

Individual Influence Maximization via Link Recommendation

Guowei Ma¹, Qi Liu^{1(✉)}, Enhong Chen¹, and Biao Xiang²

¹ University of Science and Technology of China, Hefei, Anhui, China
gwma@mail.ustc.edu.cn, {qiliuql, cheneh}@ustc.edu.cn

² Microsoft, Search Technology Center Asia (STCA), Beijing, China
bixian@microsoft.com

Abstract. Recent years have witnessed the increasing interest in exploiting social influence in social networks for many applications, such as viral marketing. Most of the existing research focused on identifying a subset of influential individuals with the maximum influence spread. However, in the real-world scenarios, many individuals also care about the influence of herself and want to improve it. In this paper, we consider such a problem that maximizing a target individual's influence by recommending new links. Specifically, if a given individual/node makes new links with our recommended nodes then she will get the maximum influence gain. Along this line, we formulate this link recommendation problem as an optimization problem and propose the corresponding objective function. As it is intractable to obtain the optimal solution, we propose greedy algorithms with a performance guarantee by exploiting the submodular property. Furthermore, we study the optimization problem under a specific influence propagation model (i.e., Linear model) and propose a much faster algorithm (*uBound*), which can handle large scale networks without sacrificing accuracy. Finally, the experimental results validate the effectiveness and efficiency of our proposed algorithms.

1 Introduction

Social network platforms, such as Twitter and Facebook, play an important and fundamental role for the spread of influence, information, or innovations. These diffusion processes are useful in a number of real-world applications, for instance, the social influence propagation phenomenon could be exploited for better viral marketing [1]. To this end, both modeling the influence propagation process and identifying the influential individuals/nodes in social networks have been hot topics in recent years [2].

Indeed, researchers have proposed several influence models to describe the dynamic of influence propagation process, such as Independent Cascade (IC) model [3], Linear Threshold (LT) model [4], a stochastic information flow model [5] and the linear social influence model (Linear) [6]. Meanwhile, other researchers focus on learning the real or reasonable influence propagation probability between two individuals in the influence models [7, 8]. Based on the influence propagation models and the influence propagation probabilities, influence

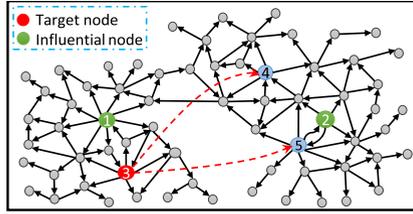


Fig. 1. A toy example

maximization (IM) is the problem of identifying a subset with K influential nodes in a social network so that their aggregated influence in the network is maximized. Since influence maximization is a fundamental problem in viral marketing, various aspects of it have been studied extensively in the last decade [2, 9, 10]. For instance, Eftekhari et al. [10] studied influence propagation at group scale, where they aimed at identifying the influential groups instead of a subset of individuals.

However, most of existing works about IM focus on identifying a small subset of individuals or groups in a network so that their aggregated influence spread is maximized. In the real-world scenarios, an individual also cares about her own influence and wants to improve it by making new links. Formally, if a given target node (e.g., a person or a company) in a social network wants to maximize its influence by making several new links (i.e., target node could spread its influence through these links), which nodes should it link with? In this situation, linking it with the most influential nodes or the nodes with largest degree, may not lead to the maximum influence gain, since we have to consider the topology of the target node and the overlap of influence spread between the target node and the selected nodes. Let us take the network in Fig.1 as a toy example. If node 3 is the given target node and we want to improve its influence by recommending two new links for node 3. Suppose that node set $\{1, 2\}$ are the most influential node set found by IM method (e.g, by CELF [11]). Actually, the total influence of node 3 after linking with nodes $\{1, 2\}$ is less than that with nodes $\{4, 5\}$. The reason may be that there is much overlap of the influence spread between nodes 3 and 1.

Though similar link recommendation problems have been studied in the literature (e.g., adding new links or strengthening the weakened social links to boost the information spread across the entire network [12]), the problem of eliminating the influence overlap to maximize the target node's influence via link recommendation remains pretty much open. Actually, there are two challenges to solve this problem efficiently: First, how to design a rational measure to eliminate the influence overlap between nodes; Second, because the computation of influence spread is very time-consuming, it is urgent to propose an efficient algorithm which can sharply reduce the times of influence spread estimations. To address these challenges, in this paper, we provide a focused study on the problem of maximizing the target individual's own influence by recommending

new links for this individual (i.e., individual influence maximization via link recommendation). Our contributions could be summarized as follows:

- We formulate this individual influence maximization-oriented link recommendation problem as an optimization problem, and define the corresponding objective function, which can be generally applied to different influence propagation models.
- To solve the intractable optimization problem effectively, we propose a *greedy* algorithm with a performance guarantee. One step further, we present another algorithm *lazy* for scaling up this simple *greedy*. Both algorithms can be used in general influence models, such as IC and LT.
- We leverage the properties of influence spread estimations under the specific Linear model, and propose a much faster recommendation algorithm *uBound*, which can handle large scale networks without sacrificing accuracy.
- We conduct extensive experiments on four real world datasets and the results demonstrate the effectiveness and efficiency of our proposed algorithms.

