ARTICLE IN PRESS
JID: INS [m3Gsc;November 8, 2019;15:57]
Information Sciences xxx (xxxx) xxx
Contents lists available at ScienceDirect
Information Sciences
journal homepage: www.elsevier.com/locate/ins

A new algorithm for positive influence maximization in signed networks

Weijia Ju[a], Ling Chen[a,b,*], Bin Li[a], Wei Liu[a], Jun Sheng[a], Yuwei Wang[a]

[a] College of Information Engineering of Yangzhou University, Yangzhou 225127, China
[b] State Key Lab of Novel Software Tech, Nanjing University, Nanjing, 210093, China

ARTICLE INFO

Article history:
Received 5 March 2019
Revised 8 October 2019
Accepted 25 October 2019
Available online xxx

Keywords:
Signed network
Positive influence maximization
Independent cascade model
Negative influence

ABSTRACT

With the rapid development of online social networks, the problem of influence maximization (IM) has attracted much attention from researchers and has been applied in many areas such as marketing and finance. Since positive and negative relations may exist between individuals in social networks, the problem of influence maximization in signed networks has a wide range of applications. This paper presents an efficient algorithm for positive influence maximization in signed networks in the independent cascade model. First, we propose an independent path-based algorithm to compute the activation probabilities between the node pairs. Based on the activation probability, we define a propagation increment function to avoid simulating the influence spreading for selecting candidate seed nodes. We present an algorithm to select the seed nodes to obtain the largest positive influence spreading in the signed network. Empirical results in social networks show that our algorithm can have wider positive influence spreading than other methods.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Recently, with the booming development of social networks, such as Twitter, Facebook, Google+, and China Weibo, network analyses have become a hotspot of research, such as information diffusion [6], community detection [10], direction recovery [35], metric learning [20], network structure analysis [39] and link prediction [1]. Social networks have had a great impact on information propagation and become a good platform for people to exchange opinions and to propagate information. For example, a merchant of Amazon may promote a new product through a trade network by providing discounts to some influential customers in the network to maximize the influence of the product. The goal of the merchant is to use those influential customers to influence as many direct and indirect friends as possible. After this cascade propagation process, a large number of customers begin to adopt and buy this product. This phenomenon is called the word-of-mouth effect. After the total discount is determined by the merchant according to his budget, the problem concerns how to select the initial influential customers to offer the discount to maximize the expected spreading influence. Similar problems may occur in many areas, such as bioinformatics [29], political elections, viral marketing [27,30], economics [32], recommendations [4], and public opinion monitors [42], among others.

The problem mentioned above is called influence maximization (IM) in the literature, which has aroused researchers' extensive interest. Domingos et al. [6] defined the IM problem and provided a model for it in 2001.In the model, a graph

* Corresponding author.
E-mail address: lchen@yzu.edu.cn (L. Chen).

https://doi.org/10.1016/j.ins.2019.10.061
0020-0255/© 2019 Elsevier Inc. All rights reserved.

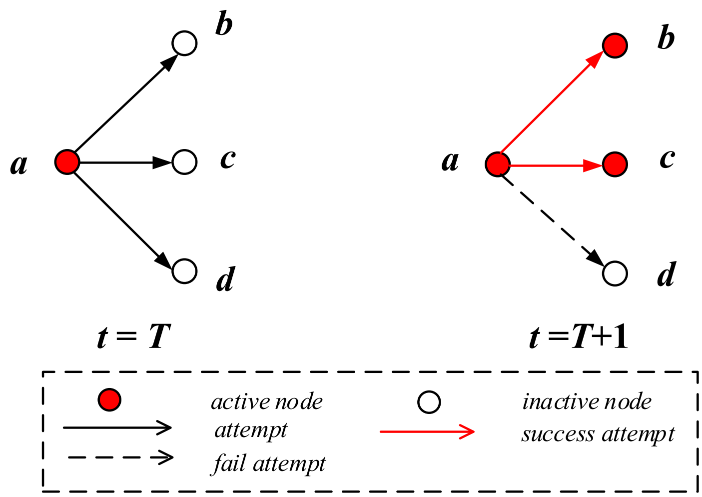

Fig. 1. The independent cascade (IC) model.

$G=(V, E)$ is used to represent an online social network, where V and E respectively denote the sets of nodes and links. A probability function p: $E \rightarrow (0,1)$ is defined to assign the propagation probability on each link in E. Given an integer $k(k \leq |V|)$, the goal of influence maximization is to select k nodes that can propagate the influence to the maximum number of nodes. We call those k nodes the seeds of influence propagation. We use $I_M(S)$ to denote the number of nodes that are directly or indirectly influenced by the seed set S in the propagation model M. The optimal seed set S^* is defined as

$$S^* = \mathop{\arg\max}_{S\subset V, |S|=k} I_M(S) \tag{1}$$

At each time step of influence propagation, a node remains in one of the two states as follows: if a node has not been influenced, then it is in the inactive state; after it is influenced, it is in the active state. Initially, the seed nodes are in the active state and the other nodes are in the inactive state. In each propagation step, a node u in the active state tries to activate its inactive neighbor v. After being activated, neighbor v attempts to activate its inactive neighbor w. This procedure is repeated until no more nodes can be activated. Kempe et al. [13] defined two common models for influence propagation: the linear threshold (LT) model and the independent cascade (IC) model.

We investigate the positive influence maximization in signed networks under the IC model. In the IC model, all the seed nodes in S are initially in the activated state, and influence spreading begins from those seed nodes. For instance, in the network shown in Fig. 1, suppose the seed set S is $\{a\}$. Initially, node a is active and tries to activate its inactivated neighbor nodes b, c and d. An activated node has only one opportunity to activate each of its inactivated neighbors. Whether b is activated by a will not be affected by the activations of other neighbors c or d. In Fig. 1, b and c are successfully activated, while d is not. After activation of node b and c, they attempt to activate their inactive neighbors in the next time step. This process of activation is iterated until no more nodes are activated.

Kempe et al. [13] showed influence maximization is NP-hard in LT and IC models and advanced a greedy algorithm to obtain a sub-optimal solution for the problem. Nevertheless, the greedy algorithm consumes a huge amount of computation time for large graphs because it needs to simulate the influence spreading process for estimating the influence increment of each candidate seed node. Due to such a time-consuming simulation, it is impractical to apply the greedy algorithm to real-world social networks. Recently, many works have been conducted to reduce the time complexity of Kempe's greedy algorithm on the premise of its guaranteed accuracy.

Most of the existing research on influence maximization has only considered networks with positive relationships, such as collaborators, trust or friends. Nevertheless, in real-world applications, apart from positive relationships, there also exist negative relationships, such as hostile relationships, distrust or foes. Due to such negative relations, different opinions always exist in social and trade networks. Such disagreement leads to two opposite influences in the network: positive and negative influences of the retailers, merchants or politicians. Therefore, one must eliminate the negative influence of his opponents when attempting to maximize his own influence. For example, two companies, C1 and C2, have a competitive relationship. If a customer A prefers the product of C1 and has a positive relation with customer B, then B could probably be positively influenced by A and prefers the product of C1. If customer A has a negative relation with customer B, then B probably dislikes the product of C1 by the negative influence from A and prefers the product of C2. Thus, customers in the social network can be implicitly partitioned into two groups: those who prefer the products of C1 and those who prefer the products of C2. When C1 wants to market his products on social networks, he tries to avoid the members in C2 group who could probably spread a negative influence for the product of C1. Additionally, C1 might initially activate several positive influential customers by offering them discounts so that they can propagate the positive influence to the maximum range and eliminate the negative influence from its opponent C2.This is the problem of competitive influence maximization in signed networks. A detailed description of the problem and related works are provided in Section 2. In recent years, many models of competitive influence diffusion and approaches for maximizing such diffusion in signed networks have been proposed [18,19,21,22,28]. The efficiency and accuracy of the algorithms for positive influence maximization have been greatly improved.

However, in the existing works mentioned above, two factors prevent high-quality results for the PIM-SN problem. First, very few researchers have focused on the effect of negative edges on competitive influence propagation. In a signed network, when influence is propagated through a negative edge, helpful influence will become harmful, and vice versa, making the negative edges have larger impact on the results of influence maximization than the positive ones. Such a critical role of negative edges is not considered and utilized by most existing methods. Second, most of the existing methods for positive influence maximization use Monte-Carlo simulation to estimate the influence spreading of every candidate seed set S. However, under the IC model, evaluating the influence spreading of a given seed set is a #P-hard problem. Such a time-consuming simulation process hinders the application of those methods in solving real-world problems.

Motivated by this background, we propose a set of propagation rules to model the competitive influence spreading. In the model, negative links play a more critical role than positive ones, since they can reverse the influence. This phenomenon is consistent with the competitive influence spreading in real world applications. Such a critical role of the negative edges is not considered and utilized by most existing methods. To avoid the Monte-Carlo simulation in estimating the influence propagation range and to reduce the computation time, we present a propagation function to estimate the positive influence spreading of the seed set. We propose a propagation path-based algorithm to compute the activation probability between each node pair. Using such an activation probability, we propose an algorithm to sequentially select the nodes that can maximize the influence increment to join the seed set.

The major innovations and contributions of our work are as follows:

1. We formally define the problem of the positive influence maximization problem in signed networks under the IC model. We also propose a set of propagation rules to model the competitive influence spreading.
2. We present an algorithm to compute the positive activation probability between each node pair using the independent paths.
3. To avoid the time-consuming process of simulating the influence propagation, we define a propagation function to estimate the influenced range of a seed set.
4. An algorithm named PIMSN (positive influence maximization in signed networks) is presented for detecting the optimal seed set using the propagation function.

The rest of this paper is structured as follows. Section 2 provides a brief survey of related works. Section 3 defines the problem of positive influence maximization in signed networks. An independent path-based algorithm is presented to compute the activation probability between each node pair. In Section 4, we define a propagation function to approximate the influence propagation of a candidate seed set. In Section 5, we present an algorithm, PIMSN, for positive influence maximization in signed networks under the IC model. Section 6 shows and analyzes the experimental results of algorithm PIMSN. Section 7 presents conclusions and our further research.

2. Related work

After Domingos et al. [6] defined the influence maximization problem, various propagation models of influence spreading in networks have been advanced [7,31]. Kempe et al. [13] defined two classical spreading models, i.e., the linear threshold model and the independent cascade model. Kimura et al. [14] defined a cascade model taking into consideration influence spreading through the shortest path. Kempe et al. [12] proposed the decreasing cascade model considering the damping phenomenon of influence spreading. Kempe et al. [13] showed that maximizing influence in Eq. (1) and obtaining the optimal seed set S^* is NP-hard in the LT and IC models. Using the greedy method, Kimura et al. [14] advanced an $(1-1/e)$-approximate algorithm for the optimization in Eq. (1). Based on the two classical models, some new problems of influence maximization are proposed and studied.

In some social and trade networks, different opinions always exist, and people may have different viewpoints about the same issue. Such disagreement leads to competition among retailers, merchants and politicians. Therefore, there probably exist two opposite influences in the network: positive and negative influences. Our goal is to propagate the positive influence to as many users as possible and eliminate the negative influence from our competitors. This is the problem of IM in competitive social networks. Recently, many researchers have investigated this issue. Feng Wang et al. [33] formulated the maximizing positive influenced users (MPIU) problem using fluid dynamics theory to reveal the time-evolving influence spread process, and they designed the Fluidspread greedy algorithm to solve it. Zhang et al. [40] proposed an opinion-based cascading model for maximizing the total positive influence of activated users taking the users' opinions into account. They presented a greedy method for the problem of MIO (maximizing influenced users' opinions) in the opinion-based cascading model. Siwar Jendoubi et al. [11] presented an evidential-based method for positive opinion influence maximization. They detected influential users holding positive attitudes and used the belief function to avoid information deficiency. Zhang et al. [41] studied the issue of minimum partial positive influence seeding, namely, finding a minimum seed set that can propagate the influence to a certain range. Yang et al. [38] investigated the relative influence maximization (RIM) problem

Table 1
Main notations used in the paper.

