Multiple Influence Maximization in Social Networks*

Honglin Sun
Shanghai Key Lab of Scalable Computing and Systems
Dept. of Computer Science and Engineering Shanghai Jiao Tong University
Shanghai, China
sunhonglin@sjtu.edu.cn

Xiaofeng Gao
Shanghai Key Lab of Scalable Computing and Systems
Dept. of Computer Science and Engineering Shanghai Jiao Tong University
Shanghai, China
gao-xf@cs.sjtu.edu.cn

Guihai Chen
Shanghai Key Lab of Scalable Computing and Systems
Dept. of Computer Science and Engineering Shanghai Jiao Tong University
Shanghai, China
gchen@cs.sjtu.edu.cn

Jian Gu
Key Laboratory of Information Network Security of Ministry of Public Security
The Third Research Institute of Ministry of Public Security
Shanghai, China
gujian@mctc.gov.cn

Yongjian Wang
Key Laboratory of Information Network Security of Ministry of Public Security
The Third Research Institute of Ministry of Public Security
Shanghai, China
wangyongjian@stars.org.cn

ABSTRACT

Influence Maximization, a technique of social analysis to help marketers select a small group of users to promote their products with the goal of maximizing their influence spread, has been studied for a long time. However, previous researches did not consider the scenario that a company may intend to promote serval kinds of products in the same social network, where each user has different preferences for different categories. Considering this scenario in this paper, we propose the *Multiple Influence Maximization (MIM)* problem based on the assumption that one seed user can accept serval kinds of goods for free, and at the same time non-seed users have enough purchasing power to accept different promotions from their social friends. To address this issue, we design the corresponding greedy algorithm framework with an approximate radio of $1 - 1/e$ if the influence function is submodular. Besides, to simplify the computation of influence spread and considering the properties of real social influence propagation, we propose a novel influence quantitative computation model in which influence function fortunately is submodular. Applying this model into our MIM problem, the experiment results show that the greedy algorithm out-performs serval traditional heuristics in influence spread without loss of performance.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

Keywords

Influence Maximization/Propagation, Social Analysis

*X. Gao and Y. Wang are the corresponding authors.

IMCOM '16, January 04-06, 2016, Danang, Viet Nam

DOI: http://dx.doi.org/10.1145/2857546.2857591

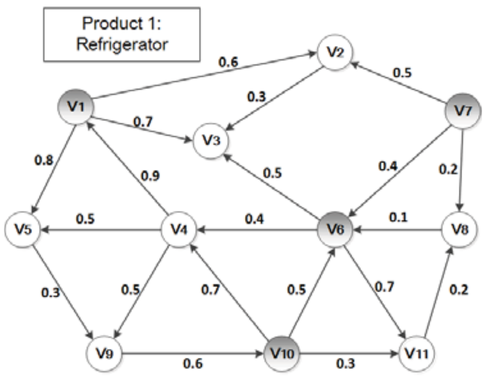
1. INTRODUCTION

Online social networks, *e.g.*,Twitter, Facebook and Flixster can provide various valuable information for marketers, and a company can make the best use of these information to promote its products to a large potential customer base. The *Influence Maximization (IM)* problem has thereby been proposed, whose goal is to select top k users in social network to maximize the influence spread. These selected users are usually called "seeds", and the limited k number reflects the additional budget constraint that the company should pay. IM problem has significance for both companies and individuals to promote their products, services, and innovative ideas through social networks, and many literatures have discussed this problem from different aspects [1–11].

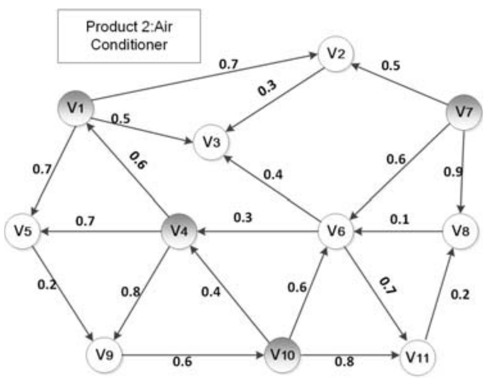
However, previous researches did not consider the scenario that a company may intend to promote several kinds of products in the same social network simultaneously. In reality, in majority of social networks, different users have different interests for different products and thus have different acceptance probabilities of promotions from their social friends. Hence, the company may adopt a promotion strategy that each seed user can freely recommend serval different kinds of items together and each non-seed user can consider accepting different categories' promotions at the same time.

Correspondingly, in this paper we consider a scenario as follows: without loss of generality, assume a company has m kinds of products to be promoted and wants to assign k items among them to a small group users (seed users) who have the highest influence in a social network, and the goal is to maximize the overall influence spread. Compared with traditional IM problem, the key issue here is how to decide the number of each product among the k items and identify the most k influential individuals to form the seed users.