2 Related Work

Influence Propagation and Maximization. Researches proposed several models for describing the influence propagation process. IC model [3] and LT model [4] are two widely used ones. However, both of them require time-consuming Monte Carlo simulations to estimate the influence spread, some researchers designed more efficient (or tractable) models, e.g., the stochastic information flow model [5] and the linear social influence (Linear) model [6]. Since learning the influence propagation process is beyond the scope of this paper, we use these existing influence models for illustration.

The IM problem can be traced back to Domingos and Richardson [1, 13]. Kempe et al. [14] first formulated it as a discrete optimization problem, demonstrated it as NP-hard and presented a greedy approximation algorithm with provable performance guarantee. From then on, researchers proposed many computationally efficient algorithms, such as CELF [11], PMIA [15], IPA [16] and TIM [17], by exploiting specific aspects of the graph structure or the influence model. Some researchers also consider other aspects of the IM problem [9, 18]. For instance, Guo et al. [18] studied local influence maximization, aiming to find the top- K local influential nodes on the target node. However, to the best of our knowledge, few attention has been paid to the problem of maximizing the target node’s own influence via link recommendation.

Recommendations in Social Networks. The user-to-user recommendation in social networks is an important task for many social network sites like Twitter, Google+ and Facebook, for the purpose of guiding user discover potential friends [19–21] or improving the connectivity of the network [12, 22]. Researchers have proposed a number of recommendation algorithms to recommend potential friends to users in a social platform, such as the Friend-of-Friend(FoF) algorithm [19] and other interest-based or profile-based algorithms [20, 21]. Some of

these works also consider the influence propagation effect, such as selecting a set of “influential” users for a new user [23] or a new product [24], like solving the cold-start problem in recommender systems.

In addition, some works in the area of *network/graph augmentation* also try to add links in the network for improving some quality of the graph [12, 22]. For instance, Tian et al. [22] suggested users to re-connect their old friends and strengthen the existing weak social ties in order to improve the social network connectivity; Chaoji et al. [12] recommended an edge set in order to increase the connectivity among users and boost the content spread in the entire social network. However, these related works pay more attention to the entire social network rather than the target individual’s own influence.

3 Individual Influence Maximization

Preliminaries. Let the directed graph $G(V, E, T)$ represents an influence network, where $V = \{1, 2, \dots, n\}$ are n nodes in graph and E stores all the influence links(edges) between nodes. $T = [t_{ij}]_{n \times n}$ is a given propagation probability matrix. For each edge $(j, i) \in E$, $t_{ij} \in (0, 1)$ denotes the influence propagation probability from node i to node j . For any edge $(j, i) \notin E$, $t_{ij} = 0$. G is assumed to be directed as influence propagation is directed in the most general case. Given this graph, the influence spread \mathbf{f}_i for each node $i \in V$ can be computed by the influence propagation models (e.g., IC [3], LT [4] and Linear [6]). Specifically, $\mathbf{f}_i = [f_{i \rightarrow 1}, f_{i \rightarrow 2}, \dots, f_{i \rightarrow n}]'$, an $n \times 1$ vector, denotes the influence of node i on each node in the network. Thus, the total influence spread of node i in network equals to the sum of influence of node i on other nodes, namely $f_{i \rightarrow V} = \sum_{j \in V} f_{i \rightarrow j}$. Indeed, $f_{i \rightarrow V}$ is the expected number of the nodes that will be influenced by node i .

3.1 Problem Statement and Formulation

In a real-world network, such as Twitter, nodes represent users, and edges represent their links/connections. If a target user wants to improve her own influence, she should make new influence links¹ with other users, especially the influential ones, then the information she posts will be read and followed by more users (e.g., by retweet). Since making new links with other nodes may require money or time, we also associate a nonnegative cost $c(j)$ with each node j . That is, the cost of linking to node j is $c(j)$ if a target node makes a new link with j . The less the cost is, the easier to create the link for the target node. We denote the total cost of the target node for making new links with a subset \mathbf{S} as $c(\mathbf{S}) = \sum_{j \in \mathbf{S}} c(j)$. Hence, the problem of individual influence maximization is to find a subset \mathbf{S} such that if the target node t makes new links with nodes in

¹ In this paper, we use the expressions of “influence link ”and “link” without distinction.

\mathbf{S} , the t 's influence gain is maximum, and $c(\mathbf{S})$ does not exceed a specific budget B . Now this problem could be formulated as an optimization problem:

$$\arg \max_{\mathbf{S}} \{f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V}\}, \text{ subject to } c(\mathbf{S}) \leq B, \quad (1)$$

where $f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V}$ is the influence gain of the target node t after linking with nodes in set \mathbf{S} . Notice that, we assume that the other parts of network structure stay unchanged before t makes links with the nodes in \mathbf{S} . To reduce complexity, in this paper, we consider $c(j) = 1$ for each $j \in V$, i.e., every new link shares the same cost. Hence, the cost $c(\mathbf{S})$ equals to the number of nodes in \mathbf{S} , namely $c(\mathbf{S}) = |\mathbf{S}|$. Let $\mathcal{F}(\mathbf{S}) = \{f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V}\}$ and $K = B$, we can rewrite Eq. (1) as below.

$$\arg \max_{\mathbf{S}} \mathcal{F}(\mathbf{S}) = \{f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V}\}, \text{ s.t. } |\mathbf{S}| \leq K. \quad (2)$$

In summary, the individual influence maximization problem is formalized as recommending a subset \mathbf{S} with K nodes such that node t can achieve the maximum influence gain by making new links with the nodes in \mathbf{S} (i.e., adding new edges $(j, t), j \in \mathbf{S}$).