Notation	Description
$G = (V,E)$	A network with node set V and edge set E
$sign(u,v)$	Sign on edge (u,v)
$p(u,v)$	Probability of propagation on edge (u,v)
$I^+(S)$	Number of the nodes positively influenced by seed set S
$P_r(L)$	Propagation probability of path L
$a^+(v,u)$	Probability for v positively influencing u
$a^-(v,u)$	Probability for v negatively influencing u
$G_W = (W,E_W)$	A sub-graph of G, $W \subset V$, E_W is the set of edges linking the nodes in W
$P^+(G_W, v, u)$	Set of independent positive paths from v to u in G_W
$G+(S,u)$	Probability for seeds S to positively activate node u
$G-(S,u)$	Probability for seeds S to negatively activate node u
$a_W{}^+(v,u)$	Probability for v positively influencing u in G_W
$a_W{}^-(v,u)$	Probability for v negatively influencing u in G_W
$\Delta_S{}^+(x)$	Positive propagation increment for adding x into S

and designed efficient algorithms to achieve the maximum relative influence, namely, the spread of positive opinions while reducing the spread of negative opinions. Dong Li et al. [19] presented a simulated annealing-based method for the positive influence maximization problem. Additionally, they also proposed two heuristics approaches to achieve higher quality results. Mohammad Mehdi Daliriet al. [9] presented a learning automaton-based algorithm for the problem of identifying a small subset of target nodes for maximizing the spread of positive influence. Arastoo Bozorgi et al. [2] proposed a community-based method to solve the competitive influence maximization problem in an extended linear threshold model.

In the real-world social and trade networks, there probably exist two opposite relations, namely, positive and negative relationships. In those signed networks, each link has a sign that is either positive or negative. Recent works have described the IM problem in signed networks. Shen et al. [28] advanced a new spreading model named LT-S in signed networks based on the traditional LT model integrating opinion information and signed relationships. They proposed a modified greedy algorithm named RLP to solve the problem. Some models [8,36] are proposed to incorporate both positive and negative relationships. He et al. [8] studied the problem of positive opinion influential node set (POINS) detection and presented a greedy algorithm POINS-GREEDY to solve POINS efficiently. Weng et al. [36] proposed an algorithm HPG-N that used cumulative features incorporating both positive and negative relationships. Maryam Hosseini-Pozveh [18] studied the influence diffusion models for signed social networks. They proposed two classes of models: cascade-based and threshold-based. Liu et al. [23] addressed the problem of influence maximization in the signed network under the independent cascade diffusion model. They proposed an independent path-based greedy algorithm to maximize the positive influence spreading in the signed network.

In research examining the influence maximization problem in signed networks, very few researchers have focused on the effect of negative edges on influence propagation. In a signed network, when influence is propagated through a negative edge, helpful influence will become harmful, and vice versa. However, existing methods cannot effectively use this feature of the signed network to propagate a helpful influence to the maximal range. In this work, we define this problem as positive influence maximization in signed networks and study how to effectively utilize the information of signed relations for detecting seed nodes to maximize the scope of the positive influence and to eliminate the negative influence. In most influence maximization algorithms for signed networks, the traditional greedy methods are used. In the greedy method, the positive influence propagation range is estimated by Monte-Carlo simulations for all candidate seed sets in each iteration. Such a time-consuming process of simulation prevents the application of greedy methods to real-world problems. To avoid such a time-consuming process of simulation, we present a propagation function to estimate the positive influence spreading of a seed set. In addition, we propose an efficient algorithm to detect the most influential nodes in the signed network to maximize the positive influence.

3. PIM-SN problem and independent path

In this section, we first formally define the problem of positive influence maximization in signed networks (PIM-SN). Then, we present an algorithm named *Comp-pp* to calculate the activating probabilities of the nodes, which will be used in Section 4 to compute the propagation function. The main notations used in this paper are listed in Table 1.

3.1. Problem formulation

In a signed network, the nodes may connected by positive or negative links. A positive link indicates the nodes have a positive relationship, such as a relative, friend or classmate, and a negative link indicates a negative relationship, such as a foe or competitor. In a competitive market, a customer may buy a product according to the suggestion of his friend and reject a product when it is advertised by his foe. The disseminator wants his influence to reach a large portion of the population and limit the influence from his competitor. Therefore, we define two activated states for each active node:

positively and negatively activated states indicating the node has been positively or negatively influenced. Our goal is to positively activate as many nodes as possible using the positive and negative relations between the nodes. We define such a problem as positive influence maximization in signed networks:

Definition 1 (Positive influence maximization in signed networks, PIM-SN)**.** A signed online social network can be represented by a graph $G = (V, E)$, where E denotes the set of directed links, and V denotes the set of nodes. A probability function p: $E\rightarrow(0,1)$ is defined as the propagation probability on each link in E. On each link $(u, v) \in E$, there is a positive or negative sign denoted as *sign*(u,v): *sign*(u,v)=1indicates the sign on edge (u,v) is positive, and *sign*$(u,v)=-1$indicates a negative sign on edge (u,v). Let $I^{+}(S)$ be the number of nodes being positively activated by seed set S following the signed influence propagation (SIP) rule. Positive influence maximization in the signed network is performed to achieve a seed set S^{*} such that

$$S^{*} = \underset{S\subset V, |S|=k}{\arg\max}\, I^{+}(S) \tag{2}$$

Here, the SIP (signed influence propagation) rule is defined as follows:

Definition 2 (Signed Influence Propagation Rule, SIP)**.** In the influence propagation on a signed network $G=(V,E)$, when node v is in a positively (or negatively) activated state, if edge (v,u) is positive, then node u will be activated to a positive (or negative) state with probability $p(v,u)$, or remain in an inactive state with probability 1- $p(v,u)$; if edge (v,u) is negative, node u will be activated to a negative (or positive) state with probability $p(v,u)$, or remain in an inactive state with probability 1- $p(v,u)$.

From the definition, we can see that in such a signed network, when helpful influence is propagated through a negative edge, it will become harmful. Additionally, harmful influence will become helpful after being propagated through a negative edge. It is important to take advantage of such negative relations to effectively select seeds that can positively activate the largest number of nodes.

3.2. The path propagation probability

There are two commonly used influence propagation models: linear threshold (LT) model and independent cascade (IC) model. We investigate the positive influence maximization in signed networks under the IC model. Let u be a node that is indirectly activated by seed v. The process of activation under the IC spreading model can be represented by a path L from v to u. The probability of seed v activating u through L can be estimated by the probability of path L defined as follows:

Definition 3 (Path Probability)**.** Suppose a path L from v to u consists of a sequence of edges e_1, e_2, ..., e_l, and p_i is the probability on edge e_i.The probability of path $L = (e_1, e_2, ..., e_l)$ from v to u can be estimated as $P_r(L) = \prod_{i=1}^{l} p_i$.

Since multiple paths may exist from v to u, the probability of seed v activating u can be calculated in independent paths.

Definition 4 (Independent Paths)**.** If two paths L_1 and L_2 do not overlap each other, namely, they have no common edges, we call them mutually independent paths.

The probability of seed v positively activating u is denoted as $a^{+}(v,u)$, and the probability of seed v negatively activating u is denoted as $a^{-}(v,u)$. To estimate $a^{+}(v,u)$ and $a^{-}(v,u)$, we need to find a set of independent paths from v to u and add up their probabilities. Since there are many sets of independent paths from v to u, we propose an algorithm to obtain a set of independent paths with the largest probabilities. Since the paths with loops are meaningless for influence propagation, we only consider the paths without repeated nodes. Based on the above discussion, the algorithm for calculating propagation probabilities between nodes is summarized in Fig. 2.

The depth-first search (DFS) method is used in Algorithm 1 to detect the paths. The DFS method in the algorithm is different from the classical depth-first search in two aspects. First, since each edge only appears in only one path in the set, the search result is a set of independent paths instead of a depth-first tree. Second, when the search reaches a node w for which all connected edges have been visited, the search will return to the starting point v, unlike in the traditional depth-first search that backtracks to visit the edges, leaving the node from which w is discovered.

Algorithm 1 searches for independent paths starting from each node v in the network by a depth-first search. At each node w in the depth-first search, the algorithm selects a neighbor of w with the largest propagation probability as the next node in the path. To ensure the constructed paths are independent, each node can join only one path. Therefore, only the unvisited nodes can be selected and added to the new path. When node u is selected, the probability of path L from v to u is added to $a^{+}(v,u)$ or $a^{-}(v,u)$, depending on the sign of path L. We call path L from v to u a positive (negative) one if there are even (odd) number of negative edges on the path. Let the probability of path L be $P_r(L)$. If path L is a positive one, then $P_r(L)$ should be added to $a^{+}(v,u)$; otherwise it should be added to $a^{-}(v,u)$.

We take the network shown in Fig. 3 as an example to illustrate the execution process of algorithm 1. Suppose all the edges in the network are positive, and the number on each directed edge (u,v) represents the probability for node u to activate v. To obtain the independent paths starting from node a, the algorithm first chooses edge (a,d), which has the largest probability. Then, from node d, the edge with the largest probability (d,g) is chosen. Since no out-edge can be chosen from node g, the first path $a\rightarrow d\rightarrow g$ ends at g. In addition to path $a\rightarrow d\rightarrow g$, two other independent paths, $a\rightarrow c\rightarrow e$ and $a\rightarrow b\rightarrow f$, can be similarly detected. The probabilities of these paths are 0.45, 0.72 and 0.24, respectively.

Algorithm1 *Comp-PP* (Computing Propagation Probabilities)
Input: $G=(V,E,P)$: the network;
Output: a^+, a^-: the matrixes of positive and negative influence propagation probability;
Begin
  For all $v, u \in V$ **do**
    $a^+(v, u)=0$; $a^-(v, u)=0$;
  endfor;
  For all nodes $v \in V$ **do**
    For all edges $(x,y) \in E$ **do**
      $visit(x,y)=0$
    endfor;
    Repeat
      $node(L)=\{v\}$;$P_r(L)=1$;$sign(L)=1$;
      /* L is a path starting from v, $node(L)$ is the set of nodes on path L*/
      /*$P_r(L)$is the probability of path L, $sign(L)$is the sign of path L*/
      /*$sign(L)=1$indicatespath L *is* a positive one, $sign(L)=-1$indicates$L$is negative */
      $x = \underset{x \in N(v), visit(v,x)=0}{\arg\max}\ p(v,x)$;$w=v$;
      /*Find an unvisited neighbor of v with the largest propagation probability; */
      /* $N(v)$ is the set of neighbors of v*; */
      /*visited(v,x)=0 indicates edge (v,x)has not been visited in the search*/
      While $x \notin node(L)$ **do**
        $P_r(L)=P_r(L) * p(w,x)$;$sign(L)=sign(L)*sign(w,x)$;
        if $sign(L)=1$ **then** $a^+(v,x)=a^+(v,x)+P_r(L)$ **else** $a^-(v,x)=a^-(v,x)+P_r(L)$;
        $node(L)=node(L) \cup \{x\}$; $visit(w,x)=1$;
        $w=x$;
        $x = \underset{u \in N(w), visit(w,u)=0}{\arg\max}\ p(w,u)$;
      End while;
    Until all the edges linking v are visited;
  Endfor v
End

Fig. 2. Algorithm Comp-PP.

