Article

Influence Maximization Algorithm Based on Reverse Reachable Set

Gengxin Sun [1], *, Chih-Cheng Chen [2,3] *

[1] School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China; sungengxin@qdu.edu.cn (G. S.);

[2] Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

[3] Department of Aeronautical Engineering, Chaoyang University of Technology, Taiwan 413310, Taiwan

* Correspondence: sungengxin@qdu.edu.cn; ccc@gm.cyut.edu.tw(C.-C C.);

Abstract: Most of the existing influence maximization algorithms are not suitable for large-scale social networks due to their high time complexity or limited influence propagation range. Therefore, a D-RIS influence maximization algorithm is proposed based on the independent cascade model and combined with the reverse reachable set sampling. Under the premise that the influence propagation function satisfies monotonicity and submodularity, the D-RIS algorithm uses automatic debugging method to determine the critical value of the number of reverse reachable sets, which not only obtains a better influence propagation range, and greatly reduce the time complexity. The experimental results on the two real data sets of Slashdot and Epinions show that D-RIS algorithm is close to the CELF algorithm and higher than RIS algorithm, HighDegree algorithm, LIR algorithm and pBmH algorithm in influence propagation range. At the same time, it is significantly better than the CELF algorithm and RIS algorithm in running time, which indicates that D-RIS algorithm is more suitable for large scale social network.

Keywords: social networks; influence maximization; information diffusion model; reverse reachable set; sub-modularity

1. Introduction

Due to the rapid development of social networks, the number of users and the scale of information dissemination continue to expand, the problem of maximizing influence has received more and more attention. It is widely used in "viral marketing" [1, 2]. "Viral marketing" is a way to maximize brand awareness through word-of-mouth effects among users. Therefore, with limited resources, the key to maximizing influence is to select the appropriate initial communication users to maximize the final communication effect.

Richardson et al. [1] regard the problem of maximizing influence as an algorithm problem, that is, under a specific information dissemination model, select k initial seed node sets from a social network to maximize the final influence dissemination range. Kempe et al. [3] proved for the first time that impact maximization is an NP-hard subject based on the Independent Cascade model (IC model) [4] and the Linear Threshold model (LT model) [5]. At the same time, a Greedy Algorithm (GA) is proposed. The algorithm selects the node with the most considerable marginal effect by iteration to ensure that it is close to the optimal solution within the range of $(1-\frac{1}{e}-\varepsilon)$. Due to the high time complexity, this algorithm is not suitable for large-scale social networks. Therefore, many researchers have proposed some optimization algorithms for the low efficiency of greedy algorithms. In 2007, Leskovec et al. [6] proposed the Cost-Effective Lazy Forwards (CELF) algorithm. It uses the characteristics of the inter-node influence propagation function to satisfy the sub-modularity, which increases the running speed of the greedy algorithm by 700 times. In 2011, Goyal et al. [7] proposed the CELF++ algorithm, which further reduced the time complexity of the CELF algorithm. These algorithms have achieved a certain degree of speed improvement. However, each time a node is selected to join the node-set, the increase in the influence of the node is calculated, so the operating efficiency is still very low, and it is difficult to apply to large-scale social networks. At present, most scholars use heuristic algorithms to improve the running speed. Literature [8-9] proposed different influence maximization algorithms on the basis of degree centrality. In 2010, Chen et al. [10] proposed the PMIA algorithm based on the maximum influence propagation path between nodes. In 2012, Jung et al. [11] proposed the IRIE heuristic algorithm for the IC model. In addition, heuristic influence maximization algorithms based on network topology have been proposed successively [12-14]. In 2016, Xie Shengnan et al. [15] proposed a new heuristic algorithm to improve operation efficiency. Cao Jiuxin et al. [16] proposed a CCA algorithm based on K core. Still, these algorithms only focus on the topological structure of the network and lack a specific theoretical guarantee, which may cause the algorithm to fail to obtain the optimal solution. Based on the above problems, Brogs et al. [13] proposed a RIS algorithm that combines theory and actual efficiency. The algorithm selects nodes by generating a certain number of reverse reachable sets and then calculates node influence many times so that The time complexity is close to linear, and there is a specific theoretical guarantee. Although the RIS algorithm has many advantages, it still has disadvantages such as insufficient accuracy and stability in selecting the number of the reverse reachable sets. Therefore, a lot of calculation costs are required in practice.

In this paper, we proposes a Dynamic-Reverse Reachable Set (D-RIS) algorithm based on reverse reachable set. The algorithm does not need to preset the theoretical threshold of the number of reverse reachable sets in advance, but based on the monotonicity and sub-modularity of the influence propagation function, sets the judgment conditions for generating the critical value of the random reverse reachable set, and automatically debugs the generation A certain number of reverse reachable sets can avoid time wastage while obtaining a better influence spread range.

2. Influence maximization algorithm based on reverse reachable set

The social network is abstracted as a network graph G with a node-set V (user) and a directed edge set E (the relationship between users), with $G=(V,P,E)$, $|V|=n$, $|E|=m$, and $p\in(0,1)$. Assume that each edge e in G has a propagation probability $p(E)\in(0,1)$, then $p(u,v)\in p(u,v\in V)$ represents the probability that node u activates node v. For the convenience of presentation, Table 1 lists the symbols commonly used in this article.

Table 1. Frequently used notations.

Notation	Description
G	A social network
n	The number of nodes in G
m	The number of edges in G
k	The size of the seed set of influence maximization

P(E)	The propagation probability of an edge e
S	A node set
I(S)	The spread of a node set S in an influence propagation process on G
E[I(S)]	The maximum propagation expectation for a node set
R	The set of all RR sets generated
R_j	A random RR set
θ	The critical value of the number of RR sets
α	The ratio of the RR sets

2.1. Communication model and question description

When looking for a specific set of seed nodes with the most significant influence in social networks, it is necessary to use a particular spread model to simulate the rules of spreading information on the network. The current classic information dissemination models include IC model and LT model.

The experiment in this article uses the IC model to simulate the maximum spread of user influence. In this model, a directed weighted graph G with n nodes and m edges is given to represent the underlying network. The weight of edge $e = (v, u)$ represents the probability P that node v propagates to node u along edge e. Nodes in the IC model are divided into three states: activated state, newly activated state and inactive state. Each newly activated node has one and only one chance to try to start adjacent nodes that are not activated with probability P. The higher the value of P, the greater the possibility of activation. When there are no influential active nodes in G, the propagation process ends. The influence of propagation simulation on the IC model is started by random propagation from a set of seed nodes. Let $I(S)$ be the number of random nodes eventually infected by the propagation simulation process, and $E[I(S)]$ to be the ultimate propagation impact of the node sets. This model simulates the propagation process of the infectious disease model [15, 16]. The seed set S is similar to a group of infected individuals, and the propagation simulation process of activating its adjacent nodes is identical to the spread of disease from one individual to another.

