A Fast Approximation for Influence Maximization in Large Social Networks

Jong-Ryul Lee
Dept. of Computer Science, KAIST
291 Daehak-ro, Yuseong-gu,
Daejeon, Korea
jrlee@islab.kaist.ac.kr

Chin-Wan Chung
Div. of Web Science and Technology
& Dept. of Computer Science, KAIST
291 Daehak-ro, Yuseong-gu,
Daejeon, Korea
chungcw@kaist.edu

ABSTRACT

This paper deals with a novel research work about a new efficient approximation algorithm for influence maximization, which was introduced to maximize the benefit of viral marketing. For efficiency, we devise two ways of exploiting the 2-hop influence spread which is the influence spread on nodes within 2-hops away from nodes in a seed set. Firstly, we propose a new greedy method for the influence maximization problem using the 2-hop influence spread. Secondly, to speed up the new greedy method, we devise an effective way of removing unnecessary nodes for influence maximization based on optimal seed's local influence heuristics. In our experiments, we evaluate our method with real-life datasets, and compare it with recent existing methods. From experimental results, the proposed method is at least an order of magnitude faster than the existing methods in all cases while achieving similar accuracy.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Influence Maximization; Independent Cascade; Social Networks

1. INTRODUCTION

Influence maximization, which is one of famous research problems related to viral marketing, has received great attention in recent years. In influence maximization, we want to find a k-seed set which maximizes the spread of influence in a social network for a given parameter k. A social network is represented by a graph where a node represents an individual and an edge represents a relationship between two individuals. In this work, influence propagation is modeled using the Independent Cascade (IC) model which is one of famous information diffusion models. In the IC model, if user u is influenced at time t, u has one-time chance to independently influence every uninfluenced neighbor v with some probability at time $t+1$. If u fails to influence v at time $t+1$, there is no further chance for u to influence v anymore. However, when user w is influenced at $t+1$ and there is the edge from w to v, w also has one-time chance to independently influence v at time $t+2$. Under the IC model, the influence spread is represented as the expected number of influenced users on a social network, and it is usually approximated by Monte-Carlo simulations, because it is #P hard to exactly compute the expected number of the influenced users[2].

Even if many methods are proposed for influence maximization, there are two critical obstacles to be overcome. The first obstacle is the expensive cost in calculating the influence spread of a seed set, and the second obstacle is a large number of users in a social network. In this paper, we focus on exploiting the 2-hop influence spread of a seed set to overcome the two obstacles. One may wonder whether considering only the 2-hop influence spread for influence maximization is valid. There is a line of research showing an interesting observation that an item is generally diffused from a seed within a very small number of hops in online social network services[1, 12, 10]. For example, [1] shows an observation that if a photo is uploaded in Flickr, more than 81% of users who participate in its diffusion are within 2-hops away from a seed. It means that even if we consider only users who are within 2-hops away from seeds to estimate the influence spread of the seeds, the estimated influence spread of the seeds is experimentally expected to be at least 81% of the exact influence spread. Therefore, exploiting the 2-hop influence spread is sufficiently valid to estimate the influence spread effectively.

Based on the concept of the 2-hop influence spread, we propose a fast greedy approximation method, for influence maximization, and a candidate extraction method filtering out uninfluential users from the entire users effectively. In the new greedy method, by exploiting the 2-hop influence spread, we work out efficient incremental updating of the marginal gain of our objective function. We address the first obstacle through the 2-hop influence spread and the incremental updating in the new greedy method. The candidate extraction method removes unnecessary users that are likely to be uninfluential by devising the Optimal Seed's Local Influence (OSLI) model. The OSLI model is motivated by the following idea:if a user can influence many other users in a network, the user is more likely to influence its neighbors. In other words, if a user is not likely to influence its neighbors with a high chance, we can consider that the user is

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2580063.

uninfluential and filter the user out because the user cannot influence many other users. Based on the OSLI model, the proposed method addresses the second obstacle in solving influence maximization. By handling the two obstacles efficiently, the proposed method, which contains the greedy method and the candidate extraction method, requires only a linear running time with respect to the number of users.