In addition to DFS, the breadth-first search (BDF) and Dinitz's algorithm [5] for max-flows, which combines BFS with DFS to find multiple shortest paths, can also be similarly modified to efficiently construct the independent paths. To find the max-flow in a network, Dinitz's algorithm [5,34] iteratively uses BFS to assign each node a layer number and then uses DFS to find augmenting paths from source to sink following the layer numbers. Such path detection is very similar to searching for independent paths of influence spreading. The only difference is that in any state of augmenting path for max-flow, the edges are selected according to their remaining capacities; in searching for the independent path for influence spreading,

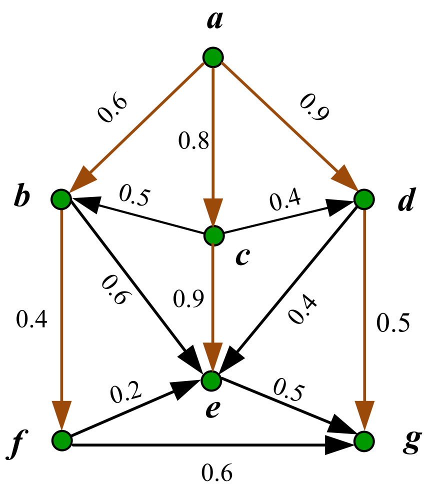

Fig. 3. An example.

the edges are selected according to their probabilities for influence propagation. In fact, detecting the independent paths for propagating the maximum influence can be transformed into the problem of finding the paths for max-flow [37]. The issue of finding the maximum number of nodes reachable from a given node x via independent paths in graph G can be formulated as a max-flow problem in a slightly modified graph G' [37], which can be constructed by adding a few edges on G. In this case, nodes on the paths in this network where the max-flow passes through are the nodes reachable via independent paths from the seed node x.

3.3. Propagation probability in a sub-graph

Let $W \subset V$ be a set of nodes and $G_W = (W, E_W)$ be a sub-graph of G, where $E_W = \{e | e \in E, e = (v, u), v, u \in W\}$ is the subset of edges linking the nodes in W. For the subset $W \subset V$, we use $a_W^+(v, u)$ and $a_W^-(v, u)$ to estimate the probabilities of v positively and negatively activating u in sub-graph G_W, particularly when $W == V$, $a_V^+(v, u)$ and $a_V^-(v, u)$ are just the positive and negative activation probabilities $a^+(v, u)$ and $a^-(v, u)$ in G.

Let V and W be two sets of nodes and v be a node in V. We use V-W, V-x, $W + x$ and $V - S + v$ respectively to denote the node sets $V \backslash W$, $V \backslash \{x\}$, $W \cup \{x\}$ and $\{V \backslash S\} \cup \{v\}$ in the rest of this paper.

Lemma 1. *Let $W \subset V$ be a set of nodes, v and u be two nodes in W, and x be a node in $V \backslash W$, which provides*

$$a_W^+(v, u) = a_{W+x}^+(v, u) - a_{W+x}^+(v, x) \cdot a_{W+x}^+(x, u) - a_{W+x}^-(v, x) \cdot a_{W+x}^-(x, u) \tag{3}$$

$$a_W^-(v, u) = a_{W+x}^-(v, u) - a_{W+x}^+(v, x) \cdot a_{W+x}^-(x, u) - a_{W+x}^-(v, x) \cdot a_{W+x}^+(x, u) \tag{4}$$

Proof. We call a path from v to u a positive (negative) one if there are an even (odd) number of negative edges on the path. If v is positively activated, then it may positively (negatively) activate u via a positive (negative) path. Let $P^+(G_{W+x}, v, u)$ and $P^+(G_W, v, u)$ be the set of independent positive paths from v to u in G_{W+x} and G_W, respectively, which provides

$$P^+(G_{W+x}, v, u) = P^+(G_W, v, u) \cup P^+_{v \to x \to u}$$

Here, $P^+_{v \to x \to u}$ is the set of all the independent positive paths in G_{W+x} from v to u through node x. The probability of v positively activating u via the paths in $P^+_{v \to x \to u}$ is $a_{W+x}^+(v, x) \cdot a_{W+x}^+(x, u) + a_{W+x}^-(v, x) \cdot a_{W+x}^-(x, u)$. Since$x \notin W$, all the paths in $P_{v \to x \to u}$ and $P(G_W, v, u)$ are mutually independent, and we have

$$a_{W+x}^+(v, u) = a_W^+(v, u) + a_{W+x}^+(v, x) \cdot a_{W+x}^+(x, u) + a_{W+x}^-(v, x) \cdot a_{W+x}^-(x, u)$$

Similarly, we can prove that

$$a_{W+x}^-(v, u) = a_W^-(v, u) + a_{W+x}^+(v, x) \cdot a_{W+x}^-(x, u) + a_{W+x}^-(v, x) \cdot a_{W+x}^+(x, u)$$

Q.E.D.

Let W be a partial seed set selected in a greedy method and x be a candidate seed node to be added to W. From Lemma 1, we can see that after adding node x to W, the increment of the probability for v positively activating u is $a_{W+x}^+(v, x) \cdot a_{W+x}^+(x, u) + a_{W+x}^-(v, x) \cdot a_{W+x}^-(x, u)$. Such an increment can be used to select the candidate node for seed set in the greedy algorithm.

4. Propagation function and property

Based on the activating probability defined in Section 3, we define a propagation function, which can estimate the positive influence spreading by a seed set. The propagation function will be used in Section 5 to solve the PIM-SN problem.

4.1. The propagation function

To calculate the probability that a seed S can positively or negatively activate a node u, we can use the activation probabilities from all the seed nodes in set S to u through the independent paths. For a seed node set $S \subset V$, we use $G^+(S, u)$and $G^-(S, u)$ to denote the probabilities of S positively and negatively activating u, respectively, obtaining

$$G^+(S, u) = \sum_{v \in S} a^+_{V-S+v}(v, u) \tag{5}$$

$$G^-(S, u) = \sum_{v \in S} a^-_{V-S+v}(v, u) \tag{6}$$

In (5), $a^+_{V-S+v}(v, u)$ is the probability of seed v in S activating u through independent paths in sub-graph $G_{V\text{-}S++v}$. For two seeds v and w in S, the independent paths for v and w activating u in sub-graphs $G_{V\text{-}S++v}$ and $G_{V\text{-}S++w}$ are mutually independent. Therefore, we can add up the probabilities $a^+_{V-S+v}(v, u)$ of all the nodes in seed set S to get $G^+(S, u)$. Similarly, $G^-(S, u)$can be obtained by adding up the probabilities $a^-_{V-S+v}(v, u)$ of all the nodes in seed set S.

The positive propagation function of the seed set S can be estimated as follows:

$$I^+(S) = \sum_{u \in V} G^+(S, u) \tag{7}$$

Our goal is then to achieve a seed set S^* of size k satisfying

$$S^* = \arg\max_{S \subset V, |S|=k} I^+(S) \tag{8}$$

4.2. Calculating the propagation increment

In our algorithm, we employ the greedy method to select the optimal seed set. The algorithm initializes the seed set S as an empty one. It then sequentially selects the node with the largest propagation increment and adds it to S. This seed selection procedure is iterated until the size of the seed set reaches k. To avoid simulating the influence spreading for selecting the best candidate seed nodes, we define the positive influence propagation increment based on the propagation function.

Definition 5 (Positive Influence Propagation Increment)**.** Let S be the partial seed set and $x \in V \backslash S$ be a candidate seed node. The positive propagation increment $\Delta^+_S(x)$ after adding x into S is defined as

$$\Delta^+_S(x) = I^+(S \cup \{x\}) - I^+(S) \tag{9}$$

First, we present the following theorem to analyze the activation probability of a node u after adding x to S.

Theorem 1. *Let S be the partial seed set and $x \in V \backslash S$ be a candidate node. For a node $u \in V$, we have*

$$G^+(S+x, u) - G^+(S, u) = a^+_{V \backslash S}(x, u)\left[1 - G^+(S, x)\right] - a^-_{V \backslash S}(x, u) G^-(S, x) \tag{10}$$

$$G^-(S+x, u) - G^-(S, u) = a^-_{V \backslash S}(x, u)\left[1 - G^-(S, x)\right] - a^+_{V \backslash S}(x, u) G^+(S, x) \tag{11}$$

Proof. Here, we only prove Eq. (10); Eq. (11) can be proved similarly.

$$\begin{aligned} G^+(S+x, u) - G^+(S, u) &= \sum_{v \in S+x} a^+_{V-(S+x)+v}(v, u) - \sum_{v \in S} a^+_{V-S+v}(v, u) \text{ (byEq.(5))} \\ &= a^+_{V-S}(x, u) + \sum_{v \in S} \left[a^+_{V-(S+x)+v}(v, u) - a^+_{V-S+v}(v, u)\right] \\ &= a^+_{V-S}(x, u) - \sum_{v \in S} \left[a^+_{V-S+v}(v, x) \cdot a^+_{V-S+v}(x, u) + a^-_{V-S+v}(v, x) \cdot a^-_{V-S+v}(x, u)\right] \text{ (byEq.(3))} \end{aligned}$$

In the above equation, $\sum_{v \in S} a^+_{V-S+v}(v, x) \cdot a^+_{V-S+v}(x, u) + \sum_{v \in S} a^-_{V-S+v}(v, x) \cdot a^-_{V-S+v}(x, u)$ in the second term considers all the independent positive paths from v to u through x in sub-graph G_{V-S+v}, i.e., the paths of the form $v \to x \to u$. Since there is no loop in these paths, node v will not appear on each of such paths except at the start point. Namely, v must not appear on the sub-path of $x \to u$, so the terms $a^+_{V-S+v}(x, u)$ and $a^-_{V-S+v}(x, u)$ here can be replaced by $a^+_{V-S}(x, u)$ and $a^-_{V-S}(x, u)$, respectively. Therefore, we can obtain

$$\begin{aligned}G^+(S+x,u)-G^+(S,u) &= a^+_{V-S}(x,u)-\sum_{v\in S}a^+_{V-S+v}(v,x)\cdot a^+_{V-S}(x,u)-\sum_{v\in S}a^-_{V-S+v}(v,x)\cdot a^-_{V-S}(x,u)\\ &= a^+_{V-S}(x,u)\left[1-\sum_{v\in S}a^+_{V-S+v}(v,x)\right]-a^-_{V-S}(x,u)\sum_{v\in S}a^-_{V-S+v}(v,x)\\ &= a^+_{V-S}(x,u)\big[1-G^+(S,x)\big]-a^-_{V-S}(x,u)G^-(S,x)\end{aligned}$$

Q.E.D.

According to Theorem 1, we present the following theorem to obtain an estimation of $\Delta_S^+(x)$.

Theorem 2. *Let $S\subset V$ be the current seed set and x be a candidate node in $V\backslash S$. The positive propagation increment $\Delta_S^+(x)$ for adding x into S is*

$$\begin{aligned}\Delta_S^+(x) &= I^+(S+x)-I^+(S)\\ &= \big[1-G^+(S,x)\big]\cdot\sum_{u\in V}a^+_{V-S}(x,u)-G^-(S,x)\sum_{u\in V}a^-_{V-S}(x,u)\end{aligned} \tag{12}$$

Proof.

$$\begin{aligned}\Delta_S^+(x) &= I^+(S+x)-I^+(S)\\ &= \sum_{u\in V}\big[G^+(S+x,u)-G^+(S,u)\big]\\ &= \sum_{u\in V}\Big\{\big[1-G^+(S,x)\big]a^+_{V-S}(x,u)-G^-(S,x)a^-_{V-S}(x,u)\Big\}\\ &= \big[1-G^+(S,x)\big]\cdot\sum_{u\in V}a^+_{V-S}(x,u)-G^-(S,x)\sum_{u\in V}a^-_{V-S}(x,u)\end{aligned}$$

Q.E.D.

5. Positive influence maximization

In this section, we present the framework of our algorithm PIMSN (positive influence maximization in signed networks). The algorithm first computes the activating probabilities $a^+(v,u)$ and $a^-(v,u)$ for the node pairs by calling Algorithm *Comp-pp*, which is presented in Section 3. Based on the propagation function defined in Section 4, algorithm PIMSN uses a greedy strategy to choose the node with the maximal propagation increment as the candidate seed.