The following describes an example of influence spread on the IC model.

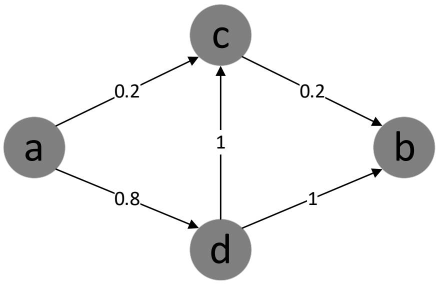


Figure 1. Initial diagram of the influence propagation social network based on the independent cascade model (IC model) seed node set

Figure 1 is an initial graph of a social network composed of four nodes, and the weight on each edge represents the propagation probability from the out-side node to the in-side node. Let the activation probability of all nodes in this social network be 0.5. The following simulates the information dissemination process of the social network in Figure 1. $\mathrm{S} = \{\mathrm{a}\}$ is the initial seed set. Activating node a at time T_1. Then at time T_2, node a has a probability of 0.2 to activate node c and probability of 0.8 to activate node d, because of $\mathrm{p}_{\mathrm{ac}} = 0.5 > \mathrm{p} = 0.2$. At time T_2, node d is activated, $\mathrm{S} = \{\mathrm{a}, \mathrm{d}\}$. At time T_3, node c and node b have a probability of 1 being activated by node d. Suppose that node d activates node c but not node b, which affects the end of the propagation process. Because no new nodes on the network can be activated. The total number of nodes activated in the propagation process is 3, that is $\mathrm{I(S)} = 3$, S={a,b,c}. If node b is activated at time node b, then $\mathrm{I(S)} = 4$, $\mathrm{S} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$. Since the IC model is a probability model [17], the propagation process and the final propagation result are not necessarily. Monte Carlo method [14] is often used in experiments to take the average of multiple runs to ensure the accuracy of the results.

Given the social network G and constants k, the problem of maximizing influence is to find a set of seed nodes S in G, so that it has the widest range of influence under the IC propagation model, that is , find the node set $\mathrm{S} \in \mathrm{V}$ and |S|=k such that $\mathrm{E[I(S)]}$ maximum.

2.2. RIS algorithm

Borgs et al. proposed the Reverse Influence Sampling (RIS) algorithm based on the IC model, which is a completely different influence maximization algorithm from other classic algorithms. The algorithm introduces a novel reverse reachable set (Reverse Reachable Set, referred to as RR set) sampling method to replace the Monte Carlo method to calculate the influence of the expected propagation of nodes. The main idea is to generate as few reverse reachable set samples as possible, and finally obtain a near optimal solution in the range of $(1 - \frac{1}{e} - \varepsilon)$. This algorithm proves that for any $\varepsilon > 0$, it can run in the time of $o(\beta(m+n)k\, log_n)$, and the time complexity is approximately linear time (β is the number of steps to select the reverse reachable centralized operation).

The RIS algorithm avoids the limitation of the high time complexity of the greedy algorithm, and also solves the problem that the heuristic algorithm lacks theoretical guarantee and cannot obtain the optimal solution. But this algorithm cannot effectively control the number of random RR sets. They proposed a threshold-based method to generate random RR sets: when the total number of generated nodes and edges reaches a predetermined theoretical threshold, they stop generating random RR sets. Although this method has approximately linear time complexity, there is a great correlation between the generation of reverse reachable sets of fixed theoretical thresholds, and the hidden constants in practice are large, resulting in two shortcomings in the RIS algorithm: (1) The actual RR set sample size generated is greater than the theoretical threshold. (2) There is no guarantee that the theoretical threshold is the minimum number of samples generated in the RR set. Therefore, the sample size of the RR set selected by this algorithm is not accurate, and it is not well suited for solving large-scale social networks.

2.3. Based on reverse reachable set: D-RIS algorithm

For most of the classic influence maximization algorithms, the time complexity is too high or the optimal solution cannot be obtained. Based on the IC model and combined with the reverse reachable set sampling method, we proposes a D-RIS (Dynamic-Reverse Influence Sampling) algorithm for maximizing influence.

The D-RIS algorithm is divided into two steps:

(1) Generate a reverse reachable set (RR set): Randomly select n nodes with replacement, and generate a set R of θ node RR sets by performing propagation simulation on a random graph g. The value of θ is determined by the method in section 2.3.1.

(2) Node selection: Use the maximum coverage method to find K nodes that cover the most RR sets, and return the node set S.

Analyzing the theory of the RIS algorithm, it can be known that if the sampling number of the random RR set is too small, the algorithm will not get the optimal solution due to insufficient selection of nodes. If the sampling number of the random RR set is too large, although the error is reduced, the time complexity will be too high. Therefore, the accuracy of selecting the seed node set determines the final influence spread range and time efficiency. Therefore, the research focus of the algorithm in this paper is: how to select the smallest possible RR set sample size, so that the algorithm can achieve a better balance between the spread of influence and operating efficiency.

This paper firstly refers to the sampling method in [17] to define a unified reverse reachable set sampling framework. On this basis, Section 2.3.1 puts forward a new critical value judgment method, which can dynamically select as few RR set samples as possible. Finally, Section 2.3.2 uses the maximum coverage method to select the seed node set.

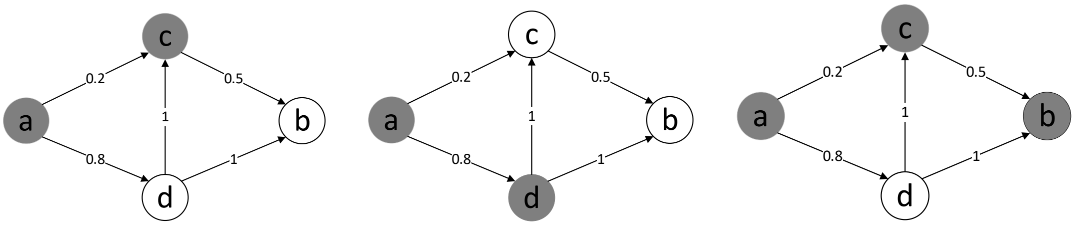
Given a network $G = \left(V, E, P\right)$, the algorithm captures the influence propagation process of nodes in G by generating a set R of random RR sets. Let R_z be a subset of the RR set of node v, that is, the random RR set of the node. Graph g is a random graph obtained by removing edge e in G with a probability of $1 - \mathrm{P(E)}$. The specific definition and sampling process are as follows:

Definition 1 Reverse Reachable Set (RR set)

The set of reachable nodes in the random graph g (for each node u in the RR set, there is a directed path from u to v in g).