The contributions of our work are as follows:

- We propose an efficient Greedy method based on the 2-hop Influence Spread (GIS). GIS does not need any additional parameter.
- We propose an effective candidate extraction method filtering out unnecessary users. For influence maximization, the candidate extraction is the first approach that filters unnecessary users, to the best of our knowledge. We experimentally show that the candidate extraction effectively filters out unnecessary users and that it helps to greatly reduce the running time of GIS.
- We demonstrate that the proposed method is very efficient while achieving high accuracy using various real-life datasets. Compared to PMIA and IRIE, which are recent methods presented in [2] and [7], the proposed method is at least an order of magnitude faster in all cases.

The rest of this paper is organized as follows. In Section 2, we review the related works. We introduce influence maximization and the IC model in Section 3. In Section 4, the proposed method, which consists of the new greedy approximation and the candidate extraction, is developed. We demonstrate the effectiveness and efficiency of the proposed method through various experiments in Section 5. We make conclusions and outline future works in Section 6.

2. RELATED WORKS

Maximizing the profit of viral marketing is studied as an algorithmic problem by Domingos et al. [4] with a social network modeled as a Markov random field. Kempe et al. formulate the influence maximization problem as a discrete optimization problem and propose the basic greedy method [8]. However, the basic greedy method is not scalable for large social networks. To address the scalability issue of the basic greedy algorithm, many methods have been proposed. Leskovec et al. [11] improve the basic greedy method with a lazy-forward optimization in selecting new seeds. By exploiting submodularity, Goyal et al. [5] propose an improved greedy method, called CELF++. Chen et al. [3] propose a new greedy method based on generating random graphs to reduce the cost in computing influence spread. They also present degree discount heuristics based on the effective degree of a node given a seed set. The degree discount heuristic method is scalable for large social networks, but the accuracy of the algorithm is relatively low. Chen et al. [2] propose a greedy method, called PMIA, using the maximum influence arborescence model assuming that a seed node influences another through the maximum influence path from the seed node to the node. Wang et al. [13] propose a community-based greedy method using a heuristic that members in a community are more likely to influence each other. Jiang et al. [6] present the simulated annealing-based methods to overcome the confinement problem of greedy methods. Jung et al. [7] propose a new method for influence ranking using a system of linear equations, and introduce a method of utilizing their ranking method for influence maximization, called IRIE. Kim et al. [9] propose a parallel algorithm for influence maximization exploiting the concept of the independent influence path. We compare the proposed method with CELF++, PMIA, and IRIE, because they are sufficiently efficient or accurate to be compared with the proposed method.

3. PROBLEM DEFINITION

In this paper, a social network is represented as a directed graph $G = (V, E)$ where V is the set of nodes which represent users and E is the set of directed edges which represent relationships between users. For every pair $(i, j) \in V \times V$, we define the influence from i to j as the probability that i influences j through paths from i to j. For every pair $(u, v) \in V \times V$, we define $p(u, v)$ as the direct influence that is the probability that u influences v through edge $(u, v) \in E$. If edge (u, v) does not exist, then $p(u, v) = 0$. $p(u, v)$ does not contain any influence through another path from u to v. Since a path consists of multiple edges, the influence on a path can be considered as a series of the direct influences of edges in the path. Given path $\mathcal{P}$, the influence on path $\mathcal{P}$, denoted $p(\mathcal{P})$, is calculated as $p(\mathcal{P}) = \prod_{(u,v)\in\mathcal{P}} p(u, v)$. In addition, we assume that the direct influences are given.

IC model and influence maximization. For every node $i \in V$, let $n_{out}(i)$ denote the set of the out-degree neighbors of i and $n_{in}(i)$ denote the set of the in-degree neighbors of i. To describe influence spread under the IC model, let $S \subseteq V$ denote the seed set. For every seed $s \in S$, s is initially influenced at time 0. For $t \geq 0$, let $S_t \subseteq V$ denote the set of nodes which are influenced at time t. In the IC model, for every node $u \in S_t$, u may independently influence every uninfluenced neighbor $v \in n_{out}(u)$ with $p(u, v)$ at time $t+1$. If v is influenced at time $t + 1$, we insert v into S_{t+1}. After a node is influenced, the node stays as an influenced node. From the initial time 0 with $S_0 = S$, this spreading process runs iteratively until $S_{t'} = \emptyset$ for $t' \geq 0$. Given seed set S, the influence spread of S is the expected number of nodes influenced in the spread process including S.

The influence maximization problem under the IC model asks, for parameter k, seed set $S \subseteq V(|S| = k)$ which maximizes the influence spread of S. Kempe et al. prove that influence maximization under the IC model is NP-hard [8].