3.2 Definition of the Objective Function

The key of the above optimization problem is to design an appropriate objective function $\mathcal{F}(\mathbf{S})$ to eliminate the influence overlap (the first challenge given in Introduction) when adding \mathbf{S} to link the target individual. For introducing our definition of $\mathcal{F}(\mathbf{S})$, we start with a single link from node t to node c .

Definition 1. *If a target node t makes a new link with a candidate node c , we define $\mathcal{F}(\mathbf{S}) = \mathcal{F}(\{c\})$ as :*

$$\begin{aligned} \mathcal{F}(\{c\}) &= f_{t \rightarrow V}^{\{c\}} - f_{t \rightarrow V} \\ &= \lambda_c \cdot (1 - f_{t \rightarrow c}) \cdot \sum_{i \in V} (1 - f_{t \rightarrow i}) f_{c \rightarrow i}, \end{aligned}$$

where $\lambda_c \in (0, 1)$ is a hyper parameter to model the real-world social influence propagation process.

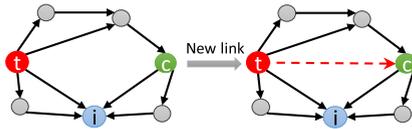


Fig. 2. A Simple Example

Definition Explanation. Let us take a simple example. Suppose we want to improve the target node t 's influence in Fig. 2, and thus we should estimate

the total influence gain of t after making a new link (the red dashed line) with a candidate node c . We first show how to estimate the influence gain of t on any node $i \in V$. Before making the new link with c (Left part of Fig. 2), the target t has an influence on node i ($f_{t \rightarrow i}$) and node c ($f_{t \rightarrow c}$) respectively, and meanwhile, the node c also has an influence on node i ($f_{c \rightarrow i}$). When t makes a new link with c (Right part of Fig. 2), we define that the influence of t on node c has increased by $\lambda_c(1 - f_{t \rightarrow c})$. Now, let's explain this definition: Suppose t always influences (or activates) c successfully ($f_{t \rightarrow c}^{\{c\}} = 1$), then the influence of t on node c will be increased by $(f_{t \rightarrow c}^{\{c\}} - f_{t \rightarrow c}) = (1 - f_{t \rightarrow c})$. However, this assumption is a little unrealistic. Thus, for better modeling the real-world influence propagation process, we introduce $\lambda_c \in (0, 1)$ to weaken the influence gain of node t on c , and get $\lambda_c(1 - f_{t \rightarrow c})$.² One step further, we can represent that the influence of t on i through c has increased by $\lambda_c(1 - f_{t \rightarrow c})f_{c \rightarrow i}$. Hence, the influence gain of t on i after making a link with c is: $f_{t \rightarrow i}^{\{c\}} - f_{t \rightarrow i} = (1 - f_{t \rightarrow i}) \cdot \lambda_c \cdot (1 - f_{t \rightarrow c}) \cdot f_{c \rightarrow i}$. Then, we can get the total influence gain of node t on the entire network: $f_{t \rightarrow V}^{\{c\}} - f_{t \rightarrow V} = \sum_{i \in V} (f_{t \rightarrow i}^{\{c\}} - f_{t \rightarrow i}) = \sum_{i \in V} (1 - f_{t \rightarrow i}) \cdot \lambda_c(1 - f_{t \rightarrow c})f_{c \rightarrow i} = \lambda_c(1 - f_{t \rightarrow c}) \sum_{i \in V} (1 - f_{t \rightarrow i})f_{c \rightarrow i}$.

From this example, we could get the implication of the Definition 1. Though we do not show more rigorous justification for this function, the extensive experimental results show that the nodes selected by this function can really obtain much real influence gain of a given target node, which illustrate that this function is rational and effective.

One step further, we introduce the following definition of the objective function $\mathcal{F}(\mathbf{S})$, i.e., the influence gain of a target node when it makes new links with nodes in \mathbf{S} .

Definition 2. *If a target node t makes new links with the nodes in \mathbf{S} , we define $\mathcal{F}(\mathbf{S})$:*

$$\begin{aligned} \mathcal{F}(\mathbf{S}) &= f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V} = \sum_{c \in \mathbf{S}} \mathcal{F}(\{c\}) \\ &= \sum_{c \in \mathbf{S}} \lambda_c(1 - f_{t \rightarrow c}) \cdot \sum_{i=1}^n (1 - f_{t \rightarrow i})f_{c \rightarrow i} \end{aligned}$$

We could demonstrate that the function $\mathcal{F}(\mathbf{S})$ satisfies the properties below:

1. $\mathcal{F}(\emptyset) = 0$, i.e., we cannot improve the influence of the target node without making any new link.
2. $\mathcal{F}(\mathbf{S})$ is nonnegative and monotonically increasing. It is obvious that making new links can not reduce the influence of the target node.
3. $\mathcal{F}(\mathbf{S})$ is *submodular*. That is, $\mathcal{F}(\mathbf{S})$ satisfies the ‘‘diminishing returns’’ property.