5.1. Selecting the seed nodes

The algorithm initializes the seed set S as an empty one. We use variables $a^+(v,u)$ and $a^-(v,u)$ respectively to denote the activation probabilities $a^+_{V\backslash S}(v,u)$ and $a^-_{V\backslash S}(v,u)$ under the current seed set S. First, the positive and negative activation probabilities $a^+(v,u)$ and $a^-(v,u)$ between all node pairs $(v,\ u)$ are calculated by algorithm *Comp-PP*. We use variables $G^+(u)$ and $G^-(u)$ to respectively denote the probabilities $G^+(S,u)$ and $G^-(S,u)$ for the current seed set S to positively and negatively activate u. The initial values of $G^+(u)$ and $G^-(u)$ are 0 since the seed set S is initially empty. We use variable $\Delta^+(x)$ to denote $\Delta_S^+(x)$ which is the increment of positive spreading when x is added to the current seed set S. The initial value of $\Delta^+(x)$ is

$$\Delta^+(x)=\sum_{v\in V}a^+(x,u)-\sum_{v\in V}a^-(x,u) \tag{13}$$

In each iteration, node x with the largest positive influence spreading increment $\Delta^+(x)$ is selected from the set $V\backslash S$ to join the seed set. After the addition of x to the seed set S, the values of variables $G^+(u)$, $G^-(u)$, $a^+(v,u)$, $a^-(v,u)$ and $\Delta^+(v)$ are updated accordingly.

According to (10) and (11), after the addition of x the seed set S, values of $G^+(u)$, $G^-(u)$ should be updated as follows:

$$G^+(u)\leftarrow G^+(u)+a^+(x,u)\big[1-G^+(x)\big]-a^-(x,u)G^-(x) \tag{14}$$

$$G^-(u)\leftarrow G^-(u)+a^-(x,u)\big[1-G^-(x)\big]-a^+(x,u)G^+(x) \tag{15}$$

Replacing the set $W++x$ in (6) with $V\backslash S$ and replacing W with $V\backslash(S++x)$, we can obtain

$$\begin{aligned}a^+_{V\backslash(S+x)}(v,u) &= a^+_{V\backslash S}(v,u)-a^+_{V\backslash S}(v,x)\cdot a^+_{V\backslash S}(x,u)-a^-_{V\backslash S}(v,x)\cdot a^-_{V\backslash S}(x,u)\\ a^-_{V\backslash(S+x)}(v,u) &= a^-_{V\backslash S}(v,u)-a^+_{V\backslash S}(v,x)\cdot a^-_{V\backslash S}(x,u)-a^-_{V\backslash S}(v,x)\cdot a^+_{V\backslash S}(x,u)\end{aligned}$$

Therefore, after the addition of x to the seed set S, the values of $a^+(v,u)$ and $a^-(v,u)$ should be updated as follows:

$$\begin{aligned} a^+(v,u) &\leftarrow a^+(v,u) - a^+(v,x)\cdot a^+(x,u) - a^-(v,x)\cdot a^-(x,u) \\ a^-(v,u) &\leftarrow a^-(v,u) - a^+(v,x)\cdot a^-(x,u) - a^-(v,x)\cdot a^+(x,u) \end{aligned} \tag{16}$$

According to (12), using the updated values of $G^+(u)$, $G^-(u)$, $a^+(v,u)$ and $a^-(v,u)$, the increment of positive spreading $\Delta^+(v)$ can be updated as follows:

$$\Delta^+(v) \leftarrow \left[1 - G^+(v)\right]\cdot\sum_{u\in V} a^+(v,u) - G^-(v)\sum_{u\in V} a^-(v,u) \tag{17}$$

In the next iteration, a new seed with the largest increment of positive spreading can be selected. This iterative process is repeated until number of the nodes in the seed set reaches k.

5.2. Framework of algorithm pimsn

Based on the above discussion, the framework of algorithm PIMSN (positive influence maximization in signed networks) is described in Fig. 4.

In Algorithm 2, step 1 first initializes the seed set as an empty one. Then, step 2 computes the activating probabilities $a^+(v,u)$ and $a^-(v,u)$ for the node pairs by calling Algorithm *Comp-pp*. Based on the activating probabilities, the initial value of $\Delta^+(x)$ can be computed according to (14) and (15) in step 3. Since the seed set is initially empty, the probabilities $G^+(x)$ and $G^-(x)$ for positively and negatively activating each node u is initialized as zero.

In step 4, the procedure of selecting a seed node is iterated k times, and k nodes are selected to form a seed set S. In each iteration, node x in $V\backslash S$ with the largest increment $\Delta^+(x)$ is selected as a new seed. After the addition of node x to the seed set, values of $G^+(v)$, $G^-(v)$, $a^+(v,u)$ and $a^-(v,u)$ are updated accordingly. Based on the new values of $G^+(v)$, $G^-(v)$, $a^+(v,u)$ and $a^-(v,u)$, the new increment $\Delta^+(v)$ of each node v in $V\backslash S$ can be computed according to (17). The node in $V\backslash S$ with the largest increment is selected as a new seed. Repeating this procedure k times, the seed set S of size k can be constructed.

Fig. 5 shows the process of computing data G, a and Δ in the iterations. The figure depicts the dependencies between datasets G, a and Δ. In the figure, $G(u)$ represents both $G^+(u)$ and $G^-(u)$, $a(v,u)$ represents both $a^+(v,u)$ and $a^-(v,u)$ and $\Delta(u)$ represents both $\Delta^+(u)$ and $\Delta^-(u)$.

5.3. Complexity analysis for algorithm pimsn

In the algorithm, step 2 computes the activating probabilities $a^+(v,u)$ and $a^-(v,u)$ for the node pairs by Algorithm 1. When the network is represented by the adjacency list, the time required for DFS to access the adjacency list of all edges is $O(|E|)$, and the time required to access the nodes is $O(|V|)$. In this case, the time complexity of algorithm 1 is $O(|V|+|E|)$.When the network is represented by an adjacency matrix, the time required by the modified DFS in Algorithm 1 to scan the neighbors of each node is $O(|V|)$. Therefore, the total time complexity of Algorithm 1 is $O(|V|^2)$ in this case. Step 2 requires $O(|V|^2)$ memory to store $a^+(v,u)$ and $a^-(v,u)$ for all pairs of nodes.

Step 3 initializes $\Delta^+(x)$, $G^+(v)$ and $G^-(v)$ in $O(|V|)$ time. In step 4, seeds are selected iteratively to form the set S. In each time step, the node x, which has the highest increment $\Delta^+(x)$, is chosen as the new seed to join S, and the values of $G^+(v)$, $G^-(v)$, $a^+(v,u)$, $a^-(v,u)$ and $\Delta^+(x)$ are updated accordingly in $O(|V|^2)$ time. To select k seed nodes, step 4 requires $O(k.\ |V|^2)$ time. Since k is a constant, the time complexity of algorithm PIMSN is $O(|V|^2)$ when the network is represented by an adjacency matrix. When the network is represented by the adjacency list, the time complexity of algorithm PIMSN is $O(|E|+|V|^2)$.Since $|E|\le|V|^2$, the time complexity of the algorithm is $O(|V|^2)$ in this case. The memory requirement of algorithm PIMSN is also $O(|V|^2)$.

6. Experimental results and analysis

To verify the effectiveness and performance of our algorithm PIMSN, we test it on four real-world networks. Section 6.1 introduces the networks tested in the experiment. Section 6.2 describes the influence maximization algorithms tested and compared with PIMSN in the experiments. Sections 6.3 to 6.5 show and analyze the experimental results.

In our experiments, all algorithms are coded in C++ and run on Intel(R) Core(TM)i5 CPU 2.40 GHz, 10.0GB RAM under Windows 7 system.

6.1. Datasets

We conduct experiments to test our algorithm PIMSN on four network datasets: Epinions, Wiki-Vote, BlogCatalog and Gnutella.

Epinions [15] is a website where consumers can discuss commodities. They can also give their opinions about the comments of other users. According to their opinions, positive and negative relations between them can be defined.

Algorithm 2 *PIMSN* (positive influence maximization in signed networks)

Input: $G=(V,E,P)$: the signed network;

k: size of the seed set;

Output: S: the seed nodes set;

Begin

1. $S=\phi$; //*Initialize the seed set*//

2. *Comp-pp*;

   //* Calculate $a^+(v,u)$ and $a^-(v,u)$ for all the node pairs by algorithm *Comp-pp* *//;

3. **for** all $x \in V$ **do** //*Initialize $\Delta^+(x)$, $G^+(x)$, $G^-(x)$*//

   $$\Delta^+(x)=\sum_{u\in V} a^+(x,u)-\sum_{u\in V} a^-(x,u);$$

   $$G^+(x)=G^-(x)=0 \quad;$$

   endfor;

4. **Repeat**

   4.1 $x=\arg\max_{v\in V\setminus S} \Delta^+(v)$

   /* Select x with the largest increment from $V\setminus S$ as a new seed*/ ;

   4.2 **For** $\forall v\in V\setminus S$ **do**

   update the value of $G^+(v)$ according to (14)

   update the value of $G^-(v)$ according to (15)

   endfor;

   4.3 **for** $\forall u,v\in V$ **do**

   update the values of $a^+(v,u)$ and $a^-(v,u)$ according to (16)

   endfor ;

   4.4 $S=S\cup\{x\}$;

   4.5 **For** $\forall v\in V\setminus S$ **do**

   Update the value of $\Delta^+(v)$ according to (17)

   endfor;

   Until the size of S is k

End

Fig. 4. Algorithm PIMSN.

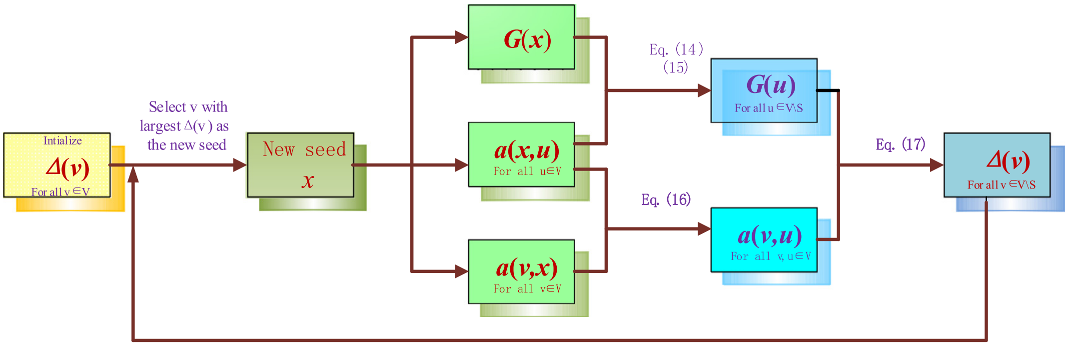

Fig. 5. Process of computing data G, a and Δ.

Table 2
Topological attributes of the four tested networks.

Datasets	#Nodes	#Edges	Average degree
Epinions	75,879	508,837	13.4
Wiki-Vote	7115	103,689	26.64
BlogCatalog	10,312	333,983	32.39
Gnutella	62,586	147,892	4.73

Wiki-Vote [16] is a network formed according to Wikipedia. It records the voting history in Wikipedia. Nodes in the network stand for the users of Wikipedia, and a signed link from user u to v indicates that u voted for or against recommending v to the administration.

BlogCatalog [11] is a social blog directory website where each blog is associated with information, such as the positive and negative relations in the social network crawled and the memberships of various groups. The nodes represent the users, and undirected links indicate the positive and negative relations between users.

Gnutella [17] is the first decentralized peer-to-peer large scale network. In the network, hosts are represented by the nodes, while the interconnections among the hosts are represented by the links. Gnutella consists of 9 snapshots observed in one month.