Sampling process: (1) Randomly select a node $v \in V$. (2) Generate a sample random graph g on the network G. (3) Return the reverse reachable set R_z of node v in g.

The node v in the above sampling process is called the source in R_z, and all nodes in R_z have a certain probability to activate the source node v. Therefore, the presence of a certain node in more RR sets means that more nodes can be activated, and at the same time, this node can produce a larger influence spread range. Based on the same inference, if the node set S with k nodes covers a large number of RR sets, the k nodes in the network G have strong propagation ability to spread to the maximum range, that is, $I(S) = nPr[S\ Covers\ R_z]$. Therefore, the influence of the node set S is proportional to the probability that S and the RR set intersect. So to solve the problem of maximizing influence is how to determine the lower bound of the R set. Section 2.3.1, based on this reverse reachable set sampling framework, sets up a dynamic debugging method to determine the minimum number of R sets.



(a) Random graph g_1 (b) Random graph g_2 (c) Random graph g_3

Figure 2. Three random graphs generated based on the social network in Figure 2

Use an example to illustrate the process of generating a reverse reachable set for the social network G in Figure 1 under the IC model, and set $k = 1$. Figure 2 shows three random RR sets g_1, g_2, g_3 generated on G. Three random RR sets $R_1 = \{c, a\}$, $R_2 = \{d, a\}$, $R_3 = \{b, c, a\}$ generated for three randomly selected source nodes c, b, d. Because node a appears in three random RR sets, node a is the most influential node. Therefore, the final return result is $S = \{a\}$.

2.3.1 Determination of the number of reverse reachable sets

Analysis of the selection of the number of random RR sets in the RIS algorithm shows that the more the number (the larger the R set), the more accurate the selected seed node set, but it will cause a waste of time. Therefore, this section proposes a method to control the number of generated R sets as small as possible without affecting the final influence spread.

In the experiment of Section 3.2.1, we found that as the number of random RR sets increases, the increase in the spread of influence is not linear but diminishing in utility. Therefore, the relationship between the number of RR sets and the influence propagation range function satisfies both monotonicity and submodularity (diminishing marginal utility), which is defined as follows:

(1) Monotonicity: Set the influence propagation range function f, for any number of reverse reachable sets $q_1 < q_2$, there are $f(q_1) \leq f(q_2)$.

(2) Sub-modularity: For the total number of nodes in the graph t, set the influence propagation range function f, for the number of reverse reachable sets $q_1 < q_2$, and all $a > 0$, if $q_1 < q_2 < t$, there are $f(q_1 + a) \leq f(q_2 + a)$.

Based on the above theory, for a given G, the algorithm sets a critical value θ for the number of random RR sets, where $\theta = n \times \alpha$ (=α is the random RR set selection ratio). When the number of random RR sets is less than θ, the maximum influence spread range cannot be achieved because the number of random RR sets selected is not enough. When the number of random RR sets is greater than θ, due to diminishing marginal benefits, the range of influence increases too slowly or no longer increases, resulting in a waste of time.Therefore, based on the current propagation situation of the nodes in the network, the algorithm automatically doubles the generation of reverse reachable sets in each round until the critical value judgment condition set in Algorithm 1 (line 7) is met three times, and the number of RR sets generated by the algorithm is considered to be infinitely close Critical value. The specific description is as follows:

Set the influence spread range of this round to f_c, and the influence spread range of the last round to f_p. Algorithm 1 gives the pseudo code in the process of generating the reverse reachable set of the D-RIS algorithm. The specific process is:

(1) Set the initial reverse reachable set number ratio to a very small value α（For example, in algorithm 1, the value of α is 0.001, then $\theta = 0.001n$）, randomly select nodes with a ratio of α from the node set S in the graph G to generate an RR set and calculate the impact Power

Algorithm 1 D-RIS algorithm (generate reverse reachable set)

```
Input:  $G = (V, P, E),\ k$
Output: $S,\ \alpha$
        $R = \emptyset,\ \alpha = 0.001, I_p = 0\ ,\ f_p = 0$
        While$(\alpha < 1\&\&flag < 3)$
        Generate a set of seed nodes with $\alpha$ ratio
        $z \leftarrow$ Simulate_ influence_spread()
        generate $\theta$ random RR Sets and add all $R_z$ to $R$
        $f_c = Cnt\_f_c(), I_c = f_c - f_p$
        if $I_c \leq 0$or $(I_p > 0$ and $I_c < math.log(I_p, 2))$
            $flag = flag + 1$
        else $flag = 0, f_p = f_c$
            if $flag = 2$
                break
          $proportion = proportion * 2$
          return $S,\ \alpha$
```

transmission range f_c（Algorithm 1: Lines 4-6）.

(2) Each round doubles the value of α, and calculates the increase in the spread of influence in this round I_C, which is $I_C = f_c - f_p$. The following will make an effective judgment on the increase in the scope of influence in this round（Algorithm 1: 7 lines), if the conditions are met，it is determined that this round of α doubling has no effect on the growth of influence, and may have been close to the critical value.

Judgment condition: if $I_C \leq 0$, or $I_c < mat\hbar.log(I_p, 2)$. That is, the increase in the range of influence of this round is less than or equal to 0 or less than the result of the root sign of the increase in the range of influence of the previous round.

(3) Repeat the above steps until three consecutives α doublings are invalid, or when the value of α is greater than or equal to 1, stop generating reverse reachable sets. At this time, the number of random RR sets generated by the algorithm approaches the critical value.

Suppose the final inverse reachable set ratio is α_1, at this time , a relatively stable and effective critical value of inverse reachable set θ is obtained, and at this time $\theta = n \times \alpha_1$.

In the process of dynamic debugging to determine the value of α, the value of α rises gradually until it approaches the critical value. Except for the first round, each cycle does not generate α proportional reverse reachable sets but generates $\frac{\alpha}{2}$ proportional reverse reachable sets. We will scale the previous round $\frac{\alpha}{2}$ to reverse reachable sets. The reached set is stored to combine the reverse reachable set of the α ratio of the cost round. That is, the same number of reverse reachable sets are generated based on the original reverse reachable sets to double the effect. Therefore, the time efficiency of the algorithm is greatly improved.