4. 2-HOP INFLUENCE SPREAD-BASED APPROXIMATION

4.1 Computing 2-hop Influence Spread

It is worth taking note that if a path from a seed node s to a node u has another seed s', s cannot influence u through the path because s' is already influenced. Thus, given seed set S, let $\Phi_S(s, u, d)$ denote the set of all paths $\mathcal{P}$ of length d from seed s to node u such that $\mathcal{P}$ does not have any seed as an intermediate node. Let $\Phi^*_S(s, u, d)$ denote $\bigcup_{1\leq i\leq d} \Phi_S(s, u, i)$. For any seed set S and any node $u \in V$, we define the d-hop influence from S to u as the probability that at least one of the seeds in S influences u along paths in $\bigcup_{s\in S} \Phi^*_S(s, u, d)$. We define the d-hop influence spread of S as the sum of the d-hop influences from S to nodes in V. In this work, we are interested in the case that $0 < d \leq 2$. Thus, let us denote the 2-hop influence spread of seed set S as σ_S. Specially, for every node $u \in V$, let us denote

the 1-hop influence spread of node u (i.e., single node set) as σ_u^1. By definition, $\sigma_u^1 = 1 + \sum_{c \in C_u} p(u,c)$. In addition, for every node $u \in V$, we define the 1-hop influenced cover of u, denoted as C_u, as the set of the out-degree neighbors of u. We define also the 2-hop influenced cover C_u^* of u as $C_u^* = \bigcup_{c \in C_u} C_c - \{u\}$. C_u^* specifies the region defined by the set of all nodes that are directly influenced by the nodes in C_u and indirectly influenced by u. Let us define the 2-hop influenced region of u, denoted as $V_u^* = C_u^* \cup C_u \cup \{u\}$, as the region influenced by u within 2-hops.

The 2-hop influence spread of a seed set. To estimate the 2-hop influence spread of a seed set, we exploit an interesting relationship between two paths in a graph, which is the independence between paths. If two paths have the same destination node and there is no overlapping intermediate node except for the source and the destination, we say that they are independent of each other. Note that for every pair $(u,v) \in V \times V$ such that all paths from u to v are independent of each other, the influence from u to v is calculated to be $1 - \prod_{\mathcal{P} \in \Phi(u,v)}(1 - p(\mathcal{P}))$, where $\Phi(u,v)$ is the set of all paths from u to v. Using this relationship, we estimate the 2-hop influence spread as follows.

Given seed set $S \subseteq V$, for every node $u \in V$, let $p_d(S,u)$ denote the d-hop influence from S to u. In addition, let $p_2(S,u,c)$ denote the 2-hop influence from S to u via node c, which is one of the immediate predecessors of u. By definition, $p_2(S,u,c) (= p_1(S,c)p(c,u))$ is computed as,

$$p_2(S,u,c) = \left(1 - \prod_{s \in S}(1 - p(s,c))\right) p(c,u). \tag{1}$$

Since we consider the 2-hop influence spread, by assuming that all 2-hop paths from the seeds in S to u via c are independent of each other, $p_2(S,u,c)$ is estimated to be,

$$\hat{p}_2(S,u,c) = 1 - \prod_{s \in S}(1 - p(s,c)p(c,u)). \tag{2}$$

Let us verify our estimate for $p_2(S,u,c)$. Let ω_s denote $p(s,c)$ and β denote $p(c,u)$ for abbreviation. The error of $\hat{p}_2(S,u,c)$ is close to 0 when direct influences between nodes are small, because, $\lim_{\omega_s, \beta \to 0} \left(1 - \prod_{s \in S}(1 - \omega_s)\right)\beta - \left(1 - \prod_{s \in S}(1 - \omega_s \beta)\right) = 0$. Direct influences are usually very small in social networks, so the error of $\hat{p}_2(S,u,c)$ must be very small.