² Notice that, in real-world applications, λ_c 's value could be determined based on the specific influence models and the characteristics of nodes c and t in the social network.

3.3 Greedy Strategy

Indeed, maximizing submodular function in general is NP-hard [25], and thus it is intractable to obtain the optimal solution of the problem we formulated. However, for a nonnegative monotone submodular function, such as $\mathcal{F}(\mathbf{S})$, the greedy strategy, a common used heuristic, approximates the optimum within a factor of $(1 - 1/e)$ [26].

The simple *greedy* algorithm starts with the empty set $\mathbf{S} = \emptyset$, and requires about n times influence spread estimation in each iteration to select one node (with the maximum influence margin) to join \mathbf{S} . Thus, *greedy* requires about $(n \cdot K)$ times influence spread estimations, where $K = |S|$. As each influence spread estimation calculated by influence models (e.g., IC, LT) is very time-consuming, *greedy* is quite slow.

Scaling Up. Here, we exploit the submodularity of $\mathcal{F}(\mathbf{S})$ and adopt the lazy-forward strategy [11] for scaling up the simple algorithm *greedy*. Specifically, based on the fact that the influence gain of node t after making a link with node c in the current iteration cannot be larger than its marginal influence gain in the previous iteration, we propose the algorithm *lazy* without sacrificing any accuracy. Algorithm 1 shows the details about the algorithm *lazy*. Because *lazy* just requires n times influence spread estimations in the initial iteration for calculating the upper bound of influence gain of each candidate node i , it requires totally $(n + \theta \cdot K)$ times influence spread estimations, where $\theta \ll n$ is the expected number of influence spread estimations in each iteration.

Algorithm 1. The *lazy* Algorithm

<p>Input: $G(V, E, T)$, a given target node t, K</p> <p>Output: \mathbf{S} with K nodes</p> <p>1 initialize $\mathbf{S} = \emptyset$;</p> <p>2 for each node $i \in V$ do</p> <p>3 calculate $\mathcal{F}(\{i\}) = f_{t \rightarrow V}^{\{i\}} - f_{t \rightarrow V}$;</p> <p>4 $flag_i = \mathbf{S}$; // here, $\mathbf{S} = 0$</p> <p>5 // $flag_i$ indicates that $\mathcal{F}(\{i\})$ is</p> <p>6 // calculated in the \mathbf{S} iteration</p>	<p>7 while $\mathbf{S} < K$ do</p> <p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p>	<p>$s = \text{Find the greatest } \mathcal{F}(\{s\}) \text{ in } \mathcal{F}$;</p> <p>if $flag_s == \mathbf{S}$ then</p> <p> $\mathbf{S} = \mathbf{S} \cup s$;</p> <p> $\mathcal{F}(s) = 0$;</p> <p>else</p> <p> recalculate $\mathcal{F}(s) = f_{t \rightarrow V}^{\mathbf{S} \cup s} - f_{t \rightarrow V}^{\mathbf{S}}$;</p> <p> $flag_s = \mathbf{S}$;</p> <p>Return \mathbf{S};</p>
--	--	---

4 Optimization Under the Linear Model

To address the challenge of inefficiency, we further explore this problem on a specific influence model, the linear social influence (Linear) model [6]. Specifically, the reasons could be summarized as: (1) Linear model is tractable and efficient; (2) Linear has close relations with the traditional influence models. For instance, it can approximate the non-linear stochastic model [5], and the linear

approximation method for the IC model [27] is a special case of Linear. In the following, we first review the Linear model.

Review. Given a directed graph $G(V, E, T)$, Linear model [6] is defined as below.

Definition 3. Define the influence of node i on j as

$$f_{i \rightarrow i} = \alpha_i, \quad \alpha_i \in (0, 1] \quad (3)$$

$$f_{i \rightarrow j} = d_j \sum_{k \in N_j} t_{kj} f_{i \rightarrow k}, \quad \text{for } j \neq i \quad (4)$$

where $N_j = \{u \in V | (u, j) \in E\}$, α_i is the self-confidence of node i , which represents the prior constraint of node i for spreading the information. The parameter $d_j \in (0, 1]$ is the damping coefficient for the influence propagation.

Under the Linear model, there is an upper bound to measure a node's influence [6]:

$$f_{i \rightarrow v} = \sum_{j=1}^n f_{i \rightarrow j} \leq \alpha_i \cdot (I - DT)_i^{-1} \mathbf{e} \quad (5)$$

where, I is an n -by- n identity matrix, $D = \text{diag}(d_1, d_2, \dots, d_n)$ is a diagonal matrix, \mathbf{e} is an $n \times 1$ vector consisting all 1s, $(I - DT)_i^{-1}$ denotes the i -th row of matrix $(I - DT)^{-1}$.

Optimization with Upper Bounds. In this part, we further exploit the properties of the influence computation in Linear model and demonstrate that if target node t makes a new link with an arbitrary candidate node i , the influence gain cannot be greater than the upper bound, $\lambda_i \alpha_i \cdot (I - DT)_i^{-1} \mathbf{e}$.