In short, this section proposes a method to determine the critical value θ of a random RR set based on the monotonicity and sub-modularity of the influence propagation function, according to the real-time propagation of nodes in the network, and follows the reverse reachable set sampling framework to generate θ reverse reachable sets. Next, the D-RIS algorithm calls algorithm 2 in section 2.3.2 to find the set of seed nodes S.

2.3.2 Seed node selection

The D-RIS algorithm uses the maximum coverage method for seed node selection. Algorithm 2 gives the pseudo code at this stage. Given G, k and the number of reverse reachable sets θ. First, insert the θ random RR sets generated in Algorithm 1 into the set R. If $S \cap R_j \neq \varnothing$, the seed set S covers a random RR set R_j, define $Cover_R(S) = \sum_{R_j \in R} min\{|S \cap R_j|, 1\}$. Then define the approximate value of $I(S)$ as $I\ \ (S)\ = I_R(S) = \frac{Cover_R\ \ (S)}{|R|}$. So the specific k iteration process is as follows:

(1) Each time the algorithm greedily selects a node v that covers the most number of nodes in the R set.

(2) Deletes all the nodes v in the R set in reverse Reachable set (that is, the node v in the deleted reverse reachable set has a path that can be reached through the node).

(3) Add the node v to the set S, update the R set and proceed to the next time Iteration.

(4) Selected node set S=k iteration ends.

In the process of using the maximum coverage method to select k node sets, the greedy algorithm is used to repeatedly select the nodes that cover the largest marginal revenue to join the node set S, so the approximate solution of $(1-\frac{1}{e}-\varepsilon)$ can be returned, and the nearly linear time complexity can be obtained .

Algorithm 2 D-RIS algorithm (node selection)

Input: $G = (V, P, E), k,$ number of RR sets

Output: S

```
    $S \leftarrow \emptyset$
    for  $i = 1$ to k do
    $v =$ max_coverage()
    add v to S
    for RR Sets contain v
    remove all RR Sets from R
    return S
```

The D-RIS algorithm mainly includes two stages. In the first stage, n nodes are randomly selected to generate θ reverse reachable sets, among which $\theta = n \times \alpha(\alpha < 1)$, the time complexity is $o(\theta)$. For any randomly selected node v_j, suppose the time complexity of the reverse reachable set generated by propagation simulation based on a certain propagation model is $o(\theta)$. Where EVP is the width of the random RR set (that is, the number of directed edges pointing to the node v_j in the random graph g), the time complexity of the first stage of the D-RIS algorithm is $o(\theta \times \text{EPV})$. The maximum coverage method used in the second stage selects k nodes using greedy thinking, which can get linear time complexity. So the time complexity of the D-RIS algorithm is $o(\theta \times \text{EPV})$. The time complexity of the greedy algorithm $o(kmnr)$, with r represents the number of times Monte Carlo sampling is used, n and mrepresent the total number of nodes and edges in the network G, commonly, n,m,r is very large. In contrast, D-RIS algorithm has better time complexity. Besides, compared with the RIS algorithm that can also achieve linear time complexity, the D-RIS algorithm is more accurate and reasonable in the selection of the number of reverse reachable sets. The experiment also shows that the operating efficiency of the D-RIS algorithm has a better advantage. According to the above analysis, it can be concluded that the D-RIS algorithm is more suitable for large scale social networks.

3. Experiments and results

3.1. Datasets

In order to verify the timeliness of the D-RIS influence maximization algorithm, we uses two real data sets for experiments. As shown in Table 2: The first Slashdot dataset [18] is a data set of friends sharing technology information websites. This site allows users to mark each other as "friends" or "enemies." Of these, 76.7% are "friend" relationships. In order to facilitate the comparison between different algorithms, this paper processed this data set and kept the friendship between 10,000 nodes. The number of friends after preprocessing is 36,338. The second Epinions dataset [18] is an online social network based on trust. If there is a directed edge from node to node, the node trusts the node. This paper preserves the trust relationship of 10,000 nodes after preprocessing this dataset. Both of the two datasets used [18] can be downloaded on the Stanford large network dataset website.

Table 2. Datasets information.

No	Dataset	Nodes	Edges
1	Slashdot	77357	516575
2	Epinions	75879	508837

3.2. Experimental results and analysis

The information dissemination model used in the experiment is the independent cascade (IC) model, and the dissemination probability is set to 0.08. The experiment was run 10,000 times in Monte Carlo and averaged to obtain the influence propagation range of the simulated propagation process. In order to verify the rationality and timeliness of the D-RIS algorithm, the comparative experiment algorithms we selected are currently five representative algorithms:

CELF algorithm: An improved algorithm of greedy algorithm, the core idea is basically the same and the efficiency is improved by a hundredfold. Therefore, this paper selects the CELF algorithm as a contrast algorithm with greedy thinking.

HighDegree algorithm: A most classic heuristic algorithm based on node centrality, selecting K nodes with the largest degree value as the seed node set.

LIR algorithm[13]: A heuristic algorithm based on topological structure. This algorithm selects the node with the largest local degree value and sorts it, and then selects the seed node set.

pBmH algorithm[14]: A heuristic algorithm based on topological structure. This algorithm takes into account the influence of nodes by multiple neighbor nodes, and avoids the phenomenon of rich clubs.

RIS algorithm[17]: An algorithm based on reverse reachable set sampling, the algorithm generates a certain theoretical threshold number of reverse reachable sets and then selects the seed node set.

Set up the simulation experiment as follows:

D-RIS algorithm rule verification: This experiment uses the Slashdot dataset to verify and analyze the monotonicity and sub-modality of the influence propagation function in the RIS algorithm, and test this rule on the D-RIS algorithm.

D-RIS algorithm and RIS algorithm comparison experiment verification: In this experiment, the number of reverse reachable sets of different ratios of the RIS algorithm is set on the two datasets of Slashdot and Epinions, which will affect the D-RIS algorithm separately Comparative analysis of force propagation range and running time.

Comparison of D-RIS algorithm with other four classic algorithms: Section 3.2.3 of the experiment compares D-RIS algorithm with CELF algorithm, HighDegree algorithm, LIR algorithm and pBmH algorithm on two different real data sets for influence propagation range and The comparative analysis of running time verifies that the D-RIS algorithm has better timeliness.

3.2.1 D-RIS algorithm rule verification

Set $k = 5$, the RIS algorithm starts to iterate from $\alpha = 0.001$, and double the ratio of the reverse reachable set in each round until three consecutive doublings are invalid or stop.