By exploiting the concept of the independence between paths and $\hat{p}_2(S,u,c)$, we can get a good estimate for $p_2(S,u)$. If all paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$ are independent of each other, we can easily compute the exact value of $p_2(S,u)$ using the independence between paths. It is easy to see that all 1-hop paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$ are independent of any 2-hop path in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$. As we verified for (2), it is reasonable to suppose that paths in $\bigcup_{s \in S} \Phi_S(s,u,2)$ sharing an intermediate node are independent of each other. The other paths, in $\bigcup_{s \in S} \Phi_S(s,u,2)$, which do not share any intermediate node are independent of each other by definition. Thus, we can suppose that all paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$ are independent of each other with a very small error. The error caused by this assumption is close to 0 when direct influences are very small. Therefore, $p_2(S,u)$ is estimated to be,

$$\hat{p}_2(S,u) = 1 - \prod_{s \in S}\left(1 - \left(1 - \prod_{\mathcal{P} \in \Phi_S^*(s,u,2)}(1 - p(\mathcal{P}))\right)\right).$$

$\hat{p}_2(S,u)$ is a reasonable estimate for $p_2(S,u)$, but all paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$ should be enumerated to compute $\hat{p}_2(S,u)$. Thus, we need to estimate $\hat{p}_2(S,u)$ again for efficiency. Let us consider seed set S, seed node $s \in S$, and any node $u \in V$. For any path $\mathcal{P} \in \bigcup_{s \in S} \Phi_S^*(s,u,2)$, $p(\mathcal{P})$ is estimated to be θ_u which is the average of the influences on paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$. Since direct influences are usually very small in social networks[1, 10, 12], the error of our estimate for $p(\mathcal{P})$ should be small. Therefore, our estimate for $\hat{p}_2(S,u)$ is provided by changing $p(\mathcal{P})$ in $\hat{p}_2(S,u)$ to θ_u.

For seed set $S \subseteq V$, our estimate for $p_2(S,u)$ directly provides a good estimate for $\sigma_S = \sum_{u \in V} p_2(S,u)$, which is,

$$\hat{\sigma}_S = k + \sum_{u \in V \setminus S}\left(1 - (1 - \theta_u)^{d_u}\right), \tag{3}$$

where d_u is the number of paths in $\bigcup_{s \in S} \Phi_S^*(s,u,2)$. For efficiency, we use a linear approximation for (3) according to Taylor's theorem. The linear approximation states that $f(x) \approx f(a) + f'(a)(x-a)$ (if x is close to a), where $\approx$ is a binary operator that the right operand goes to the left operand as a variable shared by the two operands goes to some value. Then, for $0 \le \theta'_u \le 1$, if θ_u is close to θ'_u,

$$\hat{\sigma}_S \approx k + \sum_{u \in V \setminus S} f(\theta_u, \theta'_u), \tag{4}$$

where $f(\theta_u, \theta'_u) = 1 - (1 - \theta'_u)^{d_u} + d_u(1 - \theta'_u)^{d_u - 1}(\theta_u - \theta'_u)$,

$$= k + \sum_{u \in V \setminus S} d_u \theta_u (\text{by setting } \theta'_u = 0) \tag{5}$$

$$= k + \sum_{s \in S} \sum_{u \in V \setminus S} \sum_{\mathcal{P} \in \Phi_S^*(s,u,2)} p(\mathcal{P}) \tag{6}$$

$$= k + \sum_{s \in S} \sum_{c \in C_s \setminus S} p(s,c)\left(1 + \sum_{d \in C_c \setminus S} p(c,d)\right) \tag{7}$$

$$= k + \left(\sum_{s \in S} \sum_{c \in C_s \setminus S} p(s,c)(\sigma_c^1 - p(c,s))\right) - \chi, \tag{8}$$

where $\chi = \sum_{s \in S} \sum_{c \in C_s \setminus S} \sum_{d \in C_c \cap S \setminus \{s\}} p(s,c)p(c,d)$,

$$= \sum_{s \in S} \hat{\sigma}_{\{s\}} - \left(\sum_{s \in S} \sum_{c \in C_s \cap S} p(s,c)(\sigma_c^1 - p(c,s))\right) - \chi. \tag{9}$$

In (5), we set $\theta'_u = 0$ according to the linear approximation. It means that (5) is close to (3) by the linear approximation when θ_u is close to 0. In social networks, θ_u is close to 0, so our linear approximation is valid. (6) is derived, because $\theta_u = \frac{\sum_{s \in S} \sum_{\mathcal{P} \in \Phi_S^*(s,u,2)} p(\mathcal{P})}{d_u}$. Since we consider the 2-hop influence spread, we only need to take nodes within 2-hops from each seed s, which are in C_s^*. Thus, (9) is derived from (6). In (9), the case that a seed is an out-degree neighbor of another seed is considered in the second term, and the case that a seed is 2-hops away from another seed is considered in the third term.

