Individual Influence Maximization via Link Recommendation : An Influence Overlap Perspective

Guowei Ma[*] Qi Liu[*] Enhong Chen[*] Biao Xiang[†]

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 global 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 influence 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 an objective function from the influence overlap perspective. As this optimization problem is NP-hard, we then propose greedy solutions 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, extensive 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 [33]. 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 [8].

Indeed, researchers have proposed several influence models to describe the dynamic of influence propagation process, such as Independent Cascade (IC) model [16] , Linear Threshold (LT) model [20], a stochastic information flow model [3] and the linear social influence model (Linear) [38]. Meanwhile, other researchers focus on learning the real or reasonable influence propagation probability between two individuals in the influence models [17, 35, 27]. Based on the influence propagation models and the influence propagation probabilities, influence maximization (IM) is a problem of identifying a subset with K influential nodes such that activating them leads to the maximum expected number of activated nodes of the social network. Since influence maximization is a fundamental problem in viral marketing, various aspects of it have been studied extensively in the last decade [8, 9, 5, 23, 15]. For example, Eftekhar et al. [15] studied influence propagation at group scale, where they aimed at identifying the influential groups instead of a subset of individuals.

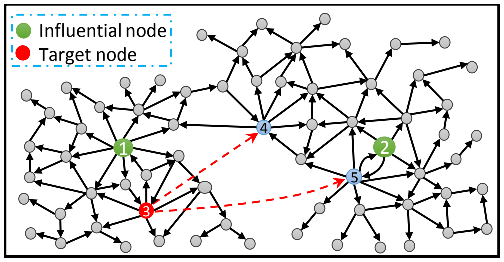


Figure 1: A toy example

However, most of existing works about influence maximization focus on identifying a subset of global influential individuals or groups with the maximum influence spread. In the real-world scenarios, an individual also cares about the influence of herself 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 marginal influence, 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 an example. If node 3 is the given target node and we want to improve its influence by recommending two new links for the node 3. Suppose that nodes 1 and 2 are the most influential node set found by influence maximization method (e.g, by CELF [28]). Actually, the total influence of the node 3 after linking with nodes 1 and 2 is less than that with nodes 4 and 5. The

[*] University of Science and Technology of China. gwma@mail.ustc.edu.cn,{qiliuql,cheneh}@ustc.edu.cn

[†] Search Technology Center Asia (STCA), Microsoft. bixian@microsoft.com

reason may be that there is much overlap of the influence spread between nodes 3 and 1. In this paper, we refer to this phenomenon (the overlap of the influence spread between nodes) as "influence overlap ".

Though similar link recommendation problems have been studied in the literature (e.g., adding new links or strengthening the weaken social links to boost the information spread across the entire network [6]), the problem of eliminating the influence overlap to maximize the target node's influence via link recommendation remains pretty much open. As a matter of fact, 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 not easy to find 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 influence of a target node/individual 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 an objective function from the influence overlap perspective. This objective function can be generally applied to different influence propagation models.
- Under our formulation, we demonstrate this optimization problem is NP-hard and 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 that our proposed objective function is rational and our algorithms are both effective and efficient.

2 Related Work

We first review the relevant work on influence modeling and influence maximization. Then we introduce the related work on the recommendation scenarios in social networks.

Influence Propagation and Maximization. Researches proposed several models for describing the influence propagation process. Independent Cascade (IC) model [16] and Linear Threshold (LT) model [20] are two widely used ones. However, both of them require Monto Carlo simulations to estimate the influence spread, which is very time-consuming, some researchers designed more efficient (or tractable) models, e.g., the stochastic information flow model [3] and the linear social influence (Linear) model [38]. Since learning the influence propagation process is beyond the scope of this paper, we use these existing influence models for illustration.

The social influence maximization problem can be traced back to Domingos and Richardson [14, 33]. Kempe et al. [25] 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 pay more attention to improving the computational efficiency of the influence maximization problem by exploiting specific aspects of the graph structure or the social influence model. Several typical algorithms include CELF [28], DegreeDiscountIC [11], P-MIA [10], LDAG [12], IRIE [24], UBLF [41] and SIMPATH [19]. Some researchers also consider other aspects of the influence maximization problem. For instance, Guo et al. [21] studied local influence maximization, aiming to find the top-K local influential nodes on the target node. Recently, Chen et al. [8] discussed some general techniques and issues about the social influence maximization problem. However, to the best of our knowledge, few attention has been paid to the problem of maximizing the target node's influence via link recommendation.

Recommendations in social networks. According to the recommended objects, most of the recommendation problems in social networks can be categorized into two types: The item recommendation [2], which recommends interesting items, such as songs, books and other products, to users. For example, Liu et al. studied the problem of selecting a set of influential seed items for user interests guiding [30]; The user recommendation, which recommends users in the same network to a given user, in order to help her discover potential friends or improve the connectivity of the networks [6, 36]. Since our problem of individual influence maximization is also a user recommendation problem, the following discussions mainly focus on this category.