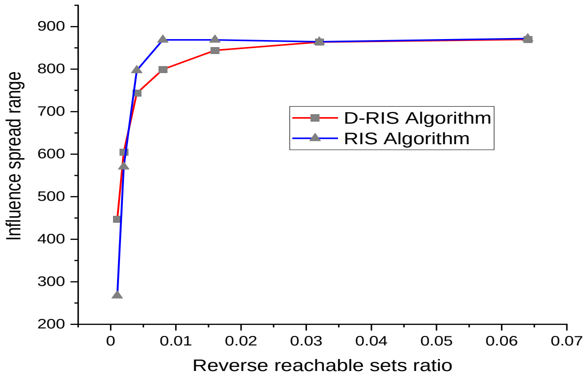


Figure 3. Relation between influence spread range and reverse reachable sets of the RIS algorithm and the D-RIS algorithm on Slashdot.

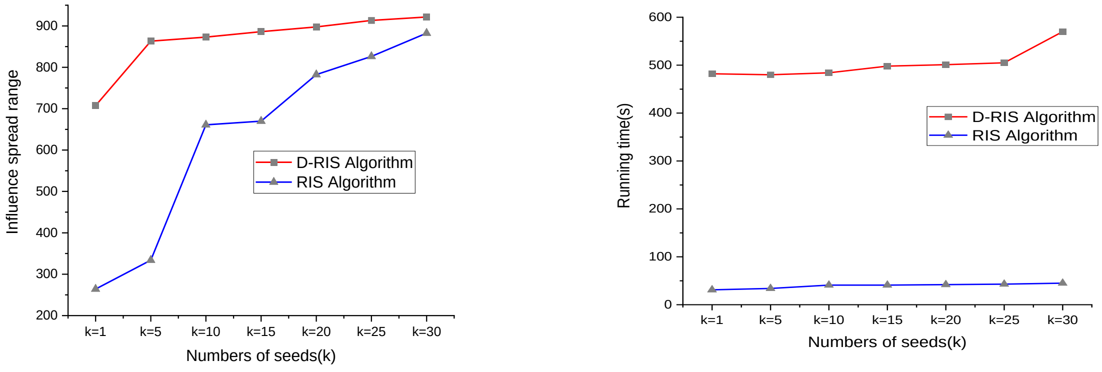
It can be seen from Figure 3 that as the reverse reachable set ratio becomes larger, the front part of the curve shows an upward trend. The spread of influence continues to increase, which shows that the spread of influence of the RIS algorithm and the D-RIS algorithm is monotonic. In the RIS algorithm, when the reverse reachable set ratio is more significant than 0.01, the upward curve with the number of reverse reachable sets tends to be flat. This shows that the influence spreading function has the property of diminishing marginal effects due to the sub-modularity. From the curve in the figure, it can be seen that the expansion of the influence range gradually weakens. When the reverse reachable set ratio is 0.03, the curve's downward trend is slow, which is in line with the actual situation. Theoretically, the influence propagation range of the algorithm is monotonic. Due to the probability model's use as the propagation model, there are inevitable fluctuations in the experiment.

Figure 3 verifies that the basic reverse reachable sets influence propagation function has certain rules based on monotonicity and sub-modularity. With this rule, the RIS algorithm can be improved, which is also the theoretical basis of the D-RIS algorithm proposed in this paper. In the figure, the D-RIS algorithm is also verified on the real data set, and the result shows that the upward trend of the curve increases with the increase of the number of reverse reachable sets and then becomes flat. And the D-RIS algorithm only needs to preset a smaller reverse reachable set ratio, and it can automatically double the debugging ratio until the condition is met. It avoids the problem that the unreasonable selection of the reverse reachable set ratio in the RIS algorithm leads to the failure of the optimal propagation range or the wasted time. This experiment shows that the D-RIS algorithm has a certain rationality and practical significance.

3.2.2 D-RIS algorithm and RIS algorithm comparison experiment verification

Set the reverse reachable set ratio of the RIS algorithm to 0.001, 0.2, 0.5. Compare the influence spread range and running time with the D-RIS algorithm on two different data sets. Figures 4 to 9 are the comparative experimental results of the two algorithms on two different data sets.

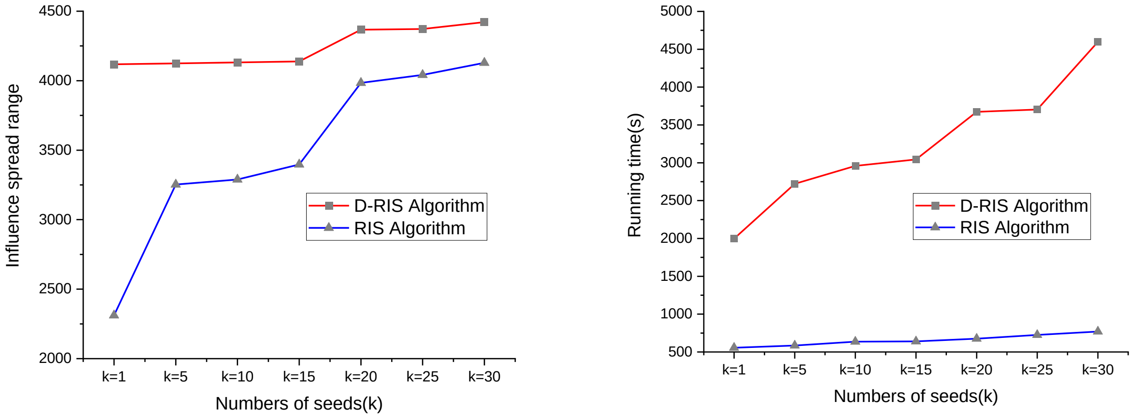
(1) Set the reverse reachable set ratio of the RIS algorithm to 0.001:



(a) Comparisons of influence spread range

(b) Comparisons of running time

Figure 4. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.001) and the D-RIS algorithm on the Slashdot.