4.2 Greedy Efficient Approximation

The bottleneck of the basic greedy algorithm is to compute the marginal gain of a new seed with respect to influence spread. To address the bottleneck, we use the 2-hop

influence spread of a seed set as an objective function and devise a novel way of incrementally updating the objective function. Let us denote $S \cup \{u\}$ as S_u. For any seed set $S \subseteq V$ and any node $u \in V$ such that $u \notin S$, we estimate $\sigma_{S,u} = \sigma_{S_u} - \sigma_S$ as,

$$\hat{\sigma}_{S,u} = \hat{\sigma}_{S_u} - \hat{\sigma}_S = \hat{\sigma}_{\{u\}} - \sum_{c \in C_u \cap S_u} p(u,c)(\sigma_c^1 - p(c,u)) - \sum_{i \in S} p(i,u)(\sigma_u^1 - p(u,i)) - \sum_{c \in C_u \setminus S_u} \sum_{d \in C_c \cap S} p(u,c)p(c,d) - \sum_{i \in S} \sum_{c \in C_i \setminus S_u} p(i,c)p(c,u) + \sum_{i \in S} \sum_{d \in C_u \cap S \setminus \{i\}} p(i,u)p(u,d).$$

In our expression for $\hat{\sigma}_{S,u}$, there are five terms each of which consists of one or multiple summations. The first and second terms come from the second term in (9). The third, fourth, and fifth terms come from the third term in (9). However, it is too expensive to compute $\hat{\sigma}_{S,u}$ for every node $u \in V$ whenever new seed s is inserted into S in a greedy method. Thus, we use $\hat{\sigma}_{S,u,s} = \hat{\sigma}_{S_s,u} - \hat{\sigma}_{S,u}$ to incrementally update $\hat{\sigma}_{S,u}$.

$$\begin{aligned}\hat{\sigma}_{S,u,s} &= \hat{\sigma}_{S_s,u} - \hat{\sigma}_{S,u} \\ &= -p(u,s)(\sigma_s^1 - p(s,u)) - p(s,u)(\sigma_u^1 - p(u,s)) \\ &- \sum_{c \in C_u \setminus (S_u \cup \{s\})} p(u,c)p(c,s) + \sum_{d \in C_s \cap S} p(u,s)p(s,d) \\ &- \sum_{c \in C_s \setminus (S_u \cup \{s\})} p(s,c)p(c,u) + \sum_{i \in S} p(i,s)p(s,u) \\ &+ \sum_{d \in C_u \cap S_s \setminus \{s\}} p(s,u)p(u,d) + \sum_{i \in S} p(i,u)p(u,s).\end{aligned}$$

Based on the incremental update of $\hat{\sigma}_{S,u}$ using $\hat{\sigma}_{S,u,s}$, we build a greedy method, denoted as GIS, which is described in Algorithm 1. In Line 2, S is initialized, and in Lines 3-6, the 1-hop influence spread of every node and the 2-hop influence spread of every node set of size 1 in V are computed. It is easy to see that Lines 3-6 can be efficiently implemented with two scans on the node set V. In Lines 7-24, we pick k seeds greedily to maximize $\hat{\sigma}_{S_u} - \hat{\sigma}_S$ per iteration. After picking node s as a seed in Line 8, we need to update $\hat{\sigma}_{S,u}$ for each $u \in V$ because S has been changed. In Lines 14,16,22, and 24, we update $\hat{\sigma}_{S,u}$ with $\hat{\sigma}_{S,u,s}$, for every node u such that u is within 2-hops away from s in inbound or outbound direction. In Lines 13,15,21, and 23, the commented numbers indicate the terms of $\hat{\sigma}_{S,u,s}$ involved in each update.

Analysis for time complexity. GIS requires only $O(nd)$ time, where $n = |V|$ and d is the average out-degree, to compute the 1-hop influence spread of every node and the 2-hop influence spread of every node set of size 1 in V. In GIS, picking k seeds greedily requires $O(kd^2 \log n)$ time with a priority queue. The total time complexity for GIS is $O(nd + kd^2 \log n)$, and it is better than that of PMIA and IRIE.