Actually, recommending users in social networks is an important task for many social network sites like Twitter, Google+ and Facebook, e.g., the "People You May Know" feature on Facebook. Thus, many 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 [1] and other interest-based or profile-based algorithms [7, 22]. Some of these works also consider the influence propagation effect, such as selecting a set of "influential" users for a new user [34] or a new product [18], 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 [36, 13]. For instance, Tian et al. [36] suggested users to re-connect their old friends and strengthen the existing weak social ties with the objective of improving the social network connectivity; Chaoji et al. [6] recommended an edge set in order to increase the connectivity among users and boost the content spread in the entire social network. However, we can see that this kind of related work pays more attention to the entire social network rather than the target individual's influence spread.

3 Individual Influence Maximization

In this section, we first show that this individual influence maximization problem could be formulated as an optimization problem and then define a rational objective function from the influence overlap perspective. Second, we exploit the properties of this objective function and propose greedy solutions with a performance guarantee.

Preliminaries. Let the directed graph $G = (V, E, T)$ represents an influence network, where $V = \{1, 2, ..., n\}$ are n nodes in graph and E stores all the influence links(edges) between nodes. $T = [t_{ij}]_{n*n}$ is a given propagation probability matrix. For each edge $(i, j) \in E$, $t_{ij} \in (0, 1)$ denotes the influence propagation probability from node i to node j. For any edge $(i, j) \notin E$, $t_{ij} = 0$. 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 [16], LT [20] and Linear [38]). Specifically, $\mathbf{f}_i = [f_{i\to1}, f_{i\to2}, ..., f_{i\to n}]'$, 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 $f_{i\to V} = \sum_{j\in V} f_{i\to j}$. Indeed, $f_{i\to V}$ 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 influence, she should make new influence links [1] 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 of nodes $\mathbf{S}$ 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 $t \in V$ (the node whose influence needs to be maximized) makes new links with nodes in $\mathbf{S}$, the influence gain of node t is maximum, and the cost $c(\mathbf{S})$ does not exceed a specific budget B. Now this problem could be formulated as an optimization problem as follows:

$$(3.1) \qquad \arg\max_{\mathbf{S}} \; \{f^{\mathbf{S}}_{t\to V} - f_{t\to V}\}\text{ , subject to } c(\mathbf{S}) \le B,$$

where $f^{\mathbf{S}}_{t\to V} - f_{t\to V}$ is the influence gain of the target node t after linking with nodes in set $\mathbf{S}$. Our goal is to solve the optimization problem above. 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 $\mathcal{F}(\mathbf{S}) = \{f^{\mathbf{S}}_{t\to V} - f_{t\to V}\}$ and $B = K$, we can rewrite Eq. (3.1) as below.

$$(3.2) \qquad \arg\max_{\mathbf{S}} \mathcal{F}(\mathbf{S}) = \{f^{\mathbf{S}}_{t\to V} - f_{t\to V}\}\text{ , s.t. } |\mathbf{S}| \le K.$$

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

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 below:*

$$\begin{aligned} \mathcal{F}(\{c\}) &= f^{\{c\}}_{t\to V} - f_{t\to V} \\ &= \lambda_c * (1 - f_{t\to c}) * \sum_{i\in V} (f_{c\to i} * [1 - f_{t\to i}]), \end{aligned}$$

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

Definition Explanation. Let us take a simple example.

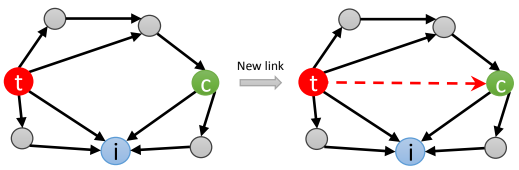


Figure 2: A Simple Example

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. Firstly, we show how to estimate

[1] In this paper, we use the expressions of "influence link "and "link" without distinction.

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 ($f_{t\to i}$) and node c ($f_{t\to c}$) respectively, and meanwhile, the node c also has an influence on node i ($f_{c\to i}$). 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 $\lambda_c(1 - f_{t\to c})$. Now, let's explain this definition: Suppose t always influences (or actives) c successfully ($f^{\{c\}}_{t\to c} = 1$), then the influence of t on node c will be increased by $(f^{\{c\}}_{t\to c} - f_{t\to c}) = (1 - f_{t\to c})$. 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 $\lambda_c(1 - f_{t\to c})$.[2] One step further, we can represent that the influence of t on i through c has increased by $\lambda_c(1 - f_{t\to c})f_{c\to i}$. Hence, the new influence of t on i can be calculated as $f^{\{c\}}_{t\to i} = f_{t\to i} + (1 - f_{t\to i})\lambda_c(1 - f_{t\to c}) * f_{c\to i}$, and the influence gain of t on i after making a link with c is: $f^{\{c\}}_{t\to i} - f_{t\to i} = (1 - f_{t\to i})\lambda_c(1 - f_{t\to c}) * f_{c\to i}$. Then, we can get the total influence gain of node t on the entire network as below:

$$\begin{aligned} f^{\{c\}}_{t\to V} - f_{t\to V} &= \sum_{i\in V}(f^{\{c\}}_{t\to i} - f_{t\to i}) \\ &= \sum_{i\in V}(1 - f_{t\to i}) * \lambda_c(1 - f_{t\to c})f_{c\to i} \\ &= \lambda_c(1 - f_{t\to c})\sum_{i\in V}(1 - f_{t\to i})f_{c\to i} \end{aligned}$$

According to 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 subset* $\mathbf{S}$, *we define* $\mathcal{F}(\mathbf{S})$ *as below:*

$$\begin{aligned} \mathcal{F}(\mathbf{S}) &= f^{\mathbf{S}}_{t\to V} - f_{t\to V} = \sum_{c\in\mathbf{S}}\mathcal{F}(\{c\}) \\ &= \sum_{c\in\mathbf{S}}\lambda_c(1 - f_{t\to c}) * \sum_{i=1}^{n}(f_{c\to i} * [1 - f_{t\to i}]). \end{aligned}$$

3.3 Properties of Function $\mathcal{F}(\mathbf{S})$ In this part, we demonstrate that the function $\mathcal{F}(\mathbf{S})$ satisfies the properties below.[3]

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 non-negative 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.

3.4 Greedy Strategy Indeed, maximizing submodular function in general is NP-hard [26]. Thus the optimization problem we formulated in this paper is NP-hard, for the objective function is submodular [25, 26]. However, for a non-negative monotone submodular function, such as $\mathcal{F}(\mathbf{S})$, the greedy strategy approximates the optimum within a factor of $(1 - 1/e)$ [25].

The simple greedy strategy starts with the empty set $\mathbf{S} = \emptyset$, and requires about n times influence spread estimation in each iteration to select one node to join $\mathbf{S}$. Thus, the *greedy* algorithm requires about $(n * K)$ times influence spread estimations, where $K = |S|$. Due to the fact that each influence spread estimation calculated by influence models (e.g., IC, LT) is very time-consuming, the *greedy* algorithm is quite slow. Algorithm 1 shows the details about the simple *greedy* algorithm.

Algorithm 1: The *greedy* algorithm

```
Input: $\mathbf{G} = (V, E, T)$, a given target node $t$, top $K$
Output: $\mathbf{S}$ with $K$ nodes
initialize $\mathbf{S} = \emptyset$;
while $|\mathbf{S}| < K$ do
    select $s = \arg\max_{u\in\mathcal{V}\setminus\mathbf{S}}\ (f^{\mathbf{S}\cup u}_{t\to V} - f^{\mathbf{S}}_{t\to V})$ ;
    $\mathbf{S} = \mathbf{S} \cup s$ ;
Return $\mathbf{S}$;
```

Scaling Up. Here, we exploit the submodular property of $\mathcal{F}(\mathbf{S})$ and adopt the lazy-forward strategy [28] for scaling up the simple greedy algorithm. 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 gain in the previous iteration, we propose the algorithm $lazy$ without sacrificing any accuracy. Algorithm 2 shows the details about the $lazy$ algorithm. Because $lazy$ just requires n times influence spead estimations in the first iteration for calculating the upper bound of influence gain of each candidate node i, it requires totally $(n + \theta * K)$ times influence spread estimations, where $\theta \ll n$ is the expected number of influence spread estimations in each iteration.

4 Optimization under the Linear Model

In the previous section, we design a rational objection function to address the challenge of influence overlapping and propose two greedy algorithms (*greedy* and *lazy*) to solve this link recommendation problem. However, these algorithms

[2] Notice that, in the experiments of this paper, we choose the same λ_c for all the nodes for simplicity. In real-world applications, its value can be also determined by the local structure and properties of nodes c and t.