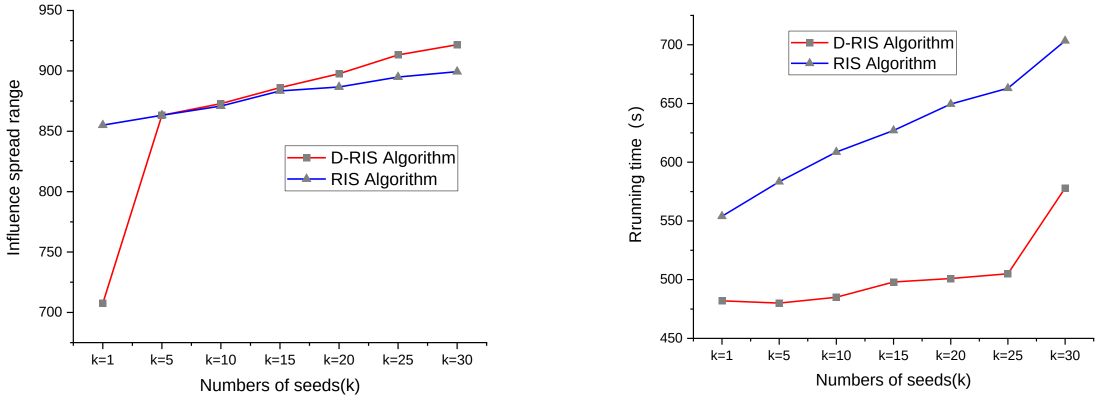
(a) Comparisons of influence spread range

(b) Comparisons of running time

Figure 5. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.001) and the D-RIS algorithm on the Epinions.

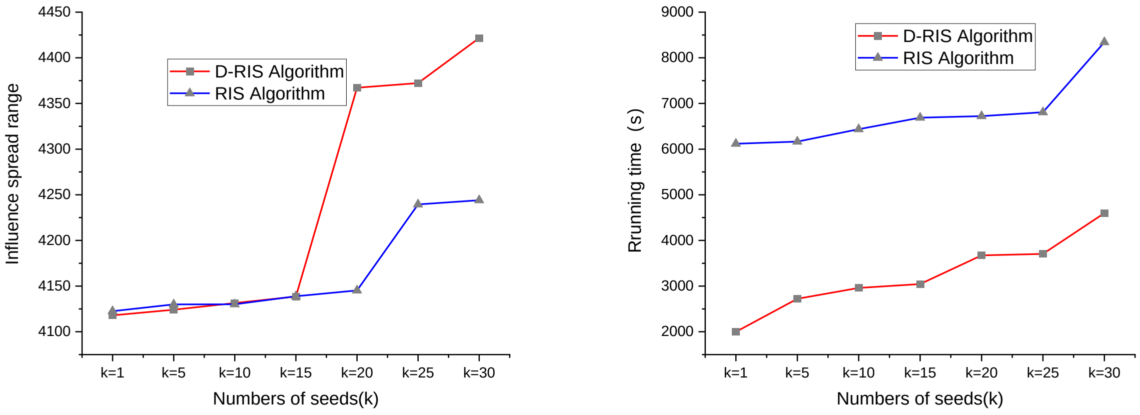
When the RIS algorithm's reverse reachable set ratio is 0.001 (Figure 4, Figure 5), the RIS algorithm runs fast, but the influence spread is smaller than the D-RIS algorithm. Especially when the k value is low, there is a doubled gap in the spread of influence between the two. This is because the threshold of the number of reverse reachable sets in the RIS algorithm is too small, which results in the insufficient number of seed nodes selected, which affects the final propagation range of the algorithm.

(2) Set the reverse reachable set ratio of the RIS algorithm to 0.2:



(a) Comparisons of influence spread range (b) Comparisons of running time

Figure 6. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.2) and the D-RIS algorithm on the Slashdot.

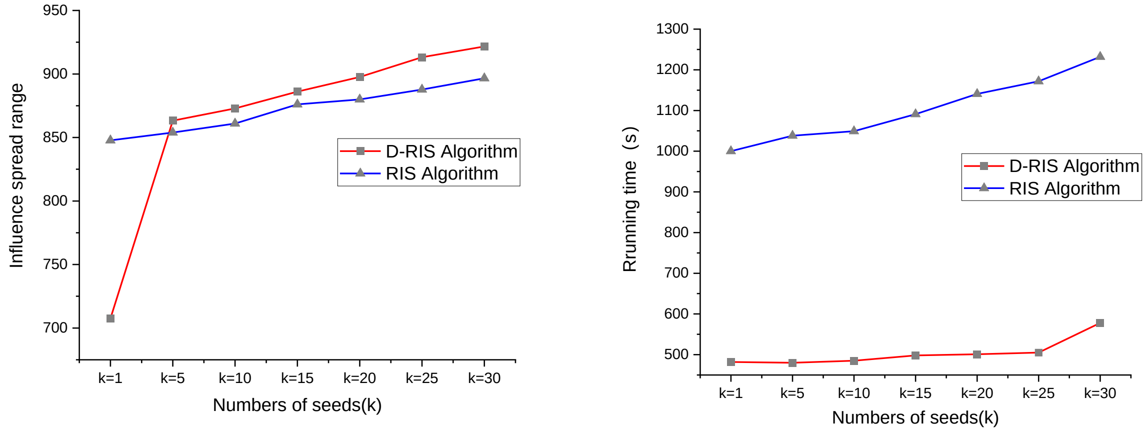


(a) Comparisons of influence spread range (b) Comparisons of running time

Figure 7. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.2) and the D-RIS algorithm on the Epinions.

As shown in Figure 6 and Figure 7, when the reverse reachable set ratio of the RIS algorithm is 0.2. In the Slashdot data set, the influence spread of the two algorithms is close, but the time efficiency of the D-RIS algorithm is higher than that of the RIS algorithm. In the Epinions data set, the D-RIS algorithm greatly improves the running time under the premise of obtaining a larger influence spread range, and the larger the selected seed node set, the more obvious the advantage.

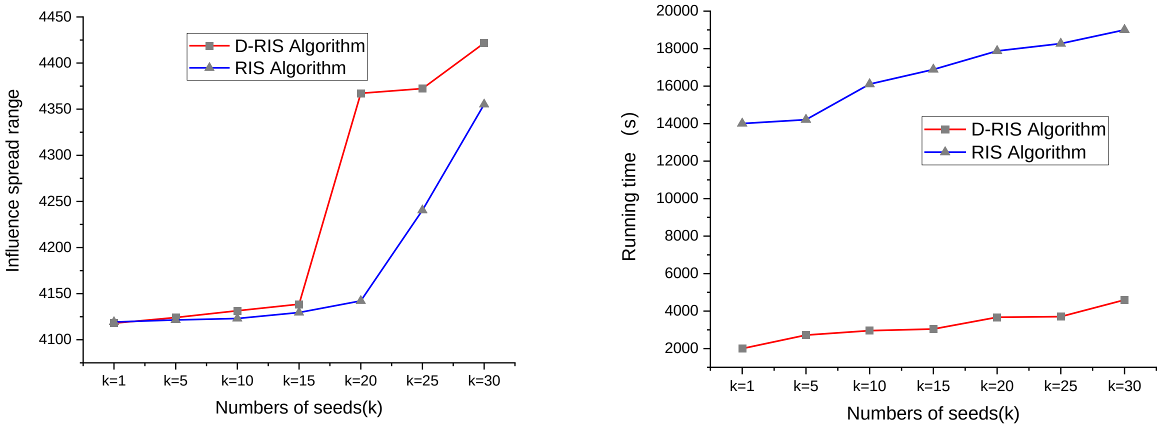
(3) Set the reverse reachable set ratio of the RIS algorithm to 0.5:



(a) Comparisons of influence spread range

(b) Comparisons of running time

Figure 8. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.5) and the D-RIS algorithm on the Slashdot.