Theorem 1. (Upper bound) If a given target node t makes a new link with node $i \in (V \setminus \{t\})$, then the influence gain of node t satisfies the equation:

$$\mathcal{F}(\{i\}) = f_{t \rightarrow v}^{\{i\}} - f_{t \rightarrow v} \leq \lambda_i \cdot \alpha_i \cdot (I - DT)_i^{-1} \mathbf{e}$$

Proof. We first prove that $\mathcal{F}(\{i\}) = f_{t \rightarrow v}^{\{i\}} - f_{t \rightarrow v} \leq \lambda_i \cdot f_{i \rightarrow v}$. According to the influence gain definition,

$$\mathcal{F}(\{i\}) = f_{t \rightarrow v}^{\{i\}} - f_{t \rightarrow v} \quad (6)$$

$$= \lambda_i (1 - f_{t \rightarrow i}) \sum_{k=1}^n (f_{i \rightarrow k} \cdot [1 - f_{t \rightarrow k}]) \quad (7)$$

$$\leq \lambda_i (1 - f_{t \rightarrow i}) \cdot \sum_{k=1}^n f_{i \rightarrow k} \quad (8)$$

$$\leq \lambda_i \sum_{k=1}^n f_{i \rightarrow k} = \lambda_i \cdot f_{i \rightarrow v} \quad (9)$$

Both Eqs. (8) and (9) hold because $f_{i \rightarrow j} \in [0, 1]$. Combining Eqs. (5) with (9), we have proved that $\mathcal{F}(\{i\}) = f_{t \rightarrow v}^{\{i\}} - f_{t \rightarrow v} \leq \lambda_i \alpha_i (I - DT)_i^{-1} \mathbf{e}$. \square

Here, we can rewrite Theorem 1 into vector: $[\mathcal{F}(\{1\}), \mathcal{F}(\{2\}), \dots, \mathcal{F}(\{n\})]' \leq \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \cdot \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n) \cdot (I - DT)^{-1} \mathbf{e}$. As $(I - DT)$ is a strictly diagonally dominant matrix, $(I - DT)^{-1} \mathbf{e}$ can be quickly calculated through

Gauss-Seidel method in $O(|E|)$ time. We use these upper bounds to replace the influence gain estimations in the initial iteration of algorithm *lazy*, and then propose the corresponding *uBound* algorithm without sacrificing any accuracy. Algorithm 2 shows the details about *uBound*. According to the analysis above, we know that *uBound* requires only $(1+\eta \cdot K)$ times influence spread estimations, where $\eta \ll n$ is the expected number of influence spread estimations and it is related to the tightness of the upper bound. In contrast, *lazy* (Algorithm 1) requires $(n + \theta \cdot K)$ times influence spread estimations.

Algorithm 2. The *uBound* Algorithm

```

Input:  $G(V, E, T)$ , a given target node  $t$ ,  $\epsilon$ 
       $K$ 
Output:  $\mathbf{S}$  with  $K$  nodes
1 initialize  $\mathbf{S} = \emptyset$ ;
2 Compute the upper bound vector
   $\mathbb{U} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \cdot$ 
   $\text{diag}(\alpha_1, \alpha_2, \dots, \alpha_n) \cdot (I - DT)^{-1} \cdot \mathbf{e}$  in
   $O(|E|)$  time;
3 for each node  $i \in V$  do
4    $\mathcal{F}(i) = \mathbb{U}_i$ ;
5    $flag_i = 0$ ; // here,  $|\mathbf{S}| = 0$ 
6 while  $|\mathbf{S}| < K$  do
7    $s = \text{Find the greatest } \mathcal{F}(s) \text{ in } \mathcal{F}$ ;
8   if  $flag_s == |\mathbf{S}|$  then
9      $\mathbf{S} = \mathbf{S} \cup s$ ;
10     $\mathcal{F}(s) = 0$ ;
11  else
12    recalculate  $\mathcal{F}(s) = f_{t \rightarrow V}^{\mathbf{S} \cup s} - f_{t \rightarrow V}^{\mathbf{S}}$ ;
13     $flag_s = |\mathbf{S}|$ ;
14 Return  $\mathbf{S}$ ;

```

5 Experiments

Experimental Setup. The experiments are conducted on four real-world datasets with different sizes. (a)Wiki-Vote, a who-votes-on-whom network at Wikipedia where nodes are users and an edge(j, i) represents that user j voted on user i ; (b)Weibo, a social media network in China, where nodes are the users and edges are their followships. We crawled this data from weibo.com³ at March 2013 and then sampled a small network which only contains the verified users for filtering the zombomb accounts; (c)Cit-HepPh, an Arxiv High Energy Physics paper citation network where nodes represent papers and an edge(j, i) represents that paper j cites paper i . Both Cit-HepPh and Wiki-Vote are downloaded from SNAP⁴; (d)Twitter, another social media network. We downloaded this data from Social Computing Data Repository at ASU⁵. Table 1 show the detailed dataset information.