[3] The detailed proof appears in the Appendix

Algorithm 2: The $lazy$ Algorithm

```
Input: $\mathbf{G} = (V, E, T)$, a given target node $t$, top $K$
Output: $\mathbf{S}$ with $K$ nodes
for each node $i \in V$ do
    calculate $\mathcal{F}(\{i\}) = f^{\{i\}}_{t \to V} - f_{t \to V}$;
    $flag_i$ = 0;
initialize $\mathbf{S}$ = $\emptyset$;
while $|\mathbf{S}| < K$ do
    s = Find the biggest $\mathcal{F}(\{s\})$ in $\mathcal{F}$;
    if $flag_s$ == $|\mathbf{S}|$ then
        $\mathbf{S} = \mathbf{S} \cup s$ ;
        $\mathcal{F}(s) = 0$ ;
    else
        recalculate $\mathcal{F}(\{s\}) = f^{\mathbf{S} \cup s}_{t \to V} - f^{\mathbf{S}}_{t \to V}$ ;
        $flag_s = |\mathbf{S}|$ ;
Return $\mathbf{S}$;
```

require too many times of influence spread estimations. To address the challenge of inefficiency, we further explore this problem on a specific influence model, the linear social influence (Linear) model [38]. 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 [3], and the linear approximation method for the IC model [39] is a special case of Linear. In the following, we first review the Linear model. Then, we propose an algorithm *uBound* to further scale up the *lazy* algorithm by exploiting the property of influence computation in Linear.

Review. Given a directed graph $G = (V, E, T)$, Xiang et al. [38] defined the linear social influence prorogation (Linear) model as below:

DEFINITION 3. *Define the influence of node i on j as*

$$f_{i \to i} = \alpha_i, \quad \alpha_i \in (0, 1] \tag{4.3}$$

$$f_{i \to j} = d_j \sum_{k \in N_j} t_{kj} f_{i \to k}, \ \ for \ \ j \neq i \tag{4.4}$$

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 and the smaller d_j is, the more influence will be blocked by node j.

Under the linear social influence (Linear) model, there is an upper bound to measure a node's influence [38]:

$$f_{i \to V} = \sum_{j=1}^{n} f_{i \to j} \tag{4.5}$$

$$\leq \alpha_i * \mathbf{e} * (I - DT')^{-1}_{\cdot i} \tag{4.6}$$

$$= \alpha_i * (I - DT)^{-1}_{i \cdot} * \mathbf{e}' \tag{4.7}$$

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

4.1 Optimization with Upper Bounds In this part, we exploit the properties of the influence computation in Linear model and find an upper bound for each node. It's worth noting that the upper bounds of all nodes can be finished in $O(|E|)$ time. Then, we use these upper bounds to replace the first iteration of the *lazy* algorithm for guiding the computational order of the candidate nodes and propose the *uBound* algorithm without sacrificing any accuracy.

THEOREM 4.1. *(**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 :*

$$\mathcal{F}(\{i\}) = f^{\{i\}}_{t \to V} - f_{t \to V} \leq \lambda_i * \alpha_i * (I - DT)^{-1}_{i \cdot} * \mathbf{e}'$$

Proof. We prove first that $\mathcal{F}(\{i\}) = f^{\{i\}}_{t \to V} - f_{t \to V} \leq \lambda_i * f_{i \to V}$. According to the marginal influence definition,

$$\mathcal{F}(\{i\}) = f^{\{i\}}_{t \to V} - f_{t \to V} \tag{4.8}$$

$$= \lambda_i (1 - f_{t \to i}) \sum_{k=1}^{n} (f_{i \to k} * [1 - f_{t \to k}]) \tag{4.9}$$

$$\leq \lambda_i (1 - f_{t \to i}) * \sum_{k=1}^{n} f_{i \to k} \tag{4.10}$$

$$\leq \lambda_i \sum_{k=1}^{n} f_{i \to k} = \lambda_i * f_{i \to V} \tag{4.11}$$

Both Eqs. (4.10) and (4.11) hold because $f_{i \to j} \in [0, 1]$.

Combining Eqs. (4.7) with (4.11) , we have proved that

$$\mathcal{F}(\{i\}) = f^{\{i\}}_{t \to V} - f_{t \to V} \leq \lambda_i \alpha_i (I - DT)^{-1}_{i \cdot} * \mathbf{e}'.$$

Based on Theorem 4.1, we 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, $\lambda \alpha_i * (I - DT)^{-1}_{i \cdot} * \mathbf{e}'$. We can rewrite Theorem 4.1 into vector: $[\mathcal{F}(\{1\}), \mathcal{F}(\{2\}), ..., \mathcal{F}(\{n\})]' \leq diag(\lambda_1, \lambda_2, ..., \lambda_n) * diag(\alpha_1, \alpha_2, ..., \alpha_n) * (I - DT)^{-1} * \mathbf{e}'$. As $(I - DT)$ is a strictly diagonally dominant matrix, and the value of $(I - DT)^{-1} * \mathbf{e}'$ can be quickly calculated through Gauss-Seidel method in $O(|E|)$ time. Thus, we can replace influence gain estimations in the first iteration of *lazy* algorithm for each candidate node $i \in V$ by its upper bound, and then propose the $uBound$ algorithm. Algorithm 3 shows the details about the *uBound* algorithm. From the analysis above, we know that *uBound* requires only $(1 + \eta * K)$ 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 2) requires, in total, $(n + \theta * K)$ times.