(a) Comparisons of influence spread range

(b) Comparisons of running time

Figure 9. Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.5) and the D-RIS algorithm on the Epinions.

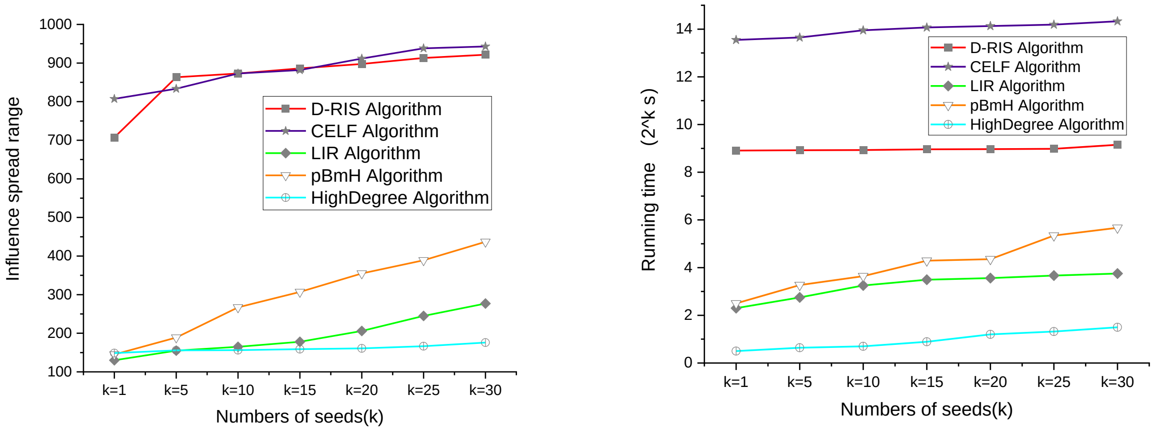
As shown in Figure 8 and Figure 9, when the RIS algorithm sets the reverse reachable set ratio to 0.5.On the two data sets, the D-RIS algorithm has a better spread range of influence, and the operating efficiency is much higher than that of the RIS algorithm. It can be seen that a too large reverse reachable set ratio will result in a waste of the final time cost of the algorithm. For the Slashdot data set, the running time of the RIS algorithm is more than twice that of the D-RIS algorithm. For the Epinions data set, the running time of the RIS algorithm is more than 7 times that of the D-RIS algorithm. Therefore, the D-RIS algorithm in this article is in the running time. The advantages are more obvious.

In summary, through experimental verification on two real data sets, it can be seen that when the theoretical threshold of the reverse reachable set of the RIS algorithm is set too small, the influence propagation range is small. When the theoretical threshold of the reverse reachable set is too large, the time efficiency of the RIS algorithm is too poor. The D-RIS algorithm can achieve a better influence spreading range and at the same time run more efficiently.

In addition, compared with the RIS algorithm, the D-RIS algorithm avoids the inaccurate setting of the theoretical threshold of the number of reverse reachable sets, which leads to the problem of not reaching the optimal influence propagation range or causing a large waste of time. For the current complex social networks, the D-RIS algorithm does not require repeated calculations, and the algorithm automatically debugs to generate a certain ratio of reverse reachable set is also more suitable for subsequent network structure changes. Therefore, the D-RIS algorithm has certain practical significance.

3.2.3 Comparison of D-RIS algorithm with other four classic algorithms.

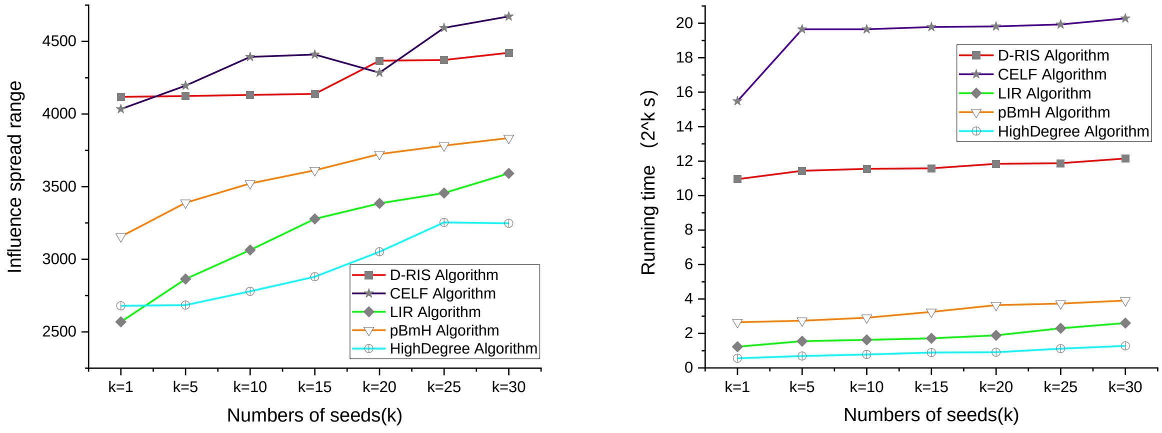
On two different data sets, the D-RIS algorithm is compared with the heuristic HighDegree algorithm, LIR algorithm and pBmH algorithm and the greedy-based CELF algorithm to compare the influence propagation range and running time.



(a) Comparisons of influence spread range

(b) Comparisons of running time, the result of the ordinate is the result of taking the logarithm of 2

Figure 10. Comparisons of the running time of the five algorithms on Slashdot.



(a) Comparisons of influence spread range

(b) Comparisons of running time, the result of the ordinate is the result of taking the logarithm of 2

Figure 11. Comparisons of the running time of the five algorithms on Epinions.