Table 2 lists the detailed topological attributes of the four networks, such as the average degrees, numbers of edges and nodes.

In the abovementioned datasets, the propagation probabilities on the edges are set by two models: random IC (RICM) and weighted IC (WICM). In the RICM model, for each link (u,v) is randomly set a probability $p \in [0, 0.09]$ to represent the activeness of the node v. In the WICM modes, each link (u,v) is set a probability of $1/d_v$ for activating v. Here, d_v denotes the degree of node v.

In those social networks tested, signs on the edges are implicitly given by the datasets. For two connected users in a social network, the sign on the link between them can be observed according to their opinions or comments on some topic or product. For instance, in Epinions network, trust (positive) or distrust (negative) relations between users are determined according to their comments on the products. Attached to the Epinions dataset is a document named “user_rating.txt”, where each user lists the IDs of the other users he trusted or distrusted. With this document, a signed network can be constructed. For the Gnutella network, we define the signs on the edges according to their directions. We construct an undirected signed graph from the original directed one. If there are two edges in opposite directions between a pair of nodes, we combine them into one positive undirected edge. If there is only one directed edge between two nodes, we set the sign on the edge according to the centralities of the nodes. If the starting node has higher centrality, we set a positive undirected edge between the nodes; otherwise we set a negative undirected edge.

6.2. Compared algorithms

In the experiments, we test and compare influence spreading by our PIMSN algorithm with five other algorithms.

Random: This is one of the most commonly used methods for influence maximization. In the method, k nodes are randomly selected to construct a seed set. In our experiments, such a random seed selection process is repeated 1000 times, and the average spreading by the 1000 seed sets is the output result of the algorithm.

MaxDegree [25]: This is a heuristic algorithm that uses degree centrality. MaxDegree considers the node with the highest degree centrality to be the most influential one. The algorithm will repeatedly select and add the node that has the highest degree centrality into the seed set until the number of seed nodes reaches k.

Greedy [24]: This algorithm initializes the seed set as an empty one and repeatedly chooses a potential candidate to join the seed nodes. In each round of seed selection, the method chooses the node with the largest marginal gain of influence spreading as the candidate seed. The marginal gains of the nodes are estimated by Monte-Carlo simulations.

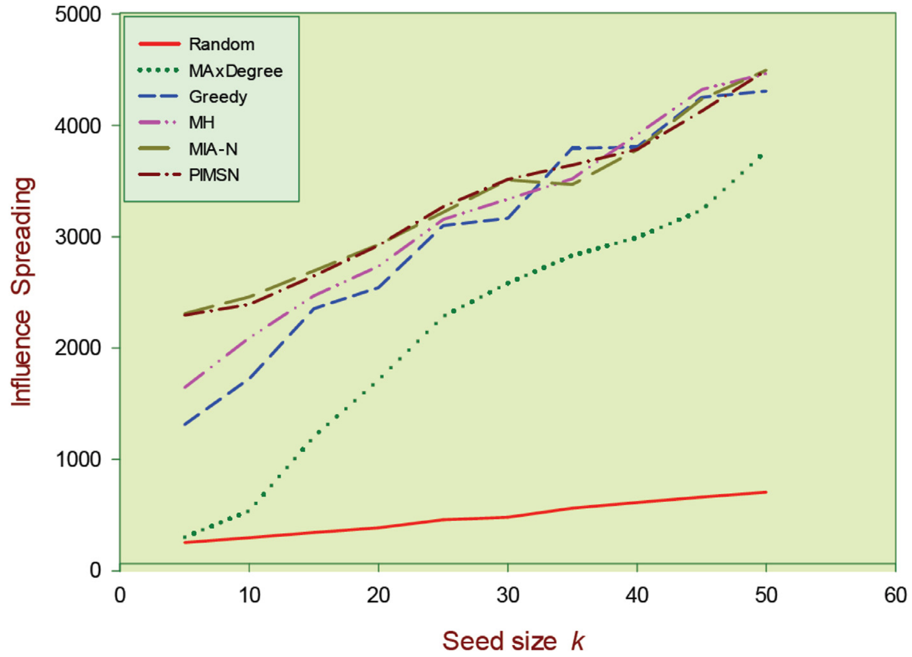

Fig. 6. Positive influence spreading by the algorithms on Epinions using the WICM model.

Meta heuristic (MH) [26]: Canh V. Pham proposed the algorithm for solving the positive influence maximization while negative influence is limited during the propagation within d hops (d-PIMCN). The method treats the D-PIMCN problem as a $0-1$ integer linear programming and solves it using heuristic optimization.

MIA-N [3]**:** Wei Chen presented the algorithm MIA-N for influence maximization in networks with two types of opposite relations. Based on an influence propagation sketch in the tree structure, MIA-N performs a heuristic optimization to maximize the positive influence.

6.3. Test of influence spreading with different seed sizes

We test and compare the positive influence spreading of the six algorithms on the four data sets using the RICM and WICM models.

6.3.1. Test of the epinions dataset

Figs. 6 and 7 show and compare the positive influence spreading by the six algorithms on the Epinions dataset. In the figures, we can see that under the RICM model, our algorithm PIMSN has the best performance among the six algorithms. MIA-N can obtain the second highest spreading among all the algorithms, and MH and Greedy have nearly the same performance. Using the WICM model, algorithm PIMSN can achieve greater positive influence spreading than the other five algorithms on all seed sizes except 40 and 45. When $k=40$ and 45, the spreading of PIMSN is slightly less than that of MH. Using the WICM model, MIA-N, MH and Greedy have nearly the same performance. Using both the RICM and WICM models, Random shows the worst performance among the six algorithms. Algorithm MaxDegree performs slightly better than Random using both models. In summary, the proposed algorithm PIMSN outperforms all five other algorithms for the Epinions dataset using both the RICM and WICM models.

6.3.2. Test of the wiki-vote dataset

Figs. 5 and 6 show and compare the positive influence spreading by different algorithms on the Wiki-Vote dataset. Figs. 8 and 9 show that algorithm PIMSN outperforms the other five algorithms for the Wiki-Vote dataset using the RICM model. Using the WICM model, algorithm PIMSN outperforms the other five algorithms with a seed size $k>20$. When $k\leq 20$, the influence spreading of PIMSN is slightly reduced compared with MIA-N. MIA-N can achieve the second largest spreading using both models, but its spreading is less than Greedy when $k>35$ when using the WICM model. Moreover, algorithm Random still shows the worst performance among all six algorithms for the Wiki-Vote dataset using both the RICM and WICM models. Algorithm MaxDegree performs slightly better than Random using both models.

6.3.3. Test of the blogcatalog dataset

Figs. 10 and 11 show and compare the positive influence spreading by the six algorithms on the BlogCatalog data set. From the figures, we can see that our algorithm PIMSN has the best performance among the six algorithms for BlogCatalog

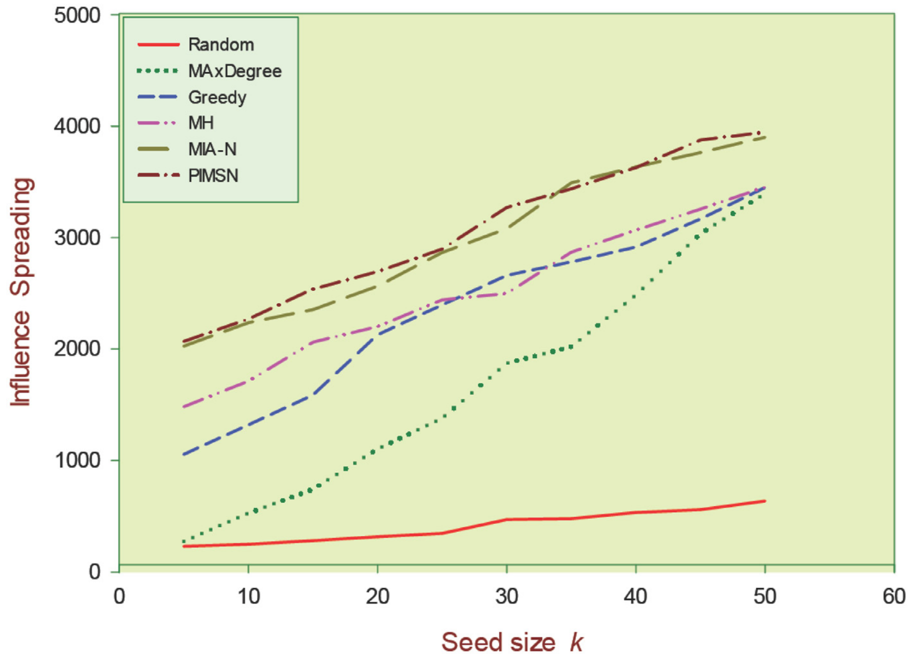

Fig. 7. Positive influence spreading by the algorithms on Epinions using the RICM model.

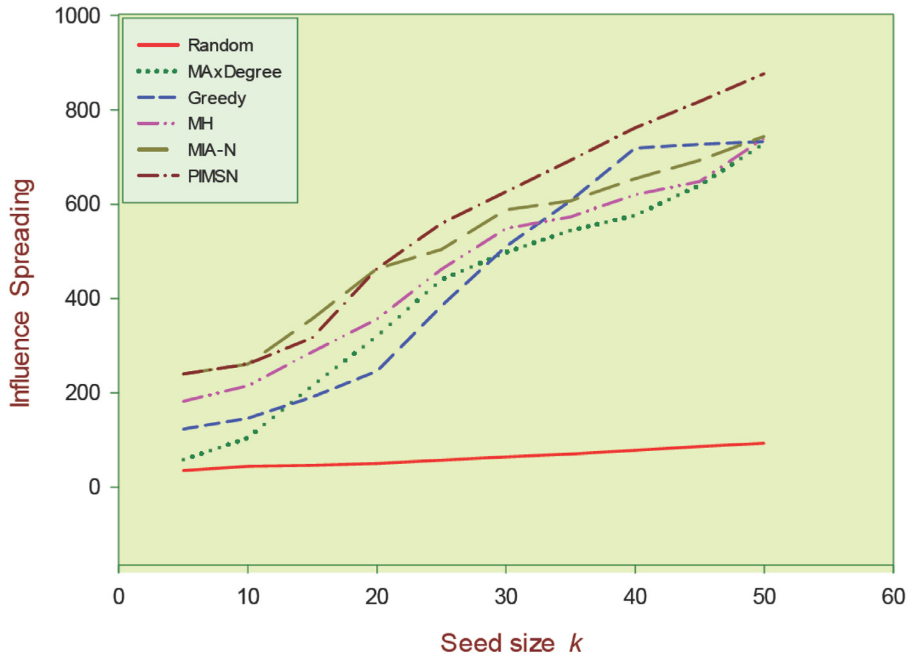

Fig. 8. Positive influence spreading by the algorithms on Wiki-Vote using the WICM model.

dataset using both models. However, for the two models, PIMSN has larger positive influence spreading with the WICM model than the RICM model. When $k \leq 25$, the positive influence spreading ranges of algorithms PIMSN, MIA-N, Greedy and MH are very close to each other. However, when $k > 25$, algorithm PIMSN achieves much larger positive influence spreading than MIA-N, MH and Greedy. Overall, algorithms MIA-N, MH and Greedy have roughly the same performance in both models. Figs. 10 and 11 show that algorithm Random performs the worst using the WICM model, while MaxDegree performs the worst using the RICM model. These findings indicate that the performances of algorithms MaxDegree and Random are unstable.