To deal with this challenge, we propose the *Multiple Influence Maximization (MIM)* problem, for which we aim to maximize the overall influence spread among all selected items comprehensively. More specifically, for each kind of product we construct a corresponding social graph and all graphs share the common vertices, but edges in different graphs have different weights reflecting the influence probabilities. Each product's influence propagates independently in the social network. Our goal is to select the most k influential nodes from these weighted graphs to maximize the overall influence spread. The final seed set consists of seed sets for each social graph and each seed set is responsible for promoting one single product category. Here, a seed node in one graph may also be a seed node in another graph (seed sets can overlap with each other), which means that a seed can simultaneously recommend different products to its outer neighbors. The only constraint here is that the sum of cardinality of all seed sets cannot extend k (we count the duplications). Now we give an example to better illustrate the



(a) Graph for Refrigerator



(b) Graph for Air Conditioner

Figure 1: Example Social Graphs for Two Products

MIM problem. For example, assume a company wants to promote 2 products (e.g., Refrigerator and Air Conditioner) and it prepares to promote 8 items in social network. Figure 1 shows two social graphs according to the two products, both of which have the same vertex set. Figure 1(a) is the social graph for Refrigerator and Figure 1(b) is the one for Air Conditioner. As described above, we need to select the 8 most influential nodes from these two graphs. Let p_1 represents the Refrigerator and p_2 represents Air Conditioner, respectively. We use G_1 to present the Figure 1(a) and G_2 to Figure 1(b), S_1 and S_2 to present the selected seed node sets for p_1 and p_2.

An example computation is shown as the grey nodes in Figure 1. We can see that seed set S_1 consists of nodes $\{v_1, v_6, v_7, v_{10}\}$, and S_2 consists of nodes $\{v_1, v_4, v_7, v_{10}\}$, which implies that $|S_1| + |S_2| = 8$. Note that v_1, v_7, v_{10} are duplicately selected for two products. It means that these seeds will promote two products simultaneously to achieve an overall influence maximization. Finally, the selected seed set should be $\{v_1, v_4, v_6, v_7, v_{10}\}$, which means that $|S_1 \cup S_2| \leq 8$. From this example, we can also find that the difficulty to solve the MIM problem not only comes from selecting k seeds from a large and complex social networks, but also comes from how to divide k efficiently to m products so that the overall influence spread is maximized.

The MIM problem is an extension of the traditional IM problem, which can be regarded as the integration of serval independent IM problems, if we divide k appropriately. Also, if $m = 1$, the MIM problem degenerates to traditional IM problem. Since IM problem is proved to be NP-hard [3] we can assert that the *Multiple Influence Maximization* problem is NP-hard.

Since dividing k is an important part to solve the MIM problem, the traditional greedy framework for IM is not suitable for MIM and we propose a new greedy framework, which selects seed node with highest marginal gain based on seed set of the graph that it belongs to. Since the influence of each kind spreads independently in the same social network and according to [12], this framework has an approximate radio of $1 - 1/e$ if the influence function is submodular. Actually, in a real social network, influence can propagate along a social path and each individual has multiple influential social source as well as multiple social paths to influence another. Thus in this paper, we reasonably assume that each individual influence another one independently along these social paths and propose a novel influence computation model. Fortunately, this influence function is submodular and the calculation of marginal gain based on this model can be completed in linear time. Therefore, it can be perfectly applied into our greedy framework and can indeed accelerate seed selection. We finally propose various numerical experiments to validate the efficiency of our designs.

In summary, we make the following contributions:

- We are the first to consider the *Multiple Influence Maximization* problem (MIM), where multiple kinds of products can promoted in the same social network simultaneously.
- As the MIM problem is NP hard, we propose a novel greedy framework to solve the MIM problem efficiently.
- Considering the properties of real social network, we propose a novel influence computation model where the influence calculation function is submodular and thus it guarantees the approximate radio of $1 - 1/e$ of MIM greedy framework without loss of performance.
- We propose many numerical experiments with trace data to validate the efficiency of our designs.

The rest of the paper is organized as follows. Section 2 introduces the related work and Section 3 formally define the Multiple Influence Maximization problem and presents the greedy framework. In Section 4 we propose the influence spread computation model and iterative marginal gain updating algorithm. The experiment results are shown in Section 5 and Section 6 concludes this paper.

2. RELATED WORK

Domingos et al. [1, 2] first proposed *Influence Maximization* problem in social network and Kempe et al. [3] proposed two widely known influence propagation model, *Independent Cascade (IC)* model and *Linear Threshold (LT)* model, where information disseminates in discrete steps, further treated it as discrete optimization problem and proved it is NP-hard for both two propagation model. Simultaneously, they worked out the greedy algorithm framework with an approximate radio of $1-1/e$. As it has low efficiency and can hardly adapt to large scale online social network, there are so much work to accelerate influence computation and seed selection, [4–10]. [4, 10] proposed the greedy algorithm with an approximate radio of $1 - 1/e - \varepsilon$ and near-optimal time complexity based on stochastic analysis technique. In [5, 6], Chen et al. proposed the PMIA algorithm for IC model and Local Directed Acyclic Graph (LDAG) for LT model. Besides, Goyal et al. [7] proposed a SIMPATH algorithm for LT model, where a node's influence computation could directly be obtained from its outer neighbors. Since finding out all the simple paths is #P-hard and social network is well known small world network, Lu et al. [8] fixed the range of influence propagation within a certain number T and proposed efficient algorithm based on simple circles in social graph for $T \leq 4$ and the random walk for $T > 4$. Xu et al. [9] considered the characteristics of online social network, and therefore proposed a novel influence calculation model, which transformed IM problem into *Maximum Cut (MC)* problem. Though all these algorithms had good performance, they neglected that in real market strategy, serval kinds of products can be simultaneously promoted using the same social network.

