, and $F_V^L(v,S)$ can be estimated by

$$\bar{F}_V^L(v,S)=\sum_{t=1}^{L}\frac{1}{R}\sum_{r=1}^{R} g^r(t)=\sum_{t=1}^{L}\sum_{r=1}^{R}\frac{g^r(t)}{R} \tag{6}$$

Based on the above analysis, we need to record all the paths of random walks. In this paper, greedy algorithm is used as the framework. When selecting the elements of the seed set S, random walk needs to be iteratively repeated many times. To reduce the computation time, the paths are recorded to avoid repeated random walks in the subsequent seed selections.

The framework of algorithm *Create_path* is as follows:

Algorithm 1: Create_path

Input: *G=(V,E,P)*:a geo-social network;
L: Maximum length of the random walk path;
q: The specified target node;
R: Number of samplings;

Output: *Path*: an array of random walks;
H(*x*):The set of nodes that *x* can activate;
Loc(*v,r,u*): position of *u* on the route of the r-th random walk starting from *v.*
δ(*x*): The expected sum of the nodes distance weights that *x* can activate;
Len(*v,r*): The length of the path of the r-th random walk from *v*;

Begin

1: **For** each $v \in V$ **do**
2: **For** *r* =1 **to** *R* **do**
3: *u*←*v*; *Len*(*v,r*)=0;
4: **for** *l*=1 **to** *L* **do**
5: Selecting a neighbor *w* of *u* by the "roulette" method;
6: **If** the "roulette" is not succeed **then** break **endif;**
7: *Path*(*v,r,l*)=*w*; *H*(*w*)= *H*(*w*) ∪ {*v*}; *Loc*(*v,r,w*) =*l*;
8: $\delta(w)=\delta(w)+d(v,q)/R$; *Len*(*v,r*)= *Len*(*v,r*)+1; *u*←*w*;
9: **Endfor** *l*;
10: **Endfor** *r*
11:**End**

We set $|V|=n$, the complexity of algorithm 1 is obviously $O(n \cdot L \cdot R)$. Since L and R are constants, the complexity of the algorithm is O(n).

4 Greedy Algorithm for Finding the Seed Set

We use greedy algorithm to select the seed set. The basic idea is to sequentially select nodes x in $V\backslash S$ which can maximize $\Delta(x)=I_q(S\cup\{x\})-I_q(S)$ to join S. To select such node x to join S, we have to simulate the propagation process of $S\cup\{x\}$ as the seed set under the IC model. This propagation process is a #p problem, which requires a large amount of computing time. Therefore we use the above sampling method

to simulate the propagation process, and record the paths of the random walks. To this end, we first give the following theorem to estimate $\Delta(x)$.

Theorem 1. Let S be the current seed set, for the node x in $V\backslash S$, we have:

$$\Delta(x) = [1 - F_V(x,S)] \sum_{v \in V} F_{V\backslash S}(v,\{x\}).d(q,v) \tag{7}$$

Here, $F_{V\backslash S}(v,\{x\})$ denotes the probability that v is activated as a seed set with $\{x\}$ in the sub-graph of V\S as a set of nodes.

To estimate $\Delta(x)$, we use the result $\bar{F}_V^L(v,S)$ of the random walk method instead of $F_V(x,S)$ in (7). By the process of random walk:

$$\bar{F}_V^L(v,S) = \frac{1}{R} \sum_{r=1}^{R} \sum_{l=1}^{L} I[path(v,r,l) \in S] \tag{8}$$

In (8), the indicator function $I[T]$ is defined as: if T is true, $I[T]=1$; otherwise $I[T]=0$.

Therefore, (8) is an estimate of the probability that the seed set S activates v in random walks.

For $\sum_{v \in V} F_{V\backslash S}(v,\{x\}) \cdot d(q,v)$ in (7), we also use the results of the method of random walks to calculate. We denote

$$\delta_q(S,x) = \sum_{v \in V} F_{V\backslash S}(v,\{x\}).d(q,v) \tag{9}$$

So we have

$$\Delta(x) = [1 - F_V(x,S)] \delta_q(S,x) \tag{10}$$

It can be seen from (9), $\delta_q(S,x)$ is the sum of the distance weights of the nodes that x can actually activate, that is, the sum of the distance weights of the starting points of all the paths that can reach x but not the nodes in S in the random walk. By replacing $F_{V\backslash S}(v,\{x\})$ in (9) with the result of the random walk $\bar{F}_{V\backslash S}^L(v,\{x\})$, we can get

$$\delta_q(S,x) = \sum_{w \in H(x)} \frac{1}{R} \sum_{r=1}^{R} d(q,w) \cdot I[loc(w,r,x) < \min_{u \in S} loc(w,r,u)] \tag{11}$$

Therefore, (11) is the expected value of the distance weight sum of the starting points of all the paths that can reach x but not pass S in the random walk.