4.3 Effective Candidate Extraction

Optimal Seed's Local Influence (OSLI) heuristics. For every node $u \in V$, we define the Most Influential (MI) node, denoted as $MI(u)$, as the node in V_u^* that is included in C_u and that has the strongest 1-hop influence spread, on nodes in V_u^*, which is larger than $\sigma_{\{u\}}$. If there is no node in C_u that has 1-hop influence spread larger than $\sigma_{\{u\}}$, let $MI(u)$ be u. For any node $c \in C_u^*$ such that $c \notin C_u$, since

Algorithm 1: 2-hop Greedy Algorithm $(G = (V, E), k)$

```
   input  : G: An input graph, k:size of a seed set
   output : S : Output seed set
1  begin
2      S = ∅;
3      for u ∈ V do
4          compute σ_u^1;
5      for u ∈ V do
6          compute σ̂_{u};
7      for i = 1 to k do
8          s = arg max_{u∈V} σ̂_{S,u}, S = S ∪ {s} ;
9          for u ∈ n_in(s) do
10             if u ∉ S then
11                 for v ∈ n_in(u) do
12                     if v ∉ S then
13                         //(3)
14                         σ̂_{S,v} = σ̂_{S,v} − p(v,u)p(u,s);
15                 // (1,4,8)
16                 σ̂_{S,u} = σ̂_{S,u} − p(u,s)(σ_s^1 − p(s,u)) +
                   Σ_{d∈C_s∩S\{s}} p(u,s)p(s,d) +
                   Σ_{i∈S\{s}} p(i,u)p(u,s);
17         for u ∈ n_out(s) do
18             if u ∉ S then
19                 for v ∈ n_out(u) do
20                     if v ∉ S then
21                         //(5)
22                         σ̂_{S,v} = σ̂_{S,v} − p(s,u)p(u,v);
23                 //(2,6,7)
24                 σ̂_{S,u} = σ̂_{S,u} − p(s,u)(σ_u^1 − p(u,s)) +
                   Σ_{i∈S\{s}} p(i,s)p(s,u) +
                   Σ_{d∈C_u∩S\{s}} p(s,u)p(u,d);
25     return S ;
```

σ_c^1 can include direct influences from c to nodes that are not in V_u^*, c is ignored when finding $MI(u)$.

Let α^* denote the maximum number of out-degree neighbors of all the nodes in V. When the 1-hop influence spread of a node is larger than or equal to α where $1 \leq \alpha \leq \alpha^* + 1$, the 2-hop influenced region of the node is called the effective 2-hop influenced region. Then, for any seed s in the optimal seed set, the Optimal Seed's Local Influence (OSLI) heuristics are as follows. Firstly, σ_s^1, which represents the degree to which s influences the nodes in C_s, is likely to be larger than or equal to α. Secondly, seed node s is likely to be the MI node in at least one of the effective 2-hop influenced regions in which s participates.

By definition, s should influence the nodes in C_s first in order to influence many other nodes in the network. If none of the nodes in C_s is influenced, there is no further chance for s to influence the other nodes in the network. That is the motivation of the first OSLI heuristic.

Let us see how the second heuristic works. For every node $u \in V$, when we find $MI(u)$, we compare $\sigma_{\{u\}}$ and all σ_c^1 such that $c \in C_u$. As a result, there are the two cases for $MI(u)$.

- **case 1** ($MI(u) = u$) In this case, u is the MI node in V_u^*. It means that there is no node c in V_u^* such that $\sigma_{\{u\}} < \sigma_c^1$, so nothing has been determined.
- **case 2** ($MI(u) \neq u$) In this case, u is not the MI node in V_u^*, but there is another node $MI(u)$. It means that

there is node c in V_u^* such that $c = MI(u)$ and $\sigma_{\{u\}} < \sigma_c^1 \leq \sigma_{\{c\}}$.

Let $deg_{in}(u)$ denote the in-degree of u. If u satisfies the first heuristic, there is one chance for u to be the MI node in V_u^*. There are $deg_{in}(u)$ chances for u to be a MI node in the effective 2-hop influenced regions in which u participates, except V_u^*. The reason we additionally give u the $deg_{in}(u)$ chances is that even if u is not the MI node in V_u^*, u can be a MI node in another effective 2-hop influenced region. If u misses all the chances, we filter out u from our candidate list for optimal seeds, because there is always another node v in the candidate list such that $\sigma_v^1 > \sigma_u^1$, or even $\sigma_v^1 > \sigma_{\{u\}}$. The meaning of $\sigma_v^1 > \sigma_{\{u\}}$ is that v is likely to have more influence to nodes which are connected from u. Thus, it is reasonable to exclude u when u misses all the chances to be a MI node. That is why the second OSLI heuristic works.