Fan et al. [13] proposed two heuristic algorithms for two propagation model (IC and LT), namely Cluster-Based heuristic algorithm and Neighborhood-Removal heuristic algorithm. Different from the previous work, in [14], Fan et al. first proposed the J-Min-Seed problem, proved it is NP hard and proposed the corresponding greedy algorithm framework. Based on work in [14], [15] proposed the SM-PCG problem, which selected minimum nodes as seeds as long as the probability of their influence exceeding the fixed threshold. However, both the two problem in [14, 15] are particularly dependent on stochastic sampling, and therefore they had low efficiency to select the seed set.

Different from these work, [16] extended traditional IM problem to *Location-Aware Influence Maximization* in those social networks with geographic information, whose goal was to find the most k influential nodes in a given query area. However, in real social network, users have their own interests. Therefore Chen et al. [17–19] took *Topic* into consideration and proposed the *Topic-Aware Influence Maximization Query* problem. Li et al. [12] first applied *Conformity* related to social psychology to IM in online social networks. Besides, they also proposed corresponding *Conformity-Aware* cascade model. Meanwhile, considering the large number of nodes in real online social graphs, they proposed the graph partition-based CINEMA seed selection algorithm. Considering the competition in social network, Li et al. [20] analyzed the existence of Nash equilibrium adopting different IM strategies. However, they didn't solve the problem how to screen out the top k users for each company.

3. MULTIPLE INFLUENCE MAXIMIZATION PROBLEM (MIM)

In this section, we formulate the *Multiple Influence Maximization (MIM)* problem formally. Assume a company wants to promote m products in one social network and assign k items to a certain number of users. We construct the corresponding m social graphs with product p_i as $G_i = (V_i, E_i, W_i), 1 \leq i \leq m$, where element in W_i represents the influence probability on edges. Our goal is to find a seed set S from these m social graphs to maximize the overall influence spread (the number of nodes activated in the process of influence propagation). Let $S_i \subseteq V_i$ denote the seed set for product p_i in social graph G_i and $\sigma(S_i)$ represents its influence spread. We define influence spread of seed set S as the sum of influence spread for each seed set S_i, that is $\sigma(S) = \sum_{i=1}^{m} \sigma(S_i)$. Now we formulate the MIM problem.

DEFINITION 1 (MULTIPLE INFLUENCE MAXIMIZATION). *Given m social graphs $G_i = (V_i, E_i, W_i), 1 \leq i \leq m$ and budget k, the goal is to find a small set of nodes S^* consisting of S_i^* from each graph G_i such that*

$$S^* = \arg\max\{\sigma(S)\}$$

where $V_i = V_j$ $(\forall i \neq j)$, $S = \bigcup_{i=1}^{m} S_i$ and $\sum_{i=1}^{m} |S_i| = k$.

Note that the seed set S consists of seed set S_i of each graph G_i and therefore its number may smaller than the budget constraint k. This is because all the vertices in each graph G_i are the same and two sets S_i and S_j may intersect each other.

As traditional greedy framework is not suitable to solve the MIM problem, we propose a novel greedy framework named MIM-Greedy, which selects the seed node with highest maximum marginal gain. As the influence of seed set S can be calculated by $\sigma(S) = \sum_{i=1}^{m} \sigma(S_i)$, according to [12], the $\sigma(S)$ is submodular if each influence function $\sigma(S_i)$ for each graph G_i is submodular. Let $\Delta_v \sigma(S_i) = \sigma(S_i \cup v) - \sigma(S_i)$ denote the marginal gain of node v on seed set S_i of graph G_i. The goal of algorithm MIM-Greedy is to find a node v with maximum $\Delta_v \sigma(S)$, and fortunately this is equivalent to find a node v such that $\Delta_v \sigma(S_i) \geq \Delta_{v'} \sigma(S_j)$ for all $j \neq i$, where $v \in V_i$ and $v' \in V_j$. Simultaneously, on account of large number of nodes in social graph and to speed up the seed set selection when MIM-Greedy finds out a new seed node, it will not immediately update marginal gain of the rest of nodes but when needed. This framework is independent of influence calculation and propagation model.

In our MIM-Greedy algorithm, for each social graph G_i we use a Max Priority Queue P_i to save its vertices' marginal influence gain. Besides, we use another Max Priority Queue P to save each Queue's head element. For the convenience of algorithm design, we use two fields *value* and *source* to describe a node's marginal gain and which graph it belongs to. Besides, we adopt a general algorithm *Mar-Gain*(v, S_i, s_i^t) to compute the accurate marginal influence value of node v based on seed set S_i and the last seed node s_i^t of current seed sequence of S_i. In each step, algorithm selects the node with maximum marginal gain from Query P. Since this marginal gain may be outdated, it first need to judge whether the *value* of the head element is accurate. As shown in line 6-17 in Algorithm 1, when the sum of capacity for each seed set reaches to budget k, algorithm terminates and returns the final seed set S. Algorithm uses s and l to record the id of head element of P and the graph it belongs to, respectively. We also adopt field *state* to present the influence state of a node, true or false. If the state of s is *true* meaning that its *value* is accurate, it is deserved the seed node of G_l and we insert it into corresponding seed set S_l. Simultaneously, if it never appears in total seed set S, it should be added into S. On the contrary, if s's *state* is $false$, we must recompute its accurate marginal gain and reinsert it into P_l shown in Line 14 in the pseudo code. In Line 16, to ensure that P saves the elements with marginal gain of each Queue P_i, algorithm needs to insert the new head into P.