According to the analysis of the experimental results in Figure 10 (Slashdot data set) and Figure 11 (Epinions data set):

(a) The influence propagation range of the D-RIS algorithm is basically similar to the CELF algorithm which is close to the optimal solution within the $(1-\frac{1}{e}-\varepsilon)$ range. But D-RIS runs faster, and the larger the seed node set, the more obvious the advantage, the difference is close to hundreds of times, this is because the CELF algorithm uses the Monte Carlo method for calculations, resulting in extremely high time complexity, so D-RIS The algorithm is more suitable for large-scale social networks.

(b) Compared with heuristic algorithms (HighDegree algorithm, LIR algorithm and pBmH algorithm), although the D-RIS algorithm performs poorly in terms of running speed, the spread of the algorithm's influence is much higher than these heuristic algorithms. In the Epinions data set, the influence spread of the heuristic algorithm is only about 50% of that of the D-RIS algorithm. In the Slashdot data set, the D-RIS algorithm has more obvious advantages in spreading influence. It can be seen that although the heuristic algorithm has extremely high operating efficiency, it does not take into account that the complex network follow-up structure results in the selection of seed nodes that are not accurate enough, and the spread of influence is small, and the optimal solution is not reached. In addition, the stability of the heuristic algorithm is not good in different data sets.

Based on the comparative experimental analysis of the above algorithms, it can be seen that the D-RIS algorithm proposed in this paper has achieved a good balance between the influence spread range and time efficiency, and has shown good versatility and stability. More suitable for large-scale social networks.

4. Conclusions

In this paper, we proposes a D-RIS influence maximization algorithm based on the independent cascade model combined with the reverse reachable set. Compared with the traditional RIS algorithm, the above algorithm obtains the number of reverse reachable sets by setting the automatic tuning threshold instead of the fixed threshold. The experimental results show that D-RIS algorithm is close to CELF algorithm and higher than RIS algorithm, HighDegree algorithm, LIR algorithm and pBmH algorithm in the spread of influence, and it is significantly better than CELF algorithm and RIS algorithm in running time. Therefore, the D-RIS algorithm proposed in this paper has dual advantages in terms of time efficiency and influence spread and can be applied to structural changes and large-scale social networks. In the following research, we will focus on extending the D-RIS algorithm to a more realistic multi-relationship influence propagation model.

Author Contributions: Conceptualization, Gengxin Sun; Data curation, Gengxin Sun; Formal analysis, Gengxin Sun; Funding acquisition, Gengxin Sun; Investigation, Gengxin Sun; Methodology, Gengxin Sun; Project administration, Chih-Cheng Chen; Resources, Chih-Cheng Chen; Software, Chih-Cheng Chen; Supervision, Chih-Cheng Chen; Validation, Chih-Cheng Chen; Visualization, Chih-Cheng Chen; Writing – original draft, Chih-Cheng Chen; Writing – review & editing, Chih-Cheng Chen.

Funding: This research was funded by Shandong Provincial Natural Science Foundation, China, grant number ZR2017MG011.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing [C]// Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, 2002: 61-70.
2. Domingos P, Richardson M. Mining the network value of customers [C]// Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, 2001: 57-66.
3. Kempe D, Kleinberg J. Maximizing the spread of influence through a social network [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003:137-146.
4. Goldenberg J, Libai B, Muller E. Talk of the network: A complex systems look at the underlying process of word-of-mouth[J]. Marketing Letters,2001,12(3):211-223.
5. Goldenberg J, Libai B, Muller E. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata[J]. Academy of Marketing Science Review,2011: 9(3):1–18.
6. Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2007:420-429.
7. Goyal A, Lu W, Lakshmanan L V S. CELF++: optimizing the greedy algorithm for influence maximization in social networks [C]// International Conference Companion on World Wide Web. ACM, 2011:47-48.
8. Li Minjia, Xu Guoyan, Zhu Shuai, Zhang Wangjuan. Influence maximization algorithm based on structure hole and degree discount [J]. Journal of Compurter Appilications, 2018,38(12):3419-3424.
9. Chen W, Wang Y, Yang S. Efficient influence maximization in social networks [C]// Proc of the 15th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York:ACM,2009:199-208.
10. Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral markerting in large-scale social networks [C]// Proc of the 16th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM,2010: 1029-1038.
11. Jung K, Heo W, Chen W. IRIE: Scalable and robust influence maximization in social networks [C] // Proc of the 12th IEEE Int Conf Data Mining (ICDM). Piscataway, NJ: IEEE, 2012: 918-923.
12. Wang Z, Wang H, Liu Q, et al. Influence nodes selection: A data reconstruction perspective [C] // Proc of the 37th Int ACM SIGIR Conf On Research & Development in Information Retrieval. New York: ACM, 2014:879-882.
13. Dong Liu, Yun Jing, Jing Zhao, et al. A fast and efficient algorithm for mining Top-k nodes in complex natworks [J]. 2017,7(1): 5-691.
14. Dut-Linh Nguyen. Tri-Hai Nguyen, Trong-Hop Do, et al. Probability-based muti-hop diffusion method for influence maximization in social networks [J]. 2017,93(4):903-916.
15. Xie Shengnan, Liu Yong, Zhu Jinghua, et al. Research on topic-based local influence maximizing algorithm in social network [J]. Journal of Frontiers of Computer Science&Technology, 2016, 10(5):646-656.
16. Cao Jiuxin, Dong Dan, Xu Shun, et al. Self-Interest influence maximization algorithm based on subject preference in competitive environment [J]. Chinese Journal of Computers, 2015, (02) :238-248Li Songyang. Research on Node Detection and Information Propagetion in Online Social Network[D]. Chongqing University of Posts and Telecommunications, 2017.

17. Borgs C, Brautbar M, Chayes J, et al. Maximizing social influence in nearly opyimal time [C]// SIAM.Proceeding of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms.[S.1.]:SIAM,2-14:946-957.
18. May RM, LIoyd Al. Infection dynamics on scale free networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2001, 64(2):066112.
19. Hethcote H W. The mathematics of infectious diseases [J]. Siam Review, 2000, 42(4):599-653.
20. Li Songyang. Research on node detection and information propagetion in online social network[D]. Chongqing University of Posts and Telecommunications, 2017.
21. Zhao Yueai, Feng Liping. Factors Affecting User behavior on virus transmission in social networks[J]. Journal of Taiyuan University of Technology, 2018, 49(4): 61-66.
22. Leskovec J, Krevl A. SNAP[EB/OL]. 2016[2016-03-10]. http://snap.stanford.edu/data.