5. Framework of the Algorithm

We use the greedy algorithm to select the seed set S. Initially, we set S as an empty set, and for all $v \in V$: $\bar{F}_V^L(v,S) = 0$. The algorithm sequentially selects nodes x from $V\backslash S$ with the largest $\Delta(x)$ to join S. The initial value of $\delta_q(S,v)$ is the output $\delta(v)$ of algorithm 1. At each time the new node x that maximizes $\Delta(x)$ is added to S, $\bar{F}_V^L(v,S)$ and $\delta_q(S,v)$ should be updated accordingly. We can get the $\bar{F}_V^L(v,S \cup \{x\})$ and $\delta_q(S \cup \{x\},v)$ by incremental calculation based on $\bar{F}_V^L(v,S)$ and $\delta_q(S,v)$, so that the new value of $\Delta(v)$ can be calculated according to (7).The rules for updating $\bar{F}_V^L(v,S)$ and $\delta_q(S,v)$ are as follows:

$$\bar{F}_V^L(v,S \cup \{x\}) - \bar{F}_V^L(v,S) = \frac{1}{R} \sum_{r=1}^{R} \sum_{l=1}^{L} I[(path(v,r,l) = x) \text{ and } (l < \max_{u \in S} loc(v,r,u))] \tag{12}$$

$$\delta_q(S,v) - \delta_q(S \cup \{x\},v) = \sum_{w \in H(v)} \frac{1}{R} \sum_{r=1}^{R} d_q(w).I[loc(w,r,x) < loc(w,r,v) < \min_{u \in S} loc(w,r,u)] \tag{13}$$

Equation (13) shows that after adding x to the seed set, the paths where v located between x and the nearest seed node u in S should be removed from $\delta_q(S,v)$, because v cannot activate the nodes in these paths after x joining S. The framework of the distance-aware influence maximization algorithm based on random walk is as follows:

Algorithm 2: RW_DA (Random Walk based Distance-Aware influence maximization)

Input: *G=(V,E,P)*: a geo-social network;
k: size of the seed set;

*q:*a given target node;

*R:*Number of samplings;

Output： *S:* seed set;

Begin

1: **For** all $v \in V$ **do**

2: Calculate distance score from *v* to *q* : $d(v,q) = ce^{-\alpha d'(v,q)}$;

3: **Endfor** *v*;

4: /*Call algorithm 1 to calculate *Path*、*H*(*x*)、*Loc*(*v,r,u*)、$\delta(x)$、*Len*(*v,r*)etc. */

5： Create_path; $S=\Phi$;

6: **For** all $v \in V$ **do** $F(v)=0$ **endfor;**

7: Select the node *x* with the largest $\delta(x)$ value; $S = S \cup \{x\}$;

8: **For** *i*=1 **to** *k* **do**

9: **For** each $w \in H(x)$ **do**

10: **For** *m*=1 **to** *Len*(*w,r*) **do**

11: *l*=*Loc*(*w,r,x*);

12: **for** *j*=*l*+1 **to** *Len*(*w,r*) **do**

13: *u*=*Path*(*w,r,j*); $\delta(u)= \delta(u)- d(w,q)/R$;

14: **endfor** *j*;

15: *Len*(*w,r*)=*l*-1; $F(w)=F(w)+1/R$;

16: **Ednfor** *m;*

17: **Ednfor** *w*;

18: **For** all $v \in V \setminus S$ **do** $\Delta(v) = [1 - F(v)]\delta(v)$ **endfor** *v;*

19: Select the node *x* with the largest $\Delta(x)$ value: $S = S \cup \{x\}$;

20: **Ednfor** *i;*

21:**End.**

Let the number of nodes in the network be |*V*|=*n*. In Algorithm 2, *i* loop runs *k* times, the loop of *w* runs *n* times, the *r* loop runs *R* times, the *l* loop runs *L* times, total time of the loops is obviously $O(n \cdot L \cdot R \cdot k)$. Since *k*, *L*, and *R* are all constants, the complexity of the algorithm is O(*n*).

6. Experiment and Analysis

6.1. Datasets and experiment settings

To test the accuracy of the proposed algorithm RW_DA, we use the following two real-world datasets of social network with geographic information:

1) Brightkite(BK): it consists of 58K nodes and 428K edges.

2）Gowalla (GW): it consists of 197K nodes and 1.9 million edges. Due to the large size of GW data set, we randomly and uniformly select 40% of the nodes to for form a sub-network to conduct the experiment.

On these two real data sets, each user has multiple check-in records. We counts the number of check-ins in different locations of the user, selects the location with the most frequent check-ins as the geographic location of each user. For a few users who have no check in record, we randomly assign each of them a location in the area.