Table 1. Experimental Datasets

Name	Wiki-Vote	Weibo	cit-HepPh	Twitter
Nodes	7,115	7,378	34,546	11,316,811
Edges	103,689	39,759	421,578	85,331,845

³ <http://www.weibo.com/>

⁴ <http://snap.stanford.edu/data/>

⁵ <http://socialcomputing.asu.edu/datasets/Twitter>

Influence Model and Propagation Probability. We validate our discoveries under the IC, LT and Linear models, as widely used in the literature [6, 11, 14, 18, 28]. For each network, we transform it into a directed influence graph $G(V, E, T)$. Specifically, if there is an edge (j, i) in the original network, we add an influence link $(i, j) \in E$ ⁶ in G and then assign the corresponding influence propagation probability $t_{ij} = 1/\text{indegree}(j)$. For LT [14], each node j chooses a *threshold* θ_j uniformly at random from $[0, 1]$, and the Monte Carlo simulation times are set to be 20,000 for both IC and LT. For Linear model, we use the same damping coefficient for all nodes similar to Xiang et al. [6, 28] (i.e., $d_i = 0.85$ for $i \in V$), and we set $\alpha_i = 1$ assuming that each initial node shares the same prior influence probability. Note that, for simplicity, we manually set the $\lambda_c = 0.85$ for all the nodes in objective function $\mathcal{F}(\mathbf{S})$.

We implemented the algorithms in Java and conducted the following experiments on Windows 64-bit OS with 2.20GHz Intel Core i3-2330M and 16GB memory.

5.1 Real Influence Gain Comparison

We first demonstrate that our objective function is rational and effective, i.e., the node set \mathbf{S} recommended based on our $\mathcal{F}(\mathbf{S})$ can help a target node t make more influence gain than the benchmark methods. Specifically, for a given target node t , we first calculate its original influence $f_{t \rightarrow V}$. Secondly, we let t make new links with the nodes in \mathbf{S} recommended by different methods, and recalculate the t 's new influence $f_{t \rightarrow V}^{\mathbf{S}}$ ⁷. Finally, we get the t 's real influence gain $\mathcal{F}(\mathbf{S}) = f_{t \rightarrow V}^{\mathbf{S}} - f_{t \rightarrow V}$. Thus, the performance of each method is evaluated by the influence gain it could provide to the target node, i.e., the larger influence gain, the better the method is. In the following, we call our method as ISIM (Individual Social Influence Maximization) and the results are based on the *lazy* algorithm. For comparison, we choose several benchmark methods:

- **Random.** Let the target node make links with K nodes that are selected randomly.
- **OutDeg.** Let the target node link to the top K nodes with the largest out-degree.
- **LongDist.** The recommended K nodes are the farthest ones from the target node, i.e., those have the fewest influence overlap with the target node. Here, the distance is measured by Random Walk with Restart [29].
- **PageRank.** Recommend the nodes with top K ranked PageRank values [30].
- **HighestInf (Highest Influence).** Let the target node make links with the top- K nodes with highest influence. This method is also competitive because the largest influential nodes can improve the target node's influence sharply. However, this method does not consider the influence overlap.

⁶ For example, if user j follows user i in Twitter, then i influences j .

⁷ After linking t to nodes in \mathbf{S} , the indegree of node $c \in \mathbf{S}$ pluses one. The influence propagation probability of each edge $(u, c) \in E$ will be updated by $1/\text{new.indegree}(c)$.

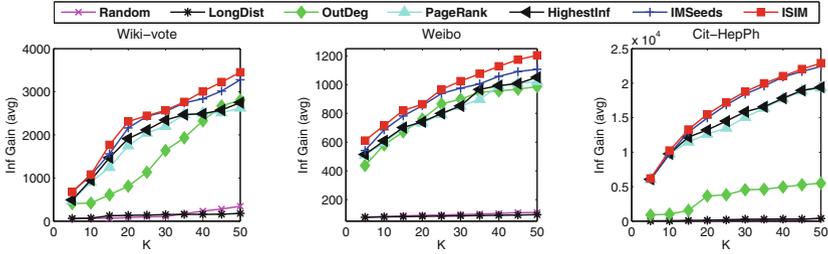


Fig. 3. Comparisons under IC Model

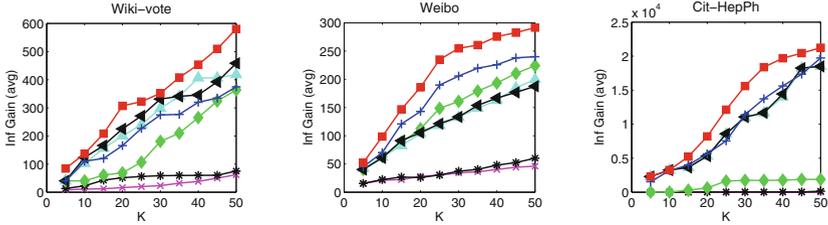


Fig. 4. Comparisons under LT Model

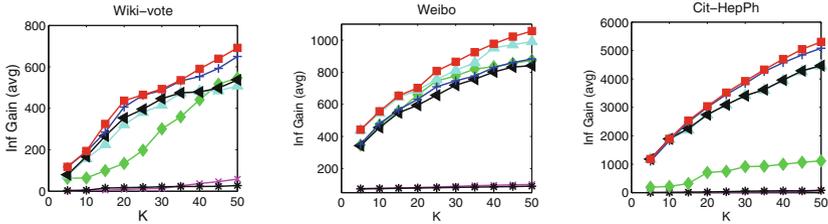


Fig. 5. Comparisons under Linear Model

- **IMSeeds**. It selects and recommends the most influential node set \mathbf{S} by using the CELF algorithm [11] for traditional social influence maximization problem. This method could alleviate the influence overlap between the nodes in \mathbf{S} . However, it does not consider the influence overlap between the target node and those in \mathbf{S} .