Algorithm 3: The $uBound$ Algorithm

```
Input: $\mathbf{G} = (V, E, T)$, a given target node $t$, top $K$
Output: $\mathbf{S}$ with $K$ nodes
Compute the upper bound vector
  $\mathbb{U} = diag(\lambda_1, \lambda_2, ..., \lambda_n) * diag(\alpha_1, \alpha_2, ..., \alpha_n) *$
  $(I - DT)^{-1} * \mathbf{e}'$ in $O(|E|)$ time;
for each node $i \in V$ do
    $\mathcal{F}(i) = \mathbb{U}_i$;
    $flag_i$ = 0;
initialize $\mathbf{S}$ = $\emptyset$;
while $|\mathbf{S}| < K$ do
    s = Find the biggest $\mathcal{F}(s)$ in $\mathcal{F}$;
    if $flag_s$ == $|\mathbf{S}|$ then
        $\mathbf{S} = \mathbf{S} \cup s$ ;
        $\mathcal{F}(s) = 0$ ;
    else
        recalculate $\mathcal{F}(s) = f^{\mathbf{S} \cup s}_{t \to V} - f^{\mathbf{S}}_{t \to V}$ ;
        $flag_s = |\mathbf{S}|$ ;
Return $\mathbf{S}$;
```

5 Experiments

In this section, we first demonstrate that our proposed objective function is rational and effective under existing influence models. Secondly, we compare the performance (e.g., efficiency) of the algorithms ($greedy$, $lazy$ and $uBound$). Finally, we use a case study to illustrate the necessity of designing individualized link recommendation algorithms.

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

5.1 Experimental Setup. The experiments are conducted on four real-world datasets with different sizes, which include: (a)Wiki-Vote, a who-votes-on-whom network at Wikipedia where nodes are users and an edge(i,j) represents that user i voted on user j; (b)Weibo, a social media network in China, where nodes are the users and edges are their followships. We crawled this data from weibo.com[4] at March 2013 and then sampled a small network which only contains the verified users for filtering the zoombie accounts; (c)Cit-HepPh, an Arxiv High Energy Physics paper citation network where nodes represent papers and an edge (i,j) represents that paper i cites paper j. Both Cit-HepPh and Wiki-Vote are downloaded from SNAP[5] [29]; (d)Twitter, another social media network. We downloaded this data from Social Computing Data Repository at ASU[6] [40]. Detailed information of these datasets can be found in Table 1.

Influence Model and Propagation Probability. We validate our discoveries under the Independent Cascade(IC) model, Linear Threshold(LT) model and Linear social influence (Linear) model, as widely used in the literature [25, 11, 19, 28, 38, 31]. 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 $(i, j) \in E$ [7] in G and then assign the corresponding influence propagation probability $t_{ij} = 1/indegree(j)$. For LT [25], 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. [38, 31] (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. In our proposed objective function, the λ_is are set to be 0.85 for simplicity.

All the experiments were conducted on Windows 64-bit operating system with 2.20GHz 8-Core Intel(R) Core(TM) i3-2330M CPU and 16GB of main memory.

5.2 Real Influence Gain Comparison We 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. For a given target node t, we first calculate its original influence. Secondly, we let node t make new links with the ones in $\mathbf{S}$ recommended by different methods, and recalculate the new influence of node t [8]. Finally, we get the real influence gain by subtracting the original influence from the new influence of t. Thus, the performance of each method is evaluated by the influence gain it could provides to the target node, i.e., the larger influence gain, the better the method is. In the following, we call our method as ISIM (**I**ndividual **S**ocial **I**nfluence **M**aximization) 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 [37].
- **PageRank**. Let the target node link to the top K ranked nodes, i.e., with the largest PageRank value[32].
- **HighestInf (Highest Influence)**. Based on the influence model (IC, LT or Linear), we compute each n-

[4] http://www.weibo.com/

[5] http://snap.stanford.edu/data/

[6] http://socialcomputing.asu.edu/datasets/Twitter

[7] For example, if user j follows user i in Twitter, then i influences j.

[8] 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 $1/new\_indegree(c)$.

ode's influence and let the target node make links with the 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.

- **IMSeeds**. It selects and recommends the most influential node set **S** by using the CELF algorithm [28] for traditional social influence maximization problem. This method could alleviate the influence overlap between the nodes in **S**. However, it does not consider the influence overlap between the target node and those in **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 $|\mathbf{S}| = 5, 10, ..., 50$) 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 sets **S** selected by our ISIM could help the target node to get more real influence gain than the benchmarks. The results also demonstrate that the global influential nodes cannot guarantee the best performance for this individual social influence maximization problem. By the way, 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.