Algorithm 1 MIM-Greedy

Input: m social graphs $G_i = (V_i, E_i, W_i), 1 \leq i \leq m$ and the budget number k.
Output: The seed set S.
1: Initialize $S = \emptyset$ and $\forall i(1 \leq i \leq m), S_i = \emptyset$;
2: Initialize each Queue P_i with influence of each $v \in V_i$;
3: **for** $i = 1$ to m **do**
4:   $v = P_i.head()$; $P_i.delete(v)$;
5: **end for**
6: **while** $\sum_i^m |S_i| < k$ **do**
7:   $s = P.head()$; $l = s.source$; $P.remove(s)$;
8:   **if** $s.state = true$ **then**
9:     $S_l = S_l \cup \{s\}$; $s_l^t = s$
10:     **if** $s \notin S$ **then**
11:       $S = S \cup \{s\}$;
12:     **end if**
13:   **else**
14:     Mar-Gain(s, S_l, s_l^t); $P_l.insert(s)$;
15:   **end if**
16:   $v = P_l.head()$; $P.insert(v)$;
17: **end while**
18: **return** S;

In reality, we can renumber the vertices of all the m social graphs and hence we can conclude that Algorithm 1 can guarantee the approximate radio of $1 - 1/e$ if each influence function $\sigma(S_i)$ of seed set S_i is submodular according to [12]. After selecting the total seed set S and each set S_i for product p_i, the company will provide these users with products. If $s \in S$ and $s \in S_{i1}, S_{i2}, S_{ik}$, user s will obtain promotion of $p_{i1}, p_{i2}, \cdots, p_{ik}$ free of charge.

4. INFLUENCE COMPUTATION MODEL

In this section, we discuss how to compute the initial influence of single node in a social graph in Subsection 4.1 and the total influence gain of seed set S in Subsection 4.2.

4.1 Influence of a signal node

In majority of online social networks, influence propagates along social links. Whether a node will be activated not only depends on its neighbors's influence, but also the social sources and propagation paths [9]. Let $p_{u,v} = (u = v_1, v_2, \cdots, v_t = v)$ denote one of the paths from node u to v. Similar to [9], we set probability of influence from u to v along path $p_{u,v}$ as $P(p_{u,v}) = \prod_{i=1}^{t-1} p(v_i, v_{i+1})$. Generally, there are serval different paths from u to influence v and let $\mathcal{P}_{u,v}$ denote set of paths from u to v. In the process of influence transitivity, influence falls off with the increase of propagation hops along a certain social link. After the number of hop exceeds 6, the decay reaches its bottleneck and reduces slowly. To simplify influence computation of two nonadjacent nodes along social paths, we also use threshold θ to prune those paths whose influence are negligible just like [5]. We denote $P(u,v)$ as the influence of u to v along the paths whose influence is greater or equal to θ. Assume that the influence spreads independently among these paths, and thus we have

$$P(u,v) = 1 - \prod_{\substack{p_{u,v} \in \mathcal{P}_{u,v}, \\ P(p_{u,v}) \geq \theta}} (1 - P(p_{u,v}))$$

For those paths whose influence smaller than θ, we reset it to 0. Thus we can compute this influence for all pairs (u,v) to construct the influence matrix P. To sum up the influence of all paths reachable from u, we have the initial influence of node u presented in Equation (1).

$$\sigma(u) = P(u) = 1 + \sum_{v \in V \setminus \{u\}} P(u,v) \tag{1}$$

Note that the number 1 in Equation 1 represents the influence of u to itself, which has special significance in computing total influence of seed set S discussed Subsection 4.2. By this equation, we can obtain the initial influence of all nodes in a social graph, which makes a contribution to select the first seed node in the first round of iteration of algorithm MIM-Greedy.

4.2 Influence of Seed Set S

Given a social graph $G = (V, E)$, a seed set S and influence propagation model, to compute the influence spread of S, $\sigma(S)$ is well known open question in [3]. The most effective way up to now to tackle this problem is to sample a possible world X, an edge set chosen independently from edge set E of graph G, and compute the expected number of nodes reachable from S through world X. Similar to [7] and [9], let $P(S,v)$ be an indicator variable which equals to 1 if there is a simple path from S to v and 0 otherwise. Then the influence spread of S through world X, defined as $\sigma_X(S)$, can be written as:

$$\sigma_X(S) = \sum_{v \in V-S} Pr[X] \cdot P(S,v)$$

Where $Pr[X]$ represents the probability of X to be chosen, which can be calculated as $Pr[X] = \prod_{e \in X} P(e)$ as it is chosen independently. Repeating this sampling process to a enough large number of worlds and summing up these influence spread of S through all these worlds, we can obtain the influence spread of S as:

$$\sigma(S) = \sum_X \sum_{v \in V-S} Pr[X] \cdot P(S,v) + |S| \tag{2}$$

To compute the expectation of $P(S,v)$, we rearrange Equation (2) as:

$$
\begin{aligned}
\sigma(S) &= \sum_{v\in V-S}\sum_{X} Pr[X]\cdot P(S,v) + |S| \\
&= \sum_{v\in V-S} E[P(S,v)] + |S| \\
&= \sum_{v\in V-S} Pr[P(S,v)=1] + |S|
\end{aligned} \tag{3}
$$