Candidate Extraction. Based on the OSLI heuristics, the proposed method extracts candidates that are not likely to be uninfluential. This candidate extraction procedure consists of the following two steps. The first step is to filter out unnecessary nodes that have 1-hop influence spread smaller than α. It is based on the first OSLI heuristic. Next, the second step is to filter out nodes which miss all the chances to be a MI node based on the second OSLI heuristic.

It is easy to implement this procedure in $O(nd)$ time with looking all nodes in V two times, because the candidate extraction is accomplished with the 1-hop influence spread of every node and the 2-hop influence spread of every node set of size 1, and they can be computed in $O(nd)$ time as we mentioned.

5. EXPERIMENTS

In these experiments, we run the experiments on an Intel(R) i7-990X 3.46 GHz CPU machine with 24GB RAM.

5.1 Experimental Environment

Comparison methods. In the experiments, let us denote the final proposed method including GIS and the candidate extraction as OGIS. In addition, comparison methods are as follows. CELF++ is an improved greedy algorithm proposed in [5]. For CELF++, the number of Monte-carlo simulations is set to 10000. OCELF++ is CELF++ using the candidate extraction. PMIA is a greedy method using maximum influence paths between nodes[2]. IRIE is one of recent algorithms for influence maximization [7]. In PMIA and IRIE, θ determines the maximum length of maximum influence paths. We use the setting of [7] for θ. For datasets that are not introduced in [7], we determine θ experimentally. For IRIE, as the authors in [7] did, we set $\alpha = 0.7$ which is a damping factor, but α in this paper is used as a parameter tin the first OLSI heuristic. Finally, Random is a method which picks seeds randomly.

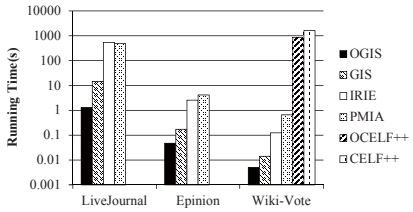
Table 1: Statistics of our datasets

Dataset	Wiki-Vote	Epinions	LiveJournal
Node	7.1K	75.8K	4,847.6K
Edge	103.6K	508.8K	68,993.8K
Avg. Degree	29.1	13.4	28.5

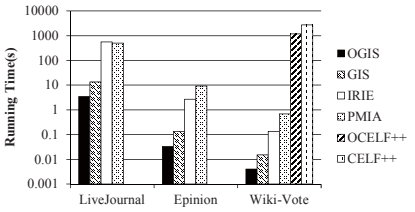
Datasets. We use three real datasets:Wiki-Vote, Epinions, and LiveJournal. They are published online by Jure Leskovec (http://snap.stanford.edu/data/). Wiki-Vote is a social network based on the elections for promoting adminship, in which a directed edge from u to v represents user u voted for user v. Epinions is a who-trust-whom online social network. LiveJournal is a free-online social network and allows members to maintain journals and blogs. Table 1 shows the statistics of the three datasets.

Table 2: α along datasets and influence models

Dataset	Wiki-Vote	Epinions	LiveJournal
WC	1.5	3.4	5.0
UP	1.4	1.2	5.0



(a) UP, $k = 50$



(b) WC, $k = 50$

Figure 1: Running time of the algorithms with $k = 50$

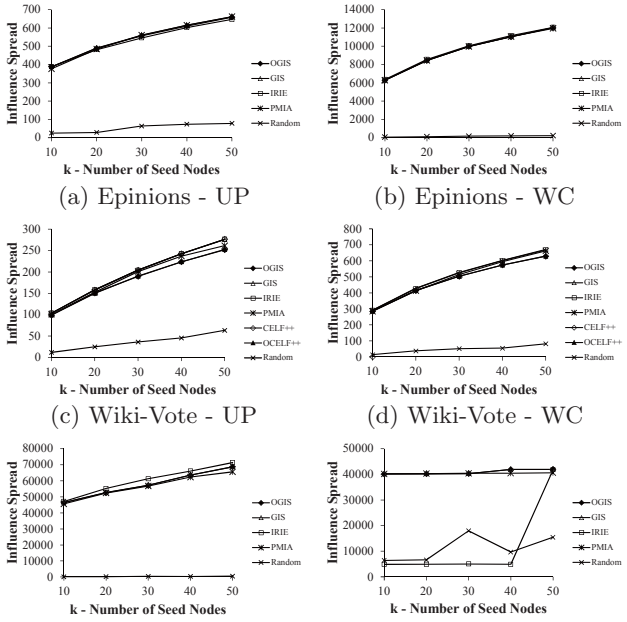
Direct influence model. To model direct influences, we use the uniform probability model and the weighted cascade model. The uniform probability model states that all direct influences are equal to p ($0 \leq p \leq 1$). In our experiments, we set $p = 0.01$. The weighted cascade model states that for every node $v \in V$, the direct influence from an in-edge neighbor of node v to v is equal to $1/(|n_{in}(v)|)$ [8]. In these experiments, the uniform probability model and the weighted cascade model are denoted as UP and WC, respectively.