5.3 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
wiki-vote	*greedy*	35208	70390	105547	140679	175786	210868	245925
	lazy	7124	7139	7171	7195	7220	7250	7306
	uBound	**19**	**53**	**84**	**119**	**157**	**204**	**265**
Weibo	*greedy*	36836	73646	110431	147191	183926	220636	257321
	lazy	7414	7446	7505	7572	7638	7738	7838
	uBound	**50**	**98**	**180**	**278**	**356**	**494**	**622**
Cit-HepPh	*lazy*	34554	34566	34586	34608	34642	34660	34691
	uBound	**17**	**37**	**64**	**89**	**137**	**161**	**194**
Twitter	*uBound*	**16**	**43**	**71**	**99**	**136**	**166**	**212**

Table 2 shows the expected numbers of influence spread estimations when selecting different top K seeds using different algorithms[9]. From the results, we know 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=35 is reduced at a rate of 96.2%, 92.1%, 99.4% on the three datasets(ie, Wiki-vote, Weibo, Cit-HepPh), respectively. The reason is that $lazy$ requires n times influence spread estimations in the first iteration to establish the initial upper bounds of the marginal influence, while $uBound$ requires only one influence spread estimation to establish the upper bounds for all the nodes.

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 $(n * K)$ 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 selected node set **S** increases

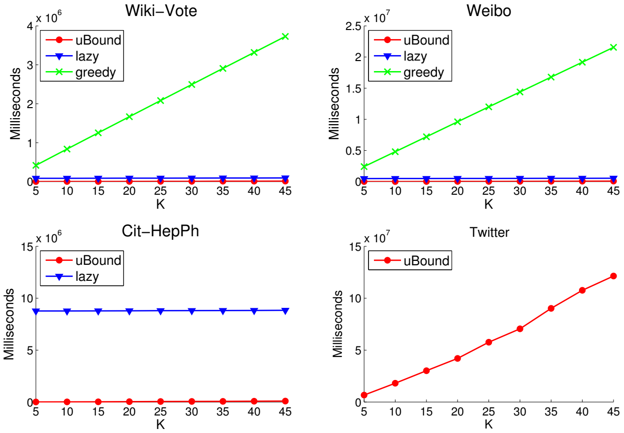
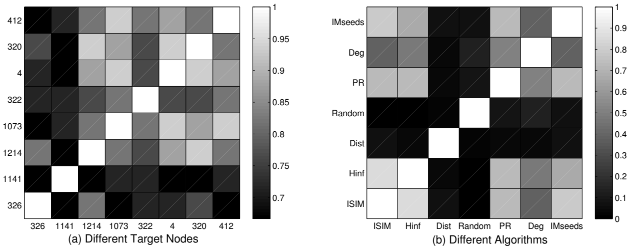


Figure 6: Real Runtime Comparisons

5.4 Case Study For the purpose of showing that it is necessary to recommend different node sets for different target nodes, we do the following case study. Fig. 7(a) shows the Jaccard similarity coefficient of the 25 nodes recommended by ISIM for 8 target nodes which are randomly selected (distinguished by node ID) from wiki-vote. From this figure we can see that the nodes recommended for different target nodes are different and this is because our proposed algorithm(i.e., ISIM) considers the target node's personalized information, such as the topology structure of the target node. Similarly, Fig. 7(b) shows the Jaccard similarity coefficient 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).



(a) Different Target Nodes

(b) Different Algorithms

Figure 7: Jaccard Similarity Comparison

[9] Similar experimental results at top K=40 and 45 are omitted.