In this work, we use the following two methods [8]to generate propagation probabilities on the edges.

(1) WC model. That is, the weight model, the propagation probability on edge (*u,v*) is $p_{u,v} = \frac{1}{N_v}$, where

N_v refers to the number of neighbors of the node v.

(2) TC model. The probability of propagation on the edge is randomly chosen from {0.1, 0.01, 0.001}, which represent the effects of high, medium, and low, respectively.

In order to evaluate the effect of the proposed algorithm RW_DA for distance-aware influence maximization problem, we tested each seed set many times, and use the average influence spread of the seed set as the result. Since 40% of the nodes are extracted from the GW data set, we quantify the influence spreading using the percentage of the final spread in the total sampled nodes. The target query location q is randomly selected from the space region. The number of seed sets k is distributed from 10-50. In the weight function of the nodes, we set up c as 10, αas 0.02. In this paper, the sampling number R is set as 500, and the path length L is set as 20.

6.2. Comparison algorithm and experimental environment

In order to evaluate the effect of the proposed algorithm RW_DA, we conducted experiments on the data set and the two propagation models described above, and compared the performance with the algorithms PMIA [6] , PR [4], and degree-discount (DD) [7].

PMIA is an algorithm improved by Wei Chen et al. based on MIA[6]. In order to support the influence maximization algorithm, the distance weights are considered, and *MIIA(u)* and *MIOA(u)* are pre-computed. PR algorithm [4]was proposed by Xiaoyang Wang et al. for the problem of distance-aware influence maximization. Since influence or marginal gain needs to be calculated for many nodes, three pruning rules are used in their method to subtract meaningless nodes. Degree-discount (DD) algorithm is short for DD algorithm is proposed by Wei Chen et al.

We take q as the center to form a region. The nodes outside the region have very little chance to be successfully affected. In our experiments, those remote nodes are used to show the impact of distance weight in influence maximization.

The algorithm is coded by Matlab and run on a PC with 2.6GHz Intel I7 6500U, 12GB memory under Microsoft Windows 10.

6.3. Experimental results and analysis

6.3.1. The result on the Brightkite dataset

The experimental results of the Brightkite dataset on the WC model and the TC model are shown in Figures 1 and 2, respectively.

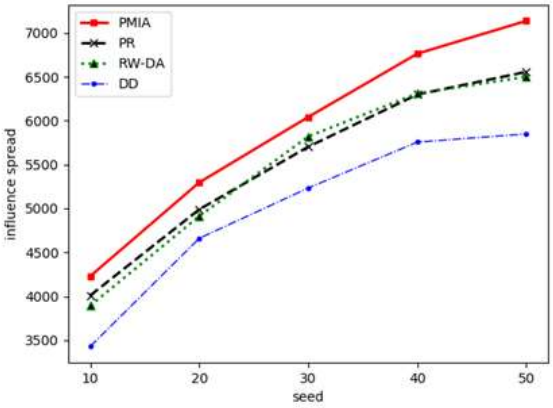

Fig. 1: Results of the Brightkite dataset on the WC model

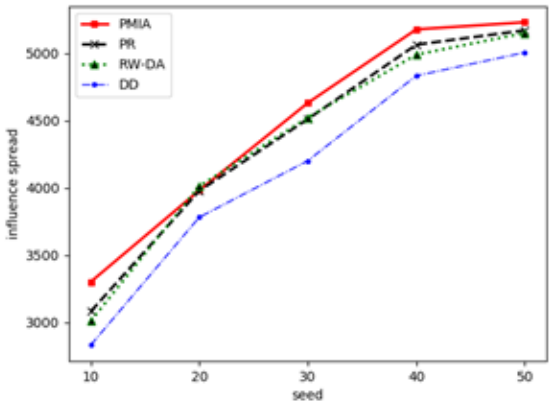

Fig. 2: Results of the Brightkite dataset on the TC model

Overall, as the number of seeds increases, the spread of influence of all algorithms increases. And the number of propagation results under the TC model is less than that under the WC model, which is due to the difference in probability distribution at the edge of different models. The PMIA algorithm can achieve slightly better results under both models; the RW-DA algorithm proposed in this paper can achieve very

similar results with PR algorithm. In some cases, the propagation range of RW-DA is larger than PR. This is because the PR algorithm combines three pruning strategies, which may achieve early termination in each iteration and may not obtain the node with the largest marginal gain. RW-DA can achieve better results than DD, and it is more practical to consider distance weights. Therefore, the seed set selected by the proposed algorithm RW-DA can achieve a larger range of propagation.