The key issue to compute $\sigma(S)$ is how to compute the probability of v to be influenced by seed set S, namely $Pr[S,v]$. For discussion purpose, we use symbol '+', '-' to present the operation of add and deletion. Since node v has multiple different social paths from S, we legitimately assume that all these nodes in S affect v independently along those paths. Thus the probability of v influenced by S is $Pr[P(S,v)=1] = 1-\prod_{u\in S}(1-P(u,v))$, where $P(u,v)$ is the probability of v to be influenced by u presented in Subsection 4.1. Then substitute Equation (3) with $Pr[P(S,v)=1]$, we have

$$
\begin{aligned}
\sigma(S) &= \sum_{v\in V-S} Pr[P(S,v)=1] + |S| \\
&= \sum_{v\in V-S} (1-\prod_{u\in S}(1-P(u,v))) + |S| \\
&= |V| - \sum_{v\in V-S}\prod_{u\in S}(1-P(u,v))
\end{aligned} \tag{4}
$$

Thus according to Equation (4), the influence spread of S just relies on the probability of $u\in S$ to influence node v. Obviously, if the influence matrix P for each graph is computed in the offline phase, we can compute the influence spread of any seed set in polynomial time. Note that if there is only one node in S, the influence spread computed by Equation (4) is equal to that computed by Equation (1). Besides, this influence function $\sigma(S)$ has an important property described in the following theorem.

THEROEM 1. *Given social graph $G=(V,E)$, seed set S and influence matrix $P=\{P(u,v)|u,v\in V\}$, the influence function described in Equation (4) is submodular. That is, for any set $S\subseteq T$ and node $w\notin T$, we have $\Delta_w\sigma(S)\ge\Delta_w\sigma(T)$.*

PROOF. According to Equation (4), the influence spread of $S+w$ is $\sigma(S+w) = |V|-\sum_{v\in V-S-w}\prod_{u\in S+w}(1-P(u,v))$. We can rewrite it as

$$
\sigma(S+w) = |V| - \sum_{v\in V-S-w}(1-P(w,v))\prod_{u\in S}(1-P(u,v))
$$

Using this equation to subtract Equation 4, we can obtain the marginal gain for S, $\Delta_w\sigma(S)=\sigma(S+w)-\sigma(S)$, as:

$$
\begin{aligned}
\Delta_w\sigma(S) &= \sum_{v\in V-S}\prod_{u\in S}(1-P(u,v)) \\
&\quad - \sum_{v\in V-S-w}\prod_{u\in S+w}(1-P(u,v)) \\
&= \sum_{v\in V-S-w} P(w,v)\prod_{u\in S}(1-P(u,v)) \\
&\quad + \prod_{u\in S}(1-P(u,w))
\end{aligned} \tag{5}
$$

Similarly, we have the corresponding marginal gain for T, that is

$$
\begin{aligned}
\Delta_w\sigma(T) &= \sum_{v\in V-T-w} P(w,v)\prod_{u\in T}(1-P(u,v)) \\
&\quad + \prod_{u\in T}(1-P(u,w))
\end{aligned} \tag{6}
$$

Without loss of generality, let $T=S+\{u_1,u_2,\cdots,u_l\}$ and $T_{i+1}=T_i+u_i$, where $S=T_1, T=T_l$. Instead of proving $\Delta_w\sigma(S)\ge\Delta_w\sigma(T)$ directly, we prove that, for any indices $i, 1\le i\le l-1$, we have $\Delta_w\sigma(T_i)\ge\Delta_w\sigma(T_{i+1})$.

let $\gamma_i(w)=\Delta_w\sigma(T_i)-\Delta_w\sigma(T_{i+1})$, combining Equation (5) and Equation (6), we have

$$
\begin{aligned}
\gamma_i(w) &= \prod_{u\in T_i}(1-P(u,w)) - \prod_{u\in T_{i+1}}(1-P(u,w)) \\
&\quad + \sum_{v\in V-T_i-w} P(w,v)\prod_{u\in T_i}(1-P(u,v)) \\
&\quad - \sum_{v\in V-T_{i+1}-w} P(w,v)\prod_{u\in T_{i+1}}(1-P(u,v))
\end{aligned}
$$

As $T_{i+1}=T_i+u_i$, the item $\prod_{u\in T_{i+1}}(1-P(u,w))$ can be written as $(1-P(u_i,w))\prod_{u\in T_i}(1-P(u,w))$. Thus, the first item of $\gamma_i(w)$, marked as $\gamma_i^1(w)$, can be simply described as

$$
\gamma_i^1(w) = P(u_i,w)\prod_{u\in T_i}(1-P(u,w)) \tag{7}
$$

For the second item of $\gamma_i(w)$, substituting T_i with $T_{i+1}-u_i$, we have

$$
\begin{aligned}
&\sum_{v\in V-T_i-w} P(w,v)\prod_{u\in T_i}(1-P(u,v)) \\
&= \sum_{v\in V-T_{i+1}-w+u_i} P(w,v)\prod_{u\in T_i}(1-P(u,v)) \\
&= \sum_{v\in V-T_{i+1}-w} P(w,v)\prod_{u\in T_i}(1-P(u,v)) \\
&\quad + P(w,u_i)\prod_{u\in T_i}(1-P(u,u_i))
\end{aligned}
$$