5.2 Experiment Results

Table 2 illustrates the values of α used in these experiments. We experimentally determine the values. For PMIA and IRIE, we set θ as 0.00999 for all datasets in UP, 0.00665 for Wiki-Vote and 0.00625 for the other datasets in WC. In addition, we compare OGIS and GIS with CELF++ and OCELF++ in only Wiki-Vote, because they are too slow in the other datasets.

Running time. Figure 1 illustrates the running time of each method when $k = 50$. In this experiment, we observe that OGIS and GIS are much faster than PMIA, IRIE, CELF++, and OCELF++. Especially, OGIS is at least an orders of magnitude faster than PMIA and IRIE in all cases. In addition, OGIS and OCELF++ are much faster than GIS and CELF++, respectively. These results clearly show the effect of the candidate extraction on running time. The running time of Random is negligible.

Influence Spread. Figure 2 shows the results about influence spread achieved by each method. In this experiment, all the comparison methods achieve similar influence spread over all datasets except Random. Recall that OGIS



(a) Epinions - UP (b) Epinions - WC

(c) Wiki-Vote - UP (d) Wiki-Vote - WC

(e) LiveJournal - WC (f) LiveJournal - UP

Figure 2: Influence spreads on the three datasets

is much faster than the other comparisons. Despite the efficiency of OGIS, OGIS achieves influence spread similar to those of CELF++, PMIA and IRIE. Meanwhile, the influence spreads of IRIE are very low for $k = 10$ to $k = 40$ in LiveJournal. One possible explanation is that IRIE may find poor seeds, in a single dense community, each of which has a big influence spread but shares many out-degree neighbors with the other seeds.

Based on these results, we demonstrate that OGIS is much more efficient than PMIA and IRIE while achieving the similar influence spread. In addition, we show that GIS and the candidate extraction successfully address the obstacles which we mentioned in Section 1.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, based on the 2-hop influence spreads, we propose a new efficient greedy method and an effective candidate extraction method for influence maximization. For the new greedy method, we exploit our estimate for the 2-hop influence spread of a seed set to update the marginal gains of the objective function efficiently. The candidate extraction is the first approach filtering unnecessary nodes for influence maximization. We experimentally demonstrate that the candidate extraction can effectively filter out unnecessary nodes and the proposed method is at least an order of magnitude faster than PMIA and IRIE while achieving similar accuracy.

We will apply that the techniques proposed in this paper can be applied to other influence models for influence maximization. In addition, we will devise new variations of influence maximization for more effective viral marketing and apply the proposed techniques to the new variations.

7. ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea grant funded by the Korean government (MSIP) (No. NRF-2009-0081365).

8. REFERENCES

[1] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 721–730, 2009.

[2] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, 2010.

[3] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, 2009.

[4] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, 2001.

[5] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, 2011.

[6] Q. Jiang, G. Song, C. Gao, Y. Wang, W. Si, and K. Xie. Simulated annealing based influence maximization in social networks. In *AAAI Conference on Artificial Intelligence*, 2011.

[7] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 918–923, 2012.

[8] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, 2003.

[9] J. Kim, S.-K. Kim, and H. Yu. Scalable and parallelizable processing of influence maximization for large-scale social networks? In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 266–277, April 2013.

[10] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, 2010.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, 2007.

[12] E. Sun, I. Rosenn, C. Marlow, and T. Lento. Gesundheit! modeling contagion through facebook news feed. *Proc. ICWSM*, 9, 2009.

[13] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, 2010.