On each dataset, we run the above selection algorithms on the randomly chosen 100 target nodes from different out-degree ranges, and then we compute and compare the average influence gain (with the size of the recommendation set $|\mathbf{S}| = 5, 10, \dots, 50$) for each algorithm. We compare the effectiveness of each algorithm under IC, LT and Linear model, respectively. Figs. 3, 4 and 5 show the corresponding results. Actually, similar results could be seen from all figures. That is, the node set \mathbf{S} selected by our ISIM could help the target node to get more real influence gain than the benchmarks; the node set \mathbf{S} recommended by IMSeeds cannot always guarantee the best performance. What's more, we only

show the results on the three data sets, because IMSeeds (i.e., CELF) cannot obtain a result within feasible time on the Twitter data.

5.2 Time Complexity Analysis

In the part, we compare the efficiency of our proposed algorithms (*greedy*, *lazy* and *uBound*) for our ISIM method under Linear model from two aspects: *the number of influence spread estimations* and the *running time* of the algorithms.

Table 2. Numbers of Influence Spread Estimation

Datasets	Top K									
	Alg.	5	10	15	20	25	30	35	40	45
Wiki-Vote	<i>greedy</i>	35208	70390	105547	140679	175786	210868	245925	280957	315964
	<i>lazy</i>	7124	7139	7171	7195	7220	7250	7306	7331	7374
	<i>uBound</i>	19	53	84	119	157	204	265	297	352
Weibo	<i>greedy</i>	36836	73646	110431	147191	183926	220636	257321	293981	330616
	<i>lazy</i>	7414	7446	7505	7572	7638	7738	7838	7917	8047
	<i>uBound</i>	50	98	180	278	356	494	622	726	868
Cit-HepPh	<i>lazy</i>	34554	34566	34586	34608	34642	34660	34691	34710	34744
	<i>uBound</i>	17	37	64	89	137	161	194	221	264
Twitter	<i>uBound</i>	16	43	71	99	136	166	212	253	286

Table 2 shows the expected numbers of influence spread estimations when selecting different K seeds using different algorithms. The results illustrate that *greedy* needs the largest number of influence spread estimations. Compared to *lazy*, the expected number of influence spread estimations of *uBound* at top $K = 45$ is reduced at a rate of 95.2%, 89.2%, 99.2% on the three datasets (ie, Wiki-Vote, Weibo, Cit-HepPh), respectively. The reason is that *lazy* requires n times influence spread estimations in the initial iteration to establish the upper bounds of the marginal influence, while *uBound* requires only one time. Correspondingly, Fig. 6 shows the real running time of different algorithms when selecting K seeds on different datasets. From the results, we know that the simple *greedy* algorithm is very time-consuming as the number K increases. That is because *greedy* requires about $(n \cdot K)$ times influence spread estimations. What's more, we can observe that *uBound* is much faster than *lazy*. Actually, *uBound* is so efficient that it can handle the Twitter data, a large scale network with tens of millions of nodes, and the running time is growing linearly as the the number K increases.

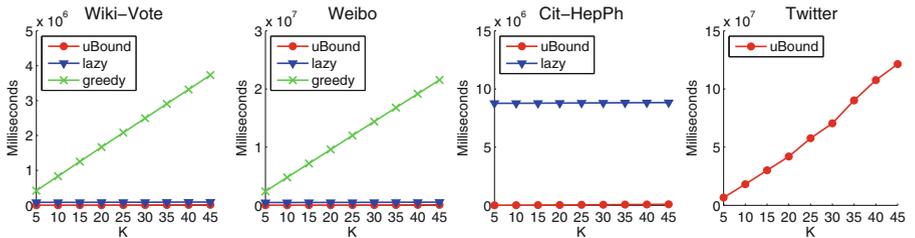


Fig. 6. Real Runtime Comparisons

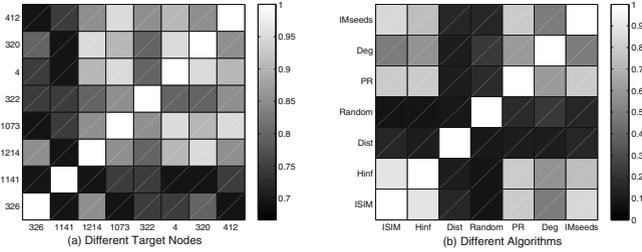


Fig. 7. Jaccard Index Comparison

5.3 Case Study

We finally use a case study to illustrate the necessity of designing individualized link recommendation algorithms. Fig. 7(a) shows the Jaccard index of the 25 nodes recommended by ISIM for 8 target nodes which are randomly selected (distinguished by node ID) from Wiki-Vote. This figure illustrates that the nodes recommended for different target nodes are different, and this is because the method ISIM considers the target node’s personalized information, such as the topology structure of the target node. Similarly, Fig. 7(b) shows the Jaccard index of different node sets recommended by different methods. This figure illustrates that the nodes selected by different algorithms are also quite different. Meanwhile, the more similar with our proposed method (i.e., ISIM), the more effective of the algorithm (combining the results in Figs. 3, 4 and 5).