We mark the last part of this equation as $\gamma_i^2(w)$, presented in Equation (8):

$$
\gamma_i^2(w) = P(w,u_i)\prod_{u\in T_i}(1-P(u,u_i)) \tag{8}
$$

Similarly, for the last item of $\gamma_i(w)$, substituting T_{i+1} with T_i+u_i, we have:

$$
\begin{aligned}
&\sum_{v\in V-T_{i+1}-w} P(w,v)\prod_{u\in T_{i+1}}(1-P(u,v)) \\
&= \sum_{v\in V-T_{i+1}-w} P(w,v)(1-P(u_i,v))\prod_{u\in T_i}(1-P(u,v))
\end{aligned}
$$

Thus, the result of substraction of the last two item of $\gamma_i(w)$ equals to $\gamma_i^2(w)$ plus $\gamma_i^3(w)$, defined in Equation (9):

$$
\gamma_i^3(w) = \sum_{v\in V-T_{i+1}-w} P(u_i,v)P(w,v)\prod_{u\in T_i}(1-P(u,v)) \tag{9}
$$

Therefore, the difference of marginal gain for T_i and T_{i+1} is equal to $\gamma_i(w)=\gamma_i^1(w)+\gamma_i^2(w)+\gamma_i^2(w)$. Since $P(u,v)\in[0,1]$ for all items in probabilistic matrix P, we have $\gamma_i^j(w)\ge$

$0, j = 1, 2, 3$. Consequently, the conclusion that for any indices i, $\gamma_i(w) \geq 0$ holds. That is to say, for any i, we have $\Delta_w\sigma(T_i) \geq \Delta_w\sigma(T_{i+1})$. Thus we can obtain $\Delta_w\sigma(S) = \Delta_w\sigma(T_1) \geq \Delta_w\sigma(T_2) \geq \cdots \geq \Delta_w\sigma(T_l) = \Delta_w\sigma(T)$, where $w \notin T$. The Theorem 1 holds. □

This model only relies on the construction of influence matrix P and has nothing to do with special *Influence Maximization* problem. In this paper, we use the equation in Subsection 4.1 to compute $P(u, v)$ and this process can be done in offline phase. Once the matrix P is built, we can apply it into the special *Influence Maximization* problem. Since $\sigma(\cdot)$ is submodular, if we apply it into our own MIM problem, Algorithm 1 has approximate radio of $1 - 1/e$.

Back to Equation (5), we note that the items $\prod_{u\in S}(1 - P(u, v))$ and $\prod_{u\in S}(1 - P(u, w))$ are both related to seed set S, which represents the probability of node v, w not to be influenced by S, respectively. This inspires us to find out an efficient way to further speed up the calculation of marginal gain for S. For discussion purposes, we denote $\Phi(S, v)$ be the probability that v is not influenced by S and let $S_i = \{s_1, s_2, \cdots, s_i\}$ be the seed sequence outputted by a certain greedy algorithm adopting our influence model. Since $\Phi(S_i, v) = \prod_{u\in S_i}(1 - P(u, v))$, we can use the Equation (10) to iteratively compute $\Phi(S_{i+1}, v)$. That is

$$\begin{aligned}\Phi(S_{i+1}, v) =& \prod_{u\in S_{i+1}} (1 - P(u, v)) \\ =&(1 - P(s_{i+1}, v)) \prod_{u\in S_i} (1 - P(u, v)) \\ =&(1 - P(s_{i+1}, v))\Phi(S_i, v)\end{aligned} \quad (10)$$

Therefore, we can rewrite the marginal gain of w for S presented in Equation (5) as:

$$\Delta_w\sigma(S) = \Phi(S, w) + \sum_{v\in V-S-w} P(w, v)\Phi(S, v) \quad (11)$$

If $\Phi(S, v)$ for any node v ($v \in V - S$) is known in advance, we can obtain the margin gain of any node w ($w \notin S$) for S in linear time by using Equation (11). Fortunately, this goal can indeed be achieved as long as we save the probability $\Phi(S_i, v)$ for each node v and update it using Equation (10). Similar to Algorithm 1, we use field *state* to describe whether a node's value of $\Phi(S_i, v)$ is real-time. The algorithm Mar-Gain describes the detail about how to update the probability $\Phi(S_{i+1}, v)$ and compute the marginal gain of w for S_{i+1}. Obviously, We can directly apply this algorithm into Algorithm 1.

5. EXPERIMENT

To evaluate our greedy framework and influence computation model, we apply this model into greedy framework and use the network Gnutella to do my experiments about the following two aspects (1) test the influence spread compared with traditional heuristic algorithms; (2) show the running time for serval kinds of products.

5.1 Experimental Step

Data Sets. The peer-to-peer network, Gnutella, is a kind of file sharing network, which owns more than 1.8 million nodes. In this paper, we directly download the part of data sets of this network from the website http://snap.stanford.edu/data/, which contains about 6.3K nodes and 21K edges. We use nodes to simulate the users, and edges to present social relationships in social network. As in our MIM problem, there must be multiple social graphs, thus we copy this topology and randomly generate weights for each graph. Besides, we use our own search algorithm with threshold $\theta = 0.1$ to compute a single node's influence and further to form probability matrix P (initial influence matrix) in offline phase. In our greedy framework coupled with influence model, influence matrix P plays a key role in updating the incremental influence gain of a node.