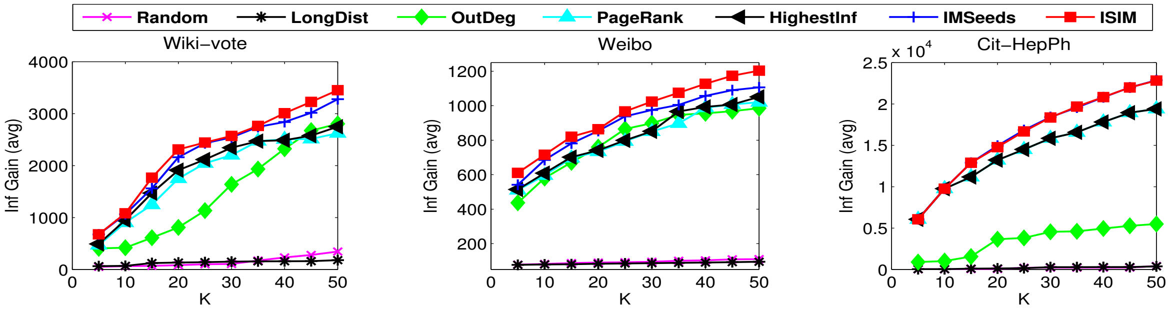


Figure 3: Comparisons under IC Model

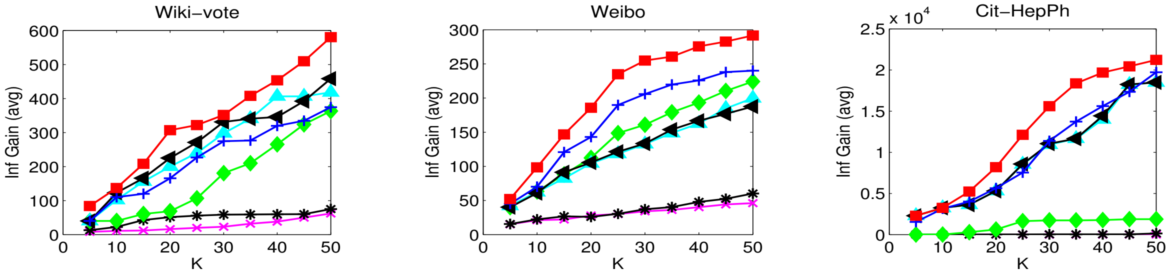


Figure 4: Comparisons under LT Model

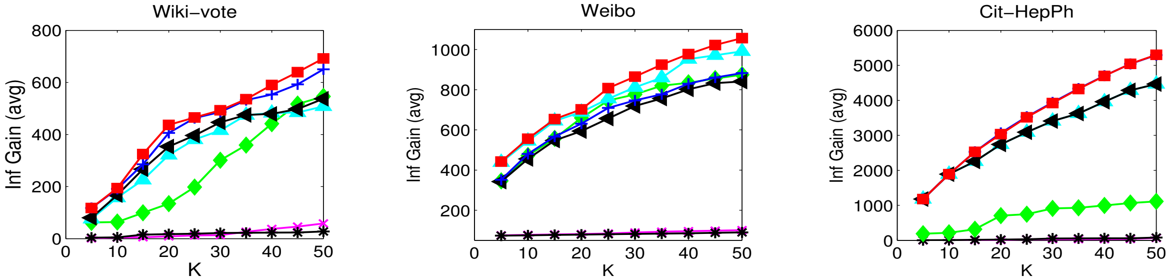


Figure 5: Comparisons under Linear Model

6 Discussions

We discuss the advantages and limitations of this study. From the experimental results, we can see that the nodes selected by our defined objective function $\mathcal{F}(\mathbf{S})$ could achieve more influence gain for the target node than other benchmarks, which illustrates that the definition of $\mathcal{F}(\mathbf{S})$ can effectively eliminate the impact of influence overlap between nodes. What's more, we scaled up the greedy solution and proposed the $uBound$ algorithm which could handle the large scale networks.

For better illustration, in this paper we deal with individual influence maximization by designing general algorithms, and our solutions could be further improved in the future. First, some constraints existing in the real world should be taken into consideration. For instance, it is better to include different costs for the link connection (i.e., the $c(\mathbf{S})$ in Eq. (3.1)) instead of treating them equally. Meanwhile, our assumption that the rest of the network stays unchanged during the link connection may be relaxed. Second, 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, besides social influence, the information diffusion process may be affected by some other factors, e.g., information topic and homophily (i.e., the correlations of user characters) [4]. For another, as is only exploratory in nature, the conclusions of the simulation studies often have a great deviation to the actual propagation data. Third, 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, similar to the proposed $uBound$ 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 recommendation algorithms.

7 Conclusions

In this paper, we studied the problem of individual influence maximization by recommending new links. Firstly we showed that this problem could be formulated as an optimization problem and defined a rational objective function from the influence overlap perspective. Since solving this function is NP-hard, we then exploited the properties of the objective function and proposed a greedy algorithm ($greedy$)

to solve it with a performance guarantee. Furthermore, we proposed two scalable algorithms $lazy$ and $uBound$ with $O(n + \theta * K)$ and $O(1 + \eta * K)$ time complexity respectively. Specifically, $uBound$ is based on the Linear model and could be applied to large scale networks. Finally, extensive experimental results validated the performance of our proposed algorithms. We hope this study could lead to more future work.

References

[1] *Facebook official blog: People You May Know*. https://www.facebook.com/notes/facebook/people-you-may-know/15610312130.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *TKDE*, 17(6):734–749, 2005.

[3] C. Aggarwal, A. Khan, and X. Yan. On flow authority discovery in social networks. In *SDM*, 2011.

[4] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*, 2008.

[5] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *Internet and Network Economics*, pages 306–311. Springer, 2007.

[6] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt. Recommendations to boost content spread in social networks. In *WWW*, 2012.

[7] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *SIGCHI*, 2009.

[8] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[9] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI*, 2012.

[10] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 2010.

[11] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, 2009.

[12] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, 2010.

[13] E. D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *Algorithm Theory-SWAT 2010*, pages 420–431. Springer, 2010.

