Information Sciences 355–356 (2016) 110–126

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Finding top-k influential users in social networks under the structural diversity model☆

Wenzheng Xu[a,*], Weifa Liang[b], Xiaola Lin[c], Jeffrey Xu Yu[d]

[a] College of Computer Science, Sichuan University, Chengdu, 610065, PR China
[b] Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia
[c] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, 510006, PR China
[d] Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 24 June 2015
Revised 10 March 2016
Accepted 16 March 2016
Available online 24 March 2016

Keywords:
Influence maximization
Structural diversity model
Social networks
Approximation algorithm

ABSTRACT

The influence maximization problem in a large-scale social network is to identify a few influential users such that their influence on the other users in the network is maximized, under a given influence propagation model. One common assumption adopted by two popular influence propagation models is that a user is more likely to be influenced if more his/her friends have already been influenced. This assumption recently however was challenged to be over simplified and inaccurate, as influence propagation process typically is much more complex than that, and the social decision of a user depends more subtly on the network structure, rather than how many his/her influenced friends. Instead, it has been shown that a user is very likely to be influenced by structural diversities of his/her friends. In this paper, we first formulate a novel influence maximization problem under this new structural diversity model. We then propose a constant approximation algorithm for the problem. We finally evaluate the effectiveness of the proposed algorithm by extensive experimental simulations, using different real datasets. Experimental results show that the users identified from a social network by the proposed algorithm have much larger influence than that found by existing algorithms.

1. Introduction

The last decade experienced the exponential growth of a variety of online social networks such as Facebook, Twitter, LinkedIn, etc [20]. A recent snapshot of the friendship network Facebook showed that there are over 1 billion users in it [14]. Not only are these social networks effective tools for people to connect their friends and share interests and backgrounds, but also they