Algorithm 2 Mar-Gain (S_{i+1}, s_{i+1}, w)

Input: Current seed set S_{i+1}, the terminal seed node s_{i+1} of sequence S_{i+1}, non-seed node w.
Output: Marginal gain $\Delta_w\sigma(S_{i+1})$.

```
if w.state = false then
    for each v ∈ V − S_{i+1} do
        Φ(S_{i+1}, v) = (1 − P(s_{i+1}, v))Φ(S_i, v);
        v.state = true;
    end for
end if
Δ_wσ(S_{i+1}) = Φ(S_{i+1}, w);
for each v ∈ V − S_{i+1} − w do
    Δ_wσ(S_{i+1}) = Δ_wσ(S_{i+1}) + P(w, v)Φ(S_{i+1}, v);
end for
return Δ_wσ(S_{i+1});
```

Baselines. We adopt the following heuristic algorithms to select the seed set under our influence computation model as baselines to compare with Algorithm 1.

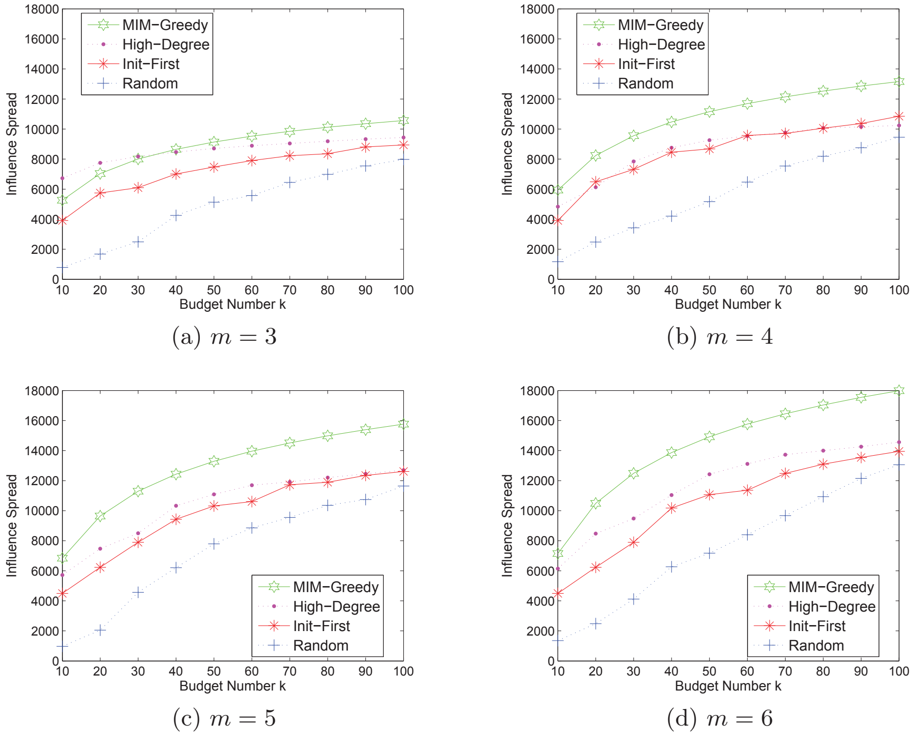
- **Random** [3,5,8,9,11,14,15]: As the baseline algorithm, it randomly selects k nodes from these m graphs.
- **High-Degree** [3,9,11,13–15]: The High-Degree algorithm selects the seed set with the decreasing order of degree for all vertices of m graphs.
- **Init-First**: Similar to traditional Distance Centrality algorithm, it selects the top k nodes with maximum initial influence value obtained from our offline phase among all these vertices of m graphs.

In the next subsection, we will present our experimental results by our influence model and greedy framework compared with above heuristic algorithms.

5.2 Experimental Results

In our experiments, we assume there are 3, 4, 5, and 6 kinds of products to be promoted and budget number k ranges from 10 to 100. For comparison purpose, the High-Degree, Init-First, and Random algorithm first screen out the seed set from the m social graph, and then adopt our influence computation model to compute its influence spread. Note that for the High-Degree algorithm, we calculate each node's degree of each graph and then sort all the nodes with degree values in these m graphs in advance. Thus, it can quickly find out the top k nodes.

Figure 2 shows the influence spread of the final seed set selected by the four algorithms. From these results, we can see that our MIM-Greedy framework outperforms all other heuristic algorithms for all categories m. The High-Degree ranks only second to MIM-Greedy and Init-First is close to



(a) $m = 3$ (b) $m = 4$

(c) $m = 5$ (d) $m = 6$

Figure 2: Influence Spread of Running algorithm: Random, MIM-Greedy, Init-First and High-Degree for different number of species: m

High-Degree for $m = 4, 5, 6$. As expected, the Random algorithm performs the worst. We can see from Figure 2, the influence spread of our MIM-Greedy nearly reaches to 11000 for $m = 3$, $k = 100$, which exceeds High-Degree about 2000, and Init-First about 3000. With the increase of the number of graphs, the influence spread of MIM-Greedy dramatically increase for each number k. However for other three algorithms, the corresponding increase is not particularly evident. For $m = 3$, 4, the influence spread of High-Degree almost keeps unchanged. Even though m changes from 5 to 6, the value only increase about 1000 and the same case can be found for Init-First. From Figure 2(d), the influence value of MIM-Greedy can reach 18000 when $m = 6$, $k = 100$, which fully proves that our work makes a lot of sense to the real marketing.