6.3.2. The result on the Gowalla dataset

The experimental results of the Gowalla dataset on the WC model and the TC model are shown in Figures 3 and 4, respectively:

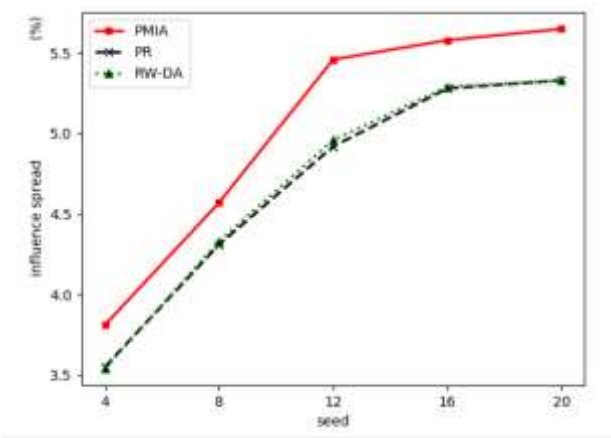

Fig. 3: Results of the Gowalla dataset on the WC model.

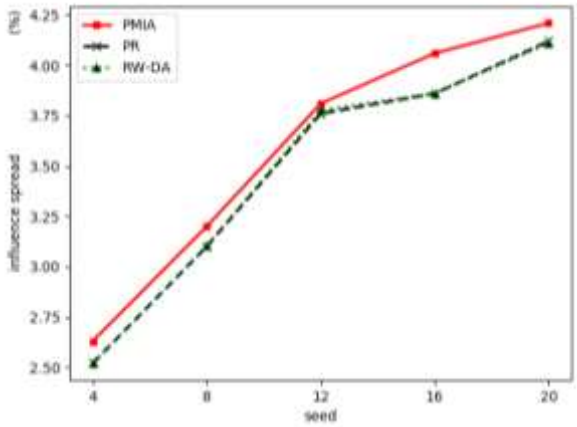

Fig. 4: Results of the Gowalla dataset on the TC model.

Due to the large GW dataset, this paper randomly and evenly extracted 40% of the total number of nodes for experiments. So we compare the percentage of diffusion. The PMIA algorithm can achieve slightly better results under both models; the RW-DA algorithm proposed in this paper can achieve very similar results with the seed set selected by the PR algorithm. In some cases, the propagation range of RW-DA is larger than PR.

7. Conclusion

Many algorithms[7] [9][10]are developed for the traditional influence maximization problem, that is, selecting *k* seed nodes in a social network to maximize the influence diffusion. However, the distance-aware influence maximization problem, which takes the distance factor into consideration, has more practical significance in real life. Based on this background, this paper proposes a method for distance-aware influence maximization under the IC model. The method uses random walks to simulate the influence propagation process, and records the paths of the random walks to avoid repeated simulations during seed selection. The greedy algorithm is used to select seed nodes sequentially. Due to the paths of the random walks are recorded, we can incrementally calculate the marginal gain for each candidate seed node. By adopting the above techniques, the computation time is reduced, and the seed set can be effectively selected to maximize the influence.

8. Acknowledgements

This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61379066, 61379064,61472344, 61402395, Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK20140492.

9. Reference

[1] D.Kemp, J.M. Kleinberg, E. Tardos, *Maximizing the spread in influence through a social network*, in KDD 2003

[2] G.Li,,S.Chen,,J.Feng, K.Tan, and W.Li, *Efficient location-aware influence maximization*, in SIGMOD, 2014, pp.

87–98.

[3] X. Li, X. Cheng, S. Su, and C. Sun, *Community-based seeds selection algorithm for location aware influence maximization*, Neuro computing, vol. 275, pp. 1601–1613, Jan. 2018.

[4] X. Wang, Y. Zhang, W. Zhang, and X. Lin, *Distance-aware influence maximization in geo-social network*, in ICDE, 2016, pp. 1–12.

[5] Y. Wu, S. Yang, and X. Yan. *Ontology-based subgraph querying*. In ICDE, pages 697–708, 2013.

[6] W. Chen, C. Wang, and Y. Wang. *Scalable influence maximization for prevalent viral marketing in large-scale social networks.* In SIGKDD, pages 1029–1038, 2010.

[7] Chen W, Wang Y, Yang S. *Efficient influence maximization in social networks*[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July. DBLP, 2009:199-208.

[8] Goyal A, Lu W, Lakshmanan L V S. SIMPATH: *An Efficient Algorithm for Influence Maximization under the Linear Threshold Model*[C]// IEEE, International Conference on Data Mining. IEEE Computer Society, 2011:211-220.

[9] A. Goyal, W. Lu, and L.V. S. Lakshmanan, "*Simpath: An efficient algorithm for influence maximization under the linear threshold model,*" in ICDM, 2011, pp. 211–220.

[10] K. Jung, W. Heo, and W. Chen, "*IRIE: scalable and robust influence maximization in social networks,*" in ICDM, 2012, pp. 918–923.