6 Discussion

In this part, we mainly discuss the limitations and possible extensions of this study. For better illustration, in this paper we only deal with individual influence maximization by designing general algorithms, and our solutions could be further improved in the future. First, it is better to include different costs for the link connection (i.e., the $c(\mathbf{S})$ in Eq. (1)) instead of treating them equally. Meanwhile, more reasonable settings for parameters λ_c or threshold θ are required when we know about more prior knowledge or real propagation action logs (like using the data-driven approach for threshold θ settings [31]). What’s more, our assumption that the rest of the network stays unchanged during the link connection may be relaxed. Secondly, it is also better to study individual influence maximization and social influence modeling from the observed data rather than the simple simulation. For one thing, the information diffusion process may be affected by some other factors, e.g., information topic and homophily [32]. For another, as is only exploratory in nature, the conclusions of the simulation studies often have a great deviation to the actual propagation data. Thirdly, this study only focuses on the one target individual’s influence, and one possible extension is to add links for improving the influence spread of several individuals simultaneously, where the competitions or cooperations between each target individual may be a big

challenge. Last but not least, like the *uBound* algorithm under Linear model, we would like to find out the upper bounds under other influence models (e.g., IC, LT) and propose the corresponding scalable algorithms.

7 Conclusion

In this paper, we studied the problem of maximizing individual's own influence by recommending new links. We first formulated it as an optimization problem and designed a rational objective function. Then we proposed three algorithms to solve this intractable problem; especially the *uBound* algorithm with $O(1+\eta \cdot K)$ time complexity could handle large scale network. The experiments have shown encouraging results, and we hope this study could lead to more future work.

Acknowledgments. This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the National High Technology Research and Development Program of China (Grant No. 2014AA015203), the Natural Science Foundation of China (Grant No. 61403358) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Qi Liu acknowledges the support of the Youth Innovation Promotion Association, CAS.

References

1. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: KDD (2002)
2. Chen, W., Lakshmanan, L.V., Castillo, C.: Information and influence propagation in social networks. *Synthesis Lectures on Data Management* **5**(4), 1–177 (2013)
3. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters* **12**(3), 211–223 (2001)
4. Granovetter, M.: Threshold models of collective behavior. *American journal of sociology*, 1420–1443 (1978)
5. Aggarwal, C., Khan, A., Yan, X.: On flow authority discovery in social networks. In: SDM, pp. 522–533 (2011)
6. Xiang, B., Liu, Q., Chen, E., Xiong, H., Zheng, Y., Yang, Y.: Pagerank with priors: an influence propagation perspective. In: IJCAI (2013)
7. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: WSDM (2010)
8. Saito, K., Ohara, K., Yamagishi, Y., Kimura, M., Motoda, H.: Learning diffusion probability based on node attributes in social networks. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 153–162. Springer, Heidelberg (2011)
9. He, X., Kempe, D.: Stability of influence maximization. In: SIGKDD, pp. 1256–1265 (2014)
10. Eftekhari, M., Ganjali, Y., Koudas, N.: Information cascade at group scale. In: KDD (2013)
11. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD (2007)

12. Chaoji, V., Ranu, S., Rastogi, R., Bhatt, R.: Recommendations to boost content spread in social networks. In: WWW (2012)
13. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD (2001)
14. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD (2003)
15. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD (2010)
16. Kim, J., Kim, S.K., Yu, H.: Scalable and parallelizable processing of influence maximization for large-scale social networks? In: ICDE, pp. 266–277. IEEE (2013)
17. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: near-optimal time complexity meets practical efficiency. In: SIGMOD, pp. 75–86. ACM (2014)
18. Guo, J., Zhang, P., Zhou, C., Cao, Y., Guo, L.: Personalized influence maximization on social networks. In: CIKM (2013)
19. Facebook official blog: People You May Know. <https://www.facebook.com/notes/facebook/people-you-may-know/15610312130>
20. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In: SIGCHI (2009)
21. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: IUI (2009)
22. Tian, Y., He, Q., Zhao, Q., Liu, X., Lee, W.c.: Boosting social network connectivity with link revival. In: CIKM (2010)
23. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Which targets to contact first to maximize influence over social network. In: Greenberg, A.M., Kennedy, W.G., Bos, N.D. (eds.) SBP 2013. LNCS, vol. 7812, pp. 359–367. Springer, Heidelberg (2013)
24. Goyal, A., Lakshmanan, L.V.S.: Recmax: exploiting recommender systems for fun and profit. In: KDD (2012)
25. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. *Inf. Process. Lett.* **70**(1), 39–45 (1999)
26. Krause, A., Golovin, D.: Submodular function maximization. *Tractability: Practical Approaches to Hard Problems* **3**, 19 (2012)
27. Yang, Y., Chen, E., Liu, Q., Xiang, B., Xu, T., Shad, S.A.: On approximation of real-world influence spread. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 548–564. Springer, Heidelberg (2012)
28. Liu, Q., Xiang, B., Zhang, L., Chen, E., Tan, C., Chen, J.: Linear computation for independent social influence. In: ICDM (2013)
29. Tong, H., Faloutsos, C., Pan, J.Y.: Random walk with restart: fast solutions and applications. In: KIS (2008)
30. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web (1999)
31. Goyal, A., Bonchi, F., Lakshmanan, L.V.: A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment* **5**(1), 73–84 (2011)
32. Anagnostopoulos, A., Kumar, R., Mahdian, M.: Influence and correlation in social networks. In: KDD, pp. 7–15. ACM (2008)