[14] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, 2001.

[15] M. Eftekhar, Y. Ganjali, and N. Koudas. Information cascade at group scale. In *KDD*, 2013.

[16] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[17] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.

[18] A. Goyal and L. V. S. Lakshmanan. Recmax: exploiting recommender systems for fun and profit. In *KDD*, 2012.

[19] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, 2011.

[20] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[21] J. Guo, P. Zhang, C. Zhou, Y. Cao, and L. Guo. Personalized influence maximization on social networks. In *CIKM*, 2013.

[22] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *IUI*, 2009.

[23] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, 2012.

[24] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In *ICDM*, 2012.

[25] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[26] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

[27] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis. Strip: stream learning of influence probabilities. In *KDD*, 2013.

[28] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[29] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[30] Q. Liu, B. Xiang, E. Chen, Y. Ge, H. Xiong, T. Bao, and Y. Zheng. Influential seed items recommendation. In *RecSys*, 2012.

[31] Q. Liu, B. Xiang, L. Zhang, E. Chen, C. Tan, and J. Chen. Linear computation for independent social influence. In *ICDM*, 2013.

[32] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[33] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

[34] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Which targets to contact first to maximize influence over social network. In *Springer Berlin Heidelberg*, 2013.

[35] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, and H. Motoda. Learning diffusion probability based on node attributes in social networks. In *ISMIS*, pages 153–162, 2011.

[36] Y. Tian, Q. He, Q. Zhao, X. Liu, and W.-c. Lee. Boosting social network connectivity with link revival. In *CIKM*, 2010.

[37] H. Tong, C. Faloutsos, and J.-Y. Pan. Random walk with restart: fast solutions and applications. *KIS*, 2008.

[38] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. Pagerank with priors: An influence propagation perspective. In *IJCAI*, 2013.

[39] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad. On approximation of real-world influence spread. In *ECML-PKDD*. 2012.

[40] R. Zafarani and H. Liu. Social computing data repository at ASU. `http://socialcomputing.asu.edu`, 2009.

[41] C. Zhou, P. Zhang, J. Guo, X. Zhu, and L. Guo. Ublf: An upper bound based approach to discover influential nodes in social networks. In *ICDM*, 2013.

Appendix

3.3 Properties of Function $\mathcal{F}(\mathbf{S})$ In this part, we 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 non-negative 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.

Properties 1 and 2 are straightforward, we only give the proof for the last property.

THEOREM 3.1. *For an influence propagation model, the objective function $\mathcal{F}(\mathbf{S})$ is submodular.*

Proof. Let $\mathbf{X} \subseteq \mathbf{Y} \subseteq \mathbf{V}$, and node $u \in \mathbf{V} \backslash \mathbf{Y}$, we demonstrate that $\mathcal{F}(\mathbf{S})$ satisfies the defining inequality for submodularity, $\mathcal{F}(\mathbf{X} \cup u) - \mathcal{F}(\mathbf{X}) \geq \mathcal{F}(\mathbf{Y} \cup u) - \mathcal{F}(\mathbf{Y})$.

We can expand the above inequality as follows:

$$\begin{aligned}\mathcal{F}(\mathbf{X} \cup u) - \mathcal{F}(\mathbf{X}) &= f_{t\to V}^{X\cup u} - f_{t\to V}^{X} \\ &= \lambda_u(1 - f_{t\to c}^{\mathbf{X}\cup u}) \sum_{i\in V}(f_{u\to i}[1 - f_{t\to i}^{\mathbf{X}}]);\end{aligned}$$

$$\begin{aligned}\mathcal{F}(\mathbf{Y} \cup u) - \mathcal{F}(\mathbf{Y}) &= f_{t\to V}^{Y\cup u} - f_{t\to V}^{Y} \\ &= \lambda_u(1 - f_{t\to c}^{\mathbf{Y}\cup u}) \sum_{i\in V}(f_{u\to i}[1 - f_{t\to i}^{\mathbf{Y}}]).\end{aligned}$$

For any node $i \in V$, $f_{t\to i}^{X} \leq f_{t\to i}^{Y}$ due to the nondecreasing property of $\mathcal{F}(\mathbf{S})$, we can see that

$$\lambda_u(1-f_{t\to u}^{X}) f_{u\to i}(1-f_{t\to i}^{X}) \geq \lambda_u(1-f_{t\to u}^{Y}) f_{u\to i}(1-f_{t\to i}^{Y}).$$

Thus $\mathcal{F}(\mathbf{X} \cup u) - \mathcal{F}(\mathbf{X}) \geq \mathcal{F}(\mathbf{Y} \cup u) - \mathcal{F}(\mathbf{Y})$. Hence, we proved that $\mathcal{F}(\mathbf{S})$ is a submodular function.