There is an obvious phenomenon that the increase rate of influence spread for MIM-Greedy, High-Degree and Iint-First changes slowly when number k reaches 50 for each graph. This is because all the three algorithms are extremely related to the network topology and edge weights. Our MIM-Greedy relies on the influence matrix's construction, which is extracted from the network topology. Similarly, the influence spread of Init-First is directly obtained from the influence matrix. For High-Degree, the higher the degree of a node, the more sources and paths it must have. When the number k reaches a certain threshold, the remaining nodes have less social source, in-neighbors and out-neighbors. Thus their influence value become relatively low. This results fully show us that our influence model and greedy framework are full of effectiveness.

Apart from testing the effectiveness of our MIM-Greedy,

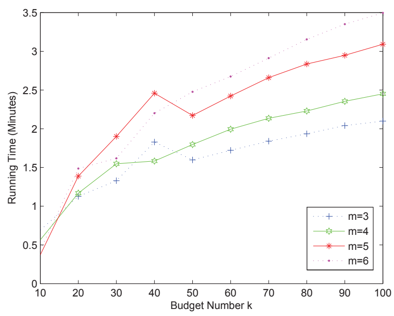


Figure 3: Running Time for MIM-Greedy when $m = 3, 4, 5, 6$ and k ranges from 10 to 100

we also record its running time as shown in Figure 3. We can see from the figure that when $k = 10$, it just needs about 0.4, 0.5, 0.6, 0.7 minutes for $m = 3, 4, 5, 6$ respectively. The running time increases with the growth of budget number k except when $m = 3, 5$ and $k = 40$. Even though when $k = 100$, selecting the seed set just takes nearly 1.6, 2, 3 and 3.5 minutes for $m = 3, 4, 5, 6$, respectively. This shows that our MIM-Greedy has good performance. As shown in Subsection 4.2, we use the Algorithm 2 to update a node's value of not to be influenced by seed set and marginal influence gain in linear time. Therefore, it can select k nodes within a few minutes. Thus we can conclude that our greedy framework and influence model have good effectiveness and efficiency.

6. CONCLUSION

In this paper, we consider the scenario that serval kinds of products can be promoted in the same social network simultaneously and propose the corresponding *Multiple Influence Maximization* problem (MIM). To effectively solve this problem, we propose the MIM-Greedy algorithm, which guarantees an approximate radio of $1 - 1/e$ if the influence function is submodular. We then discuss an influence computation model related to social source and social path. We prove that in this model, the influence function is submodular and the marginal influence gain can be obtained in linear lime, which can be seamlessly applied into the MIM-Greedy algorithm. We also validate the efficiency of our design by numerical experiments and comparisons with many other algorithms. In all, we are the first to consider the influence maximization problem with multiple products together and give efficient solutions.

7. ACKNOWLEDGMENTS

This work has been supported in part by the Natural Science Foundation of China (61202024, 61472252, 61133006, 61422208), China 973 project (2012CB316200, 2014CB340300), Shanghai Pujiang Program (13PJ1403900), the Natural Science Foundation of Shanghai (12ZR1445000), and Shanghai Educational Development Foundation (Chenguang Grant 12CG09), and in part by the Opening Project of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) and the Opening Project of Baidu.

8. REFERENCES

[1] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[2] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70, 2002.

[3] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Dining*, pages 137–146, 2003.

[4] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 946–957, 2014.

[5] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038, 2010.

[6] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *IEEE International Conference on Data Mining (ICDM)*, pages 88–97, 2010.

[7] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *IEEE International Conference on Data Mining (ICDM)*, pages 211–220, 2011.

[8] Zaixin Lu, Lidan Fan, Weili Wu, Bhavani Thuraisingham, and Kai Yang. Efficient influence spread estimation for influence maximization under the linear threshold model. *Computational Social Networks*, 1(1):1–19, 2014.

[9] Wen Xu, Zaixin Lu, Weili Wu, and Zhiming Chen. A novel approach to online social influence maximization. *Social Network Analysis and Mining*, 4(1):1–13, 2014.

[10] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *ACM SIGMOD International Conference on Management of Data*, pages 75–86, 2014.

[11] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208, 2009.

[12] Hui Li, Sourav S Bhowmick, Aixin Sun, and Jiangtao Cui. Conformity-aware influence maximization in online social networks. *The International Journal on Very Large Data Bases*, 24(1):117–141, 2015.

[13] Xiaoguang Fan and Victor Li. The probabilistic maximum coverage problem in social networks. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–5, 2011.

[14] Cheng Long and Raymond Chi-Wing Wong. Minimizing seed set for viral marketing. In *IEEE International Conference on Data Mining (ICDM)*, pages 427–436, 2011.

[15] Peng Zhang, Wei Chen, Xiaoming Sun, Yajun Wang, and Jialin Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1306–1315, 2014.

[16] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. Efficient location-aware influence maximization. In *ACM SIGMOD International Conference on Management of Data*, pages 87–98, 2014.

[17] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo A Baeza-Yates. Online topic-aware influence maximization queries. In *International Conference on Extending Database Technology (EDBT)*, pages 295–306, 2014.

[18] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. Online topic-aware influence maximization. *VLDB Endowment*, 8(6):666–677, 2015.

[19] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and Information Systems*, 37(3):555–584, 2013.

[20] Hui Li, Sourav S Bhowmick, Jiangtao Cui, Yunjun Gao, and Jianfeng Ma. Getreal: Towards realistic selection of influence maximization strategies in competitive networks. In *ACM SIGMOD International Conference on Management of Data*, pages 1525–1537, 2015.