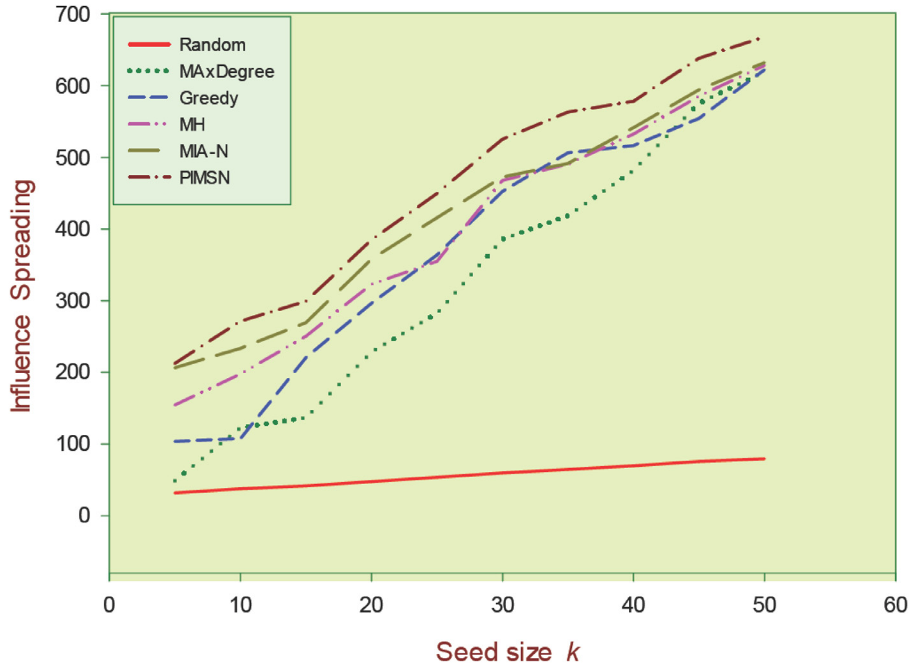

Fig. 9. Positive influence spreading by the algorithms on Wiki-Vote using the RICM model.

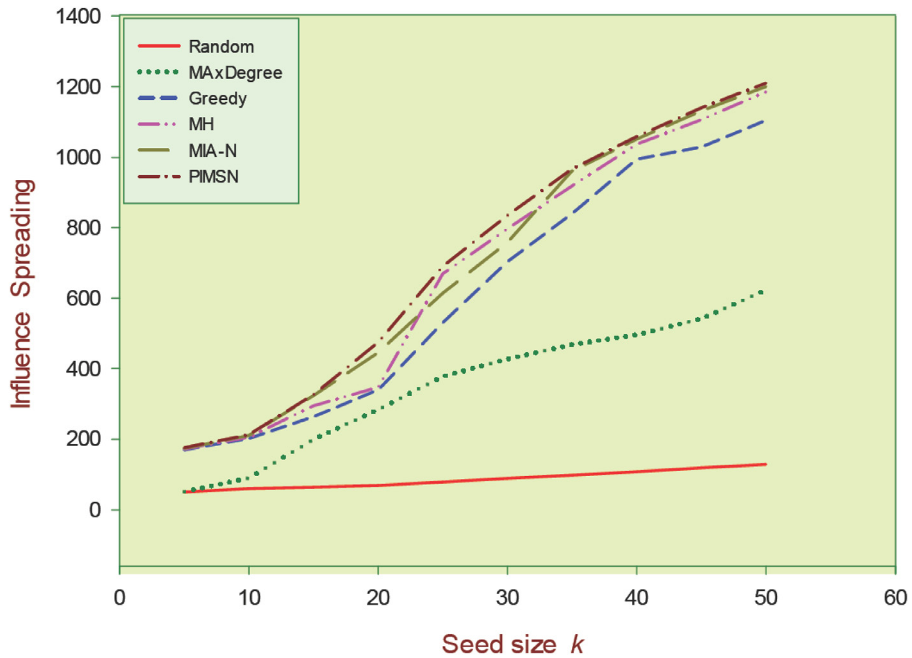

Fig. 10. Positive influence spreading by the algorithms on BlogCatalog using the WICM model.

6.3.4. Test of the gnutella dataset

Figs. 12 and 13 show and compare the positive influence spreading by the six algorithms on the Gnutella dataset. In the figures, we can see that algorithm PIMSN obtains the largest positive influence spreading compared with the other five algorithms for the Gnutella data-set using both models. Among the other five algorithms, MIA-N, MH and Greedy have nearly the same performance with seed node size k<20 using the WICM model. However, with larger seed set sizes, MIA-N outperforms algorithms MH, MaxDegree, Greedy and Random. MaxDegree's performance is better with the WICM model than the RICM model. However, the performance of Random is still the worst among the six algorithms for the Gnutella dataset.

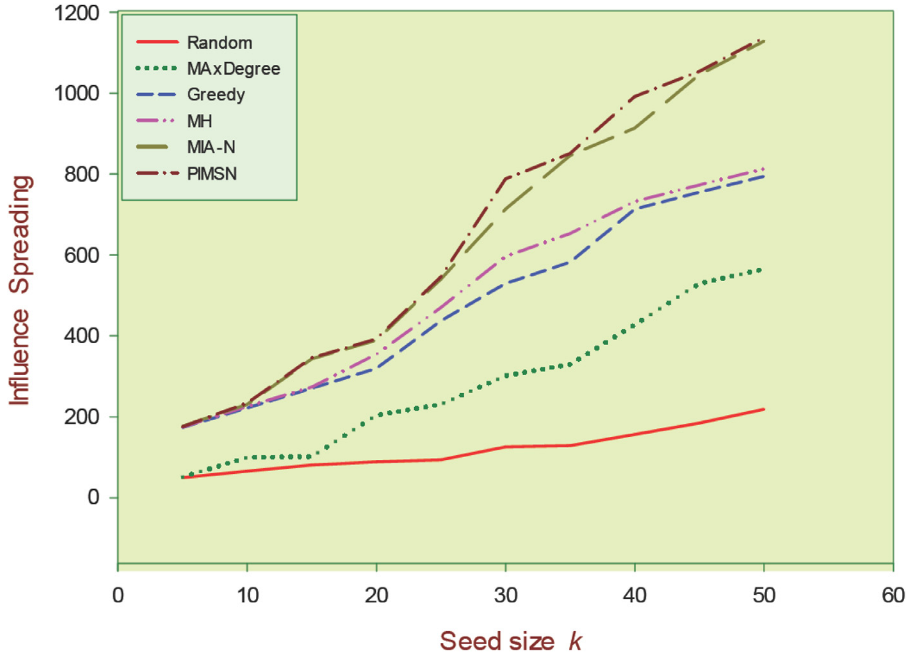

Fig. 11. Positive influence spreading by the algorithms on BlogCatalog using the RICM model.

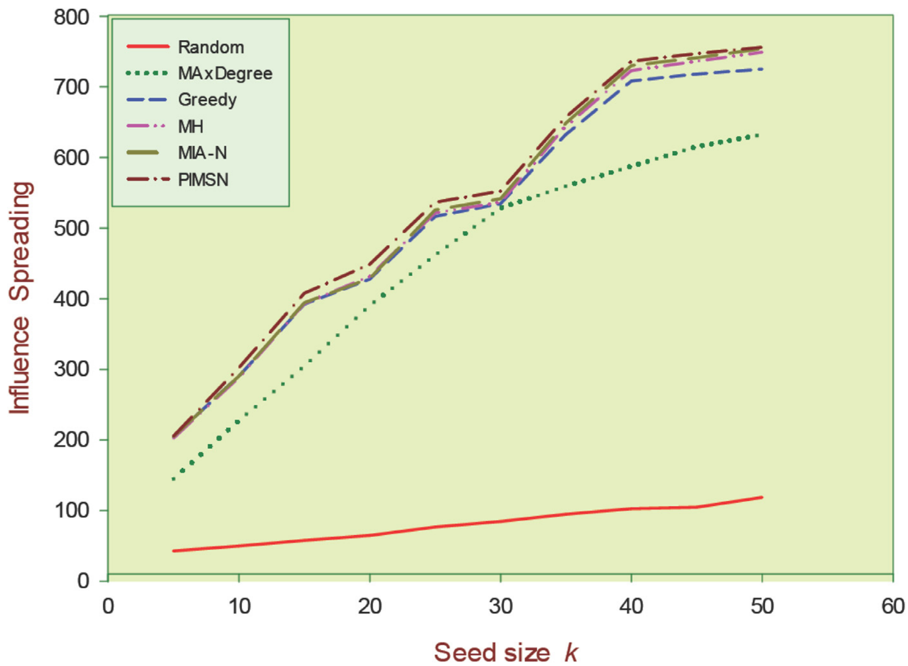

Fig. 12. Positive influence spreading by the algorithms on Gnutella using the WICM model.

From the empirical results shown above, we can find that our algorithm PIMSN has the greatest positive influence spreading among all the algorithms tested for all the real-world datasets using both models. These results indicate that PIMSN has stable performance in term of positive influence spreading. Additionally, PIMSN has larger positive influence spreading in the WICM model than in the RICM model. Among the other five algorithms, the performance of algorithm MIA-N exceeds MH, Greedy, MaxDegree and Random in most cases. Algorithm Random shows the worst performance for all datasets. Algorithm PIMSN achieves the largest positive influence spreading because it considers influence spreading for both positive and negative relations in estimating the positive influence increment and limits negative influence spreading to users.

Please cite this article as: W. Ju, L. Chen and B. Li et al., A new algorithm for positive influence maximization in signed networks, Information Sciences, https://doi.org/10.1016/j.ins.2019.10.061

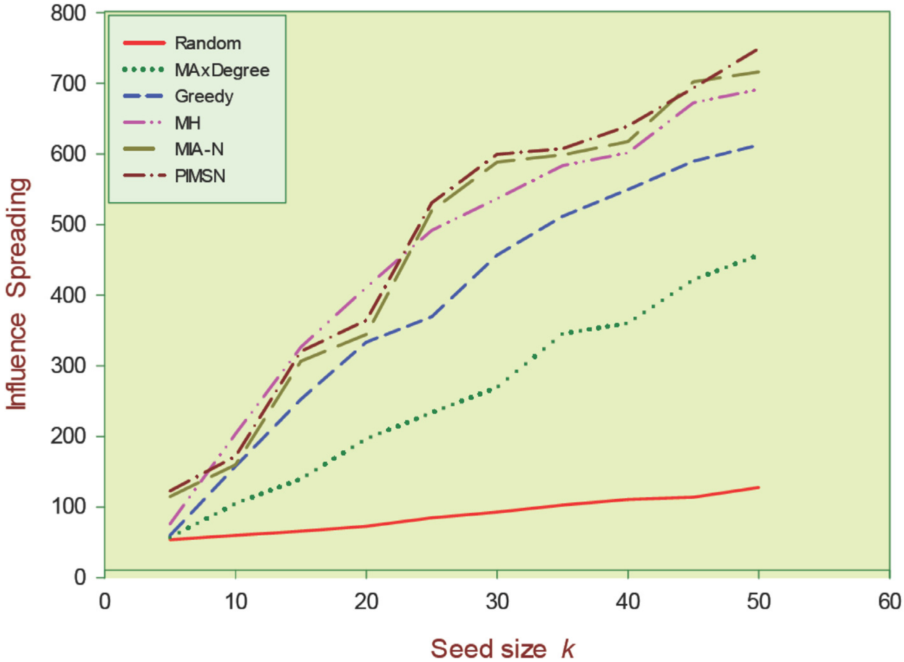

Fig. 13. Positive influence spreading by the algorithms on Gnutella using the RICM model.

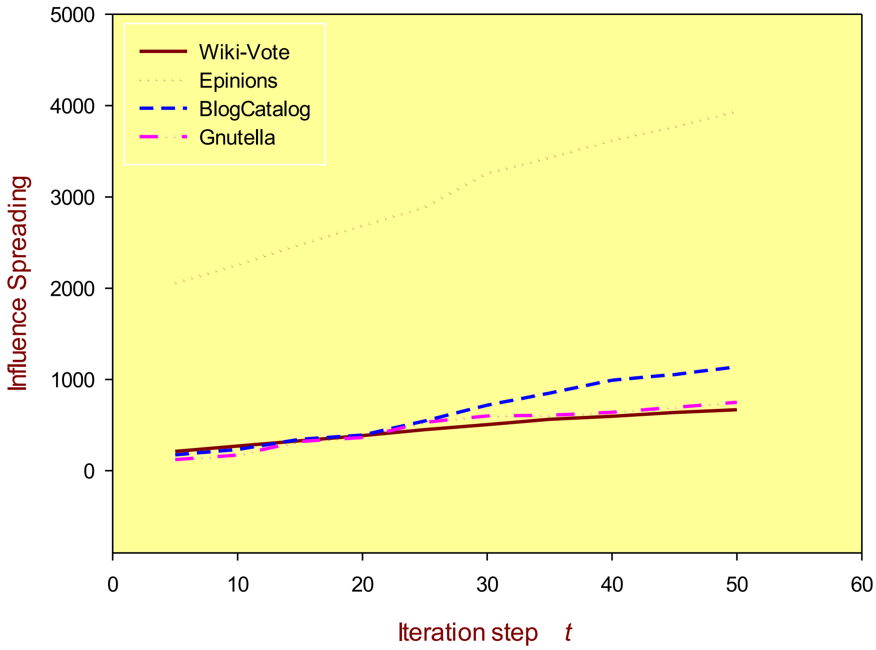

Fig. 14. Positive influence spreading in different iterations using the RICM model.

6.4. Test of the influence spreading in different iterations

To demonstrate the efficiency of our algorithm PIMSN, we also test the positive influence spreading at different iteration steps in PIMSN using the four datasets. Figs. 14 and 15 show the results using the WICM and RICM models, respectively.

The figures demonstrate that the spreading of PIMSN steadily expands with the number of iteration steps for the four datasets using both models. Especially for datasets Epinions and BlogCatalog, the spreading of positive influence grows even faster after 20 iterations. This result shows that the seed set obtained by algorithm PIMSN can effectively propagate the positive influence to the maximum range.

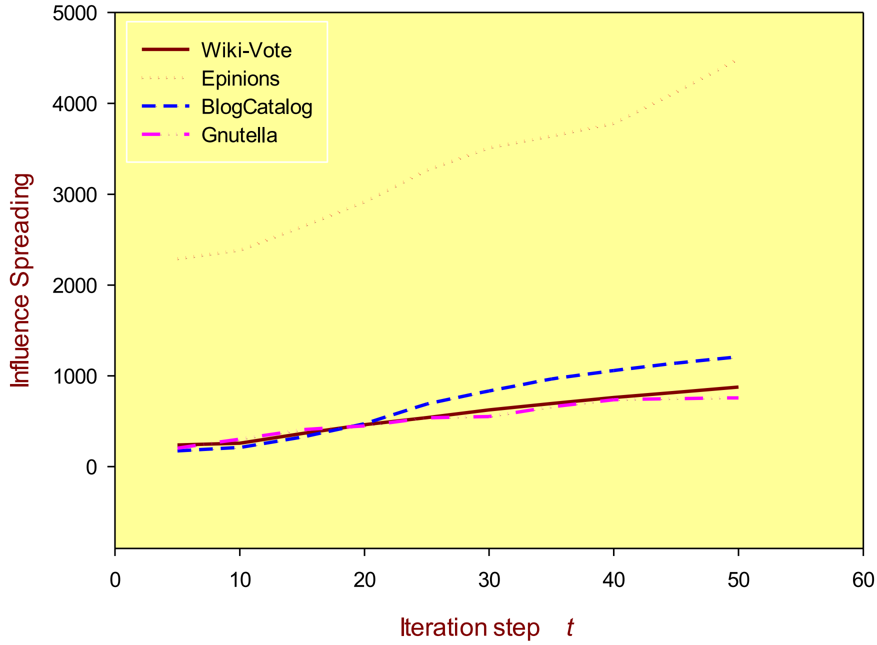

Fig. 15. Positive influence spreading in different iterations using the WICM model.

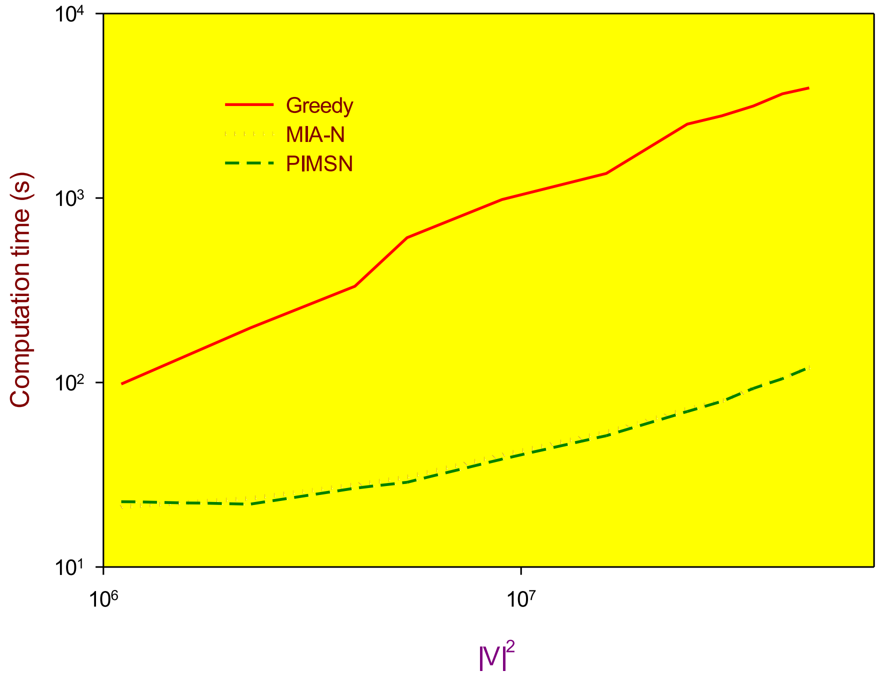

Fig. 16. Comparison of the computation time for Epinions.

6.5. Test of the computation time and memory requirements

We test the computation time and the memory requirements of the proposed algorithm PIMSN and compare them with those of other algorithms. First, we test the computation times of algorithms PIMSN, Greedy and MIA-N, since they obtain similar spreading results. We test the computation times of the algorithms for constructing the seed set of size 50for the datasets Epinions and Wiki-Vote.

The results are shown in Figs. 16 and 17, where we can see that algorithm PIMSN is several magnitudes faster than Greedy and requires almost the same amount of time as MIA-N. In the experiments, we also test the computation times of

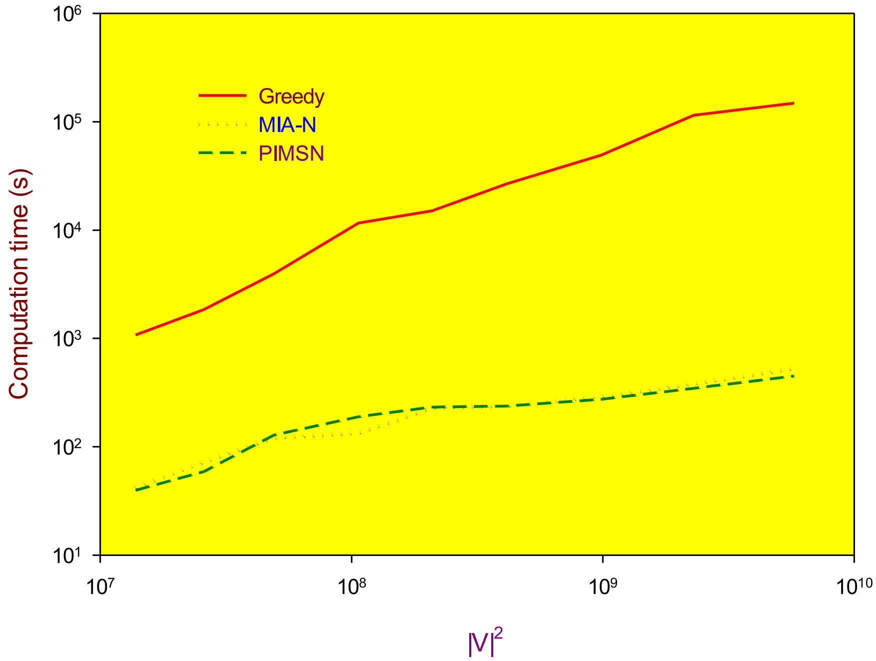

Fig. 17. Comparison of the computation time for Wiki-Vote.

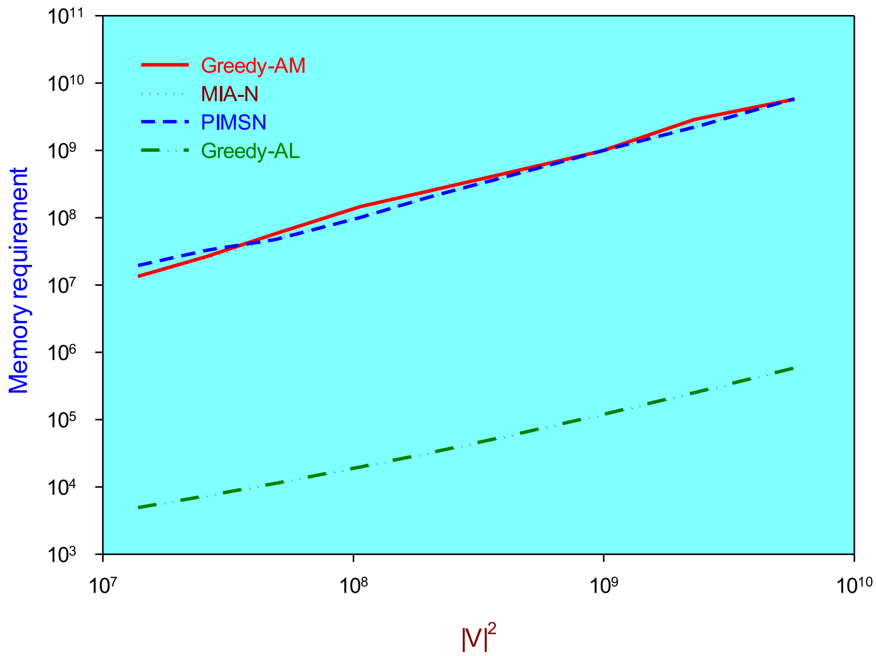

Fig. 18. Comparison of the memory requirements for Epinions.

the algorithms for networks with different numbers of nodes. From the figures, we can see that the computation time of PIMSN grows approximately linearly with $|V|^2$, which is the square of the node number. This result empirically verifies our estimation in Section 5.3 that the time complexity of algorithm PIMSN is $O(|V|^2)$.

We also test the memory requirements of the algorithms PIMSN, Greedy and MIA-N for the datasets Epinions and Wiki-Vote. The results are shown in Figs. 18 and 19. We test two versions of the Greedy algorithm: Greedy-AM, which uses the adjacent matrix to represent the network, and Greedy-AL, which uses the adjacent list. In the experiments, we compare the memory requirements of the algorithms for networks with different numbers of nodes. From the figures, we can see that the memory requirement of PIMSN is roughly proportional to $|V|^2$, which empirically verifies our estimation in Section 5.3 that

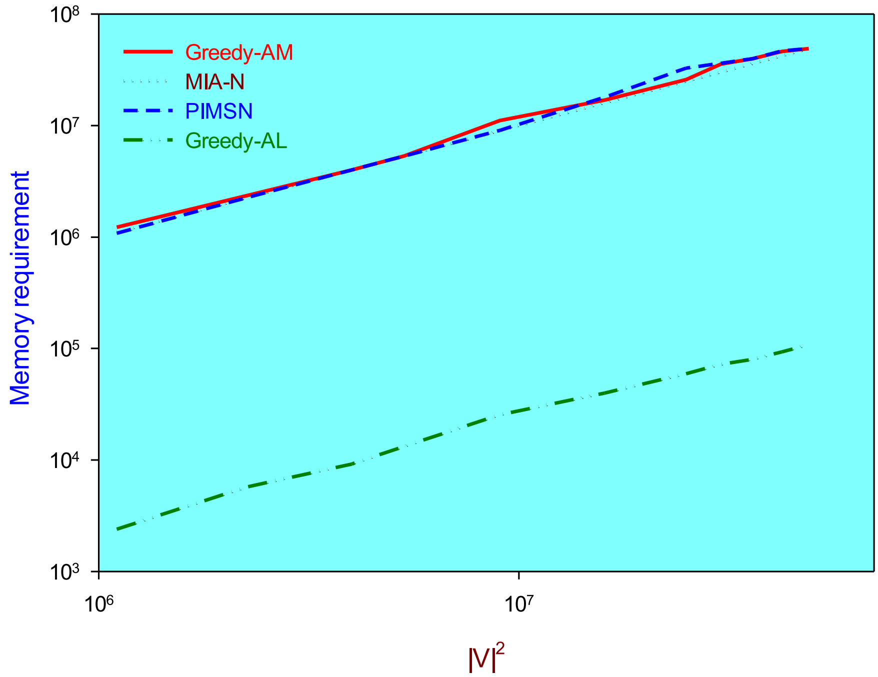

Fig. 19. Comparison of the memory requirements for Wiki-Vote.

the memory requirement of algorithm PIMSN is O($|V|^2$). The figures also show that algorithm Greedy-AL uses much less memory, which is consistent with our estimation in Section 5.3 that the memory requirement of Greedy algorithm using the adjacent list is O($|V|+|E|$).

7. Conclusions and future work

In signed social networks, there exist positive or negative relationships between individuals. In particular, negative links play a more critical role than positive ones, since they can reverse the influence. How to take advantage of different relations to effectively select influential nodes is a key issue in influence maximization for signed networks. In this work, we model this problem as positive influence maximization in signed networks in the IC model. To avoid the process of simulating the influence propagation, we propose a propagation path-based algorithm to compute the activation probabilities between node pairs. Using the activation probability, we propose a propagation increment function. To select the optimum seed set, we propose a greedy algorithm to sequentially choose the nodes with the largest influence increment to join the seed set. Empirical results on real social networks have demonstrated that our PIMSN algorithm can achieve larger positive influence spreading than other similar methods.

In some signed social networks, the relations between individuals may change frequently. Therefore, the seed selection strategy should be adjusted accordingly. As our further research work, we will design an effective approach for detecting influential nodes in such dynamic signed networks. Furthermore, we will also investigate the problem of detecting influential nodes in multiple networks, where nodes may join several social and trade networks and the influence can spread across different networks.

Declaration of Competing Interest

None.

Acknowledgments

This research was supported in part by the Chinese National Natural Science Foundation under grant nos. 61379066, 61702441, 61070047, 61379064, 61472344, 61402395, and 61602202;the Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK2012672, BK2012128, and BK20140492;and the Natural Science Foundation of Education Department of Jiangsu Province under contract 12KJB520019, 13KJB520026, 09KJB20013. Six talent peaks project in Jiangsu Province (grant no. 2011-DZXX-032).

References

[1] N.M. Ahmed, L. Chen, Y. Wang, B. Li, Y. Li, W. Liu, DeepEye: link prediction in dynamic networks based on non-negative matrix factorization, Big Data Mining and Analytics 1 (1) (2018) 19–33.
[2] A. Bozorgi, S. Samet, J. Kwisthout, T. Wareham, Community-based influence maximization in social networks under a competitive linear threshold model, Knowl. Based Syst. 134 (15) (2017) 149–158.
[3] W. Chen, A. Collens, R. Cummings, Te Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, Y. Yuan, Influence maximization in social networks when negative opinions may emerge and propagate, in: the Proceedings of the 2011 SIAM International Conference on Data Mining (SDM'2011), Mesa, Arizona, U.S.A., 2011.
[4] G. D'Angelo, L. Severini, Y. Velaj, Recommending links through influence maximization, Theor. Comput. Sci 764 (11) (2019) 30–41.
[5] Y. Dinitz, Dinitz' algorithm: the original version and even's version, in: O. Goldreich, A.L. Rosenberg, A.L. Selman (Eds.), Theoretical Computer Science: Essays in Memory of Shimon Even, Springer, 2006, pp. 218–240. ISBN 978-3-540-32880-3.
[6] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.
[7] C. Gao, Z. Su, J.M. Liu, J. Kurths, Even central users do not always drive information diffusion, Commun. ACM. 62 (2) (2019) 61–67.
[8] J.S. He, H. Kaur, M. Talluri, Positive opinion influential node set selection for social networks: considering both positive and negative relationships, in: Proceedings of the Wireless Communications, Networking and Applications, 2016, pp. 935–948.
[9] M. Daliri, K.A. Ezvanian, N. Bagherpour, M.R. Meybodi, Minimum positive influence dominating set and its application in influence maximization: a learning automata approach, Appl. Intel. 48 (3) (2018) 570–593.
[10] L. Huang, C.D. Wang, H.Y. Chao, A harmonic motif modularity approach for multi-layer network community detection, 2018 IEEE International Conference on Data Mining, (ICDM 2018), 2018, doi:10.1109/ICDM.2018.00132.
[11] S. Jendoubi, A. Martin, L. Liétard, H.B. Hadji, B.B. Yaghlane, Maximizing positive opinion influence using an evidential approach, in: Proceedings of FLINS, Aug 2016, Roubaix, France, 2016, pp. 168–174.
[12] D. Kempe, J. Kleinberg, E. Tardos, Influential nodes in a diffusion model for social networks, in: Proceedings of the 32nd International Conference on Automata, Languages and Programming, 3580, 2005, pp. 1127–1138.
[13] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 137–146.
[14] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, 2006, pp. 259–271.
[15] J. Leskovec. Epinions social network. http://snap.stanford.edu/data/soc-Epinions1.html.
[16] J. Leskovec. Wikipedia vote network. http://snap.stanford.edu/data/wiki-Vote.html.
[17] J. Leskovec, J. Kleinberg, C.Graph evolution Faloutsos, densification and shrinking diameters, ACM Trans. Knowl. Disc. Data (ACM TKDD) 1 (1) (2007) 2.
[18] M. Hosseini-Pozveh, K. Zamanifar, A.R. Naghsh-Nilchi, Assessing information diffusion models for influence maximization in signed social networks, Expert Syst Appl 119 (1) (2019) 476–490.
[19] D. Li, C.H. Wang, S.P. Zhang, G.L. Zhou, D.H. Chu, C. Wu, Positive influence maximization in signed social networks based on simulated annealing, Neurocomputing 260 (2017) 69–78.
[20] J.H. Li, C.D. Wang, P.Z. Li, J.H. Lai, Discriminative metric learning for multi-view graph partitioning, Pattern. Recognit 75 (2018) 199–213.
[21] H.J. Li, L. Pan, P. Wu, Dominated competitive influence maximization with time-critical and time-delayed diffusion in social networks, J. Comput. Sci 28 (2018) 18–327.
[22] W.X. Liang, Ch.G. Shen, X. Li, R. Nishide, I. Piumarta, H. Takada, Influence maximization in signed social networks with opinion formation, IEEE Access 7 (2019) 68837–68852.
[23] W. Liu, X. Chen, B. Jeon, L. Chen, B.L. Chen, Influence maximization on signed networks under independent cascade model, Appl. Intelligence 49 (3) (2019) 912–928.
[24] C.V. Pham, D.K. Ha, D.Q. Ngo, Q.g C. Vu, H.X. Hoang, A new viral marketing strategy with the competition in the large-scale online social networks, in: Proceedings of the IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, 2016, pp. 1–6.
[25] C.V. Pham, H.M. Dinh, H.D. Nguyen, H.T. Dang, H.X. Hoang, Limiting the spread of epidemics within time constraint on online social networks, in: Proceedings of the Eighth International Symposium on Information and Communication Technology(SoICT 2017), 2017, pp. 262–269.
[26] C.V. Pham, M.T. Thai, D. Ha, D.Q. Ngo, H.X. Hoang, Time-Critical viral marketing strategy with the competition on online social networks, in: Proceedings of the 5th International Conference on Computational Social Networks (CSoNet 2016), LNCS 9795, 2016, pp. 111–122.
[27] A. Sela, D. Goldenberg, I. Ben-Gal, E. Shmueli, Active viral marketing: incorporating continuous active seeding efforts into the diffusion model, Expert Syst. Appl 107 (1) (2018) 45–60.
[28] C.G. Shen, R. Nishide, I. Piumarta, H. Takada, W.X. Liang, Influence maximization in signed social networks, in: Proceedings of the International Conference on Web Information Systems Engineering (WISE), 2015, 2015, pp. 399–414.
[29] P.G. Sun, Y.N. Quan, Q.G. Miao, J. Chi, Identifying influential genes in protein–protein interaction networks, Inf. Sci. (Ny) 454–455 (July 2018) 229–241.
[30] A. Talukder, M.G.R. Alam, N.H. Tran, D. Niyato, C.S. Hong, Knapsack-Based reverse influence maximization for target marketing in social networks, IEEE Access 7 (2019) 44182–44198.
[31] K. Tanınmış, N. Aras, I.K. Altınel, Influence maximization with deactivation in social networks, Eur. J. Oper. Res. 278 (1) (2019) 105–119.
[32] C.W. Tsai, S.J. Liu, SEIM: search economics for influence maximization in online social networks, Future Gener. Comp. Sys. 93 (2019) 1055–1064.
[33] F. Wang, W.J. Jiang, X.L. Li, G.J. Wang, Maximizing positive influence spread in online social networks via fluid dynamics, Future Gener. Comp. Sys. 86 (2018) 1491–1502.
[34] W. Wei, Y. Liu, R. Zhang, SPLMax: exploiting the simple path introduced locality for maximum flow acceleration, IEEE Comm. Letters 22 (7) (2018) 1330–1333.
[35] Y.M. Wen, C.D. Wang, K.Y. Lin, Direction recovery in undirected social networks based on community structure and popularity, Inf. Sci. (NY) 473 (2019) 31–43.
[36] X. Weng, Z.B. Liu, Z.Y. Li, An efficient influence maximization algorithm considering both positive and negative relationships, in: Proceedings of the IEEE Trustcom/BigDataSE/ISPA, 2016, 2016, pp. 1931–1936.
[37] H.-H. Wu, S. Kucukyavuz, A two-stage stochastic programming approach for influence maximization in social networks, Comput Optim Appl 69 (3) (2018) 563–595.
[38] D.D. Yang, X.W. Liao, H.W. Shen, X.Q. Cheng, G.L. Chen, Relative influence maximization in competitive social networks, SCIENCE CHINA Information Sciences 60 (10) (2017) 108101.
[39] X.C. Yu, R. Li, T.G. Chu, IEEE access, Effects of Network Structure on Inform. Diffusion Reconstruction 7 (2019) 54834–54842.
[40] Y.H. Zhang, T.N. Dinh, M.T. Thai, Maximizing the spread of positive influence in online social networks, in: Proceedings of the IEEE Int. Conference on Distributed Computing Systems (ICDCS), 2013, pp. 317–326.
[41] Z. Zhang, Y.S. Shi, J. Willson, D.Z. Du, G.M. Tong, Viral marketing with positive influence, in: Proceedings of IEEE Conference on Computer Communications(INFOCOM), 2017, pp. 1–8.
[42] L.H. Zhu, X. Zhou, Y.M. Li, Global dynamics analysis and control of a rumor spreading model in online social networks, Physica A: Stat. Mech. and its Appl. 526 (2019) Article 120903.

Please cite this article as: W. Ju, L. Chen and B. Li et al., A new algorithm for positive influence maximization in signed networks, Information Sciences, https://doi.org/10.1016/j.ins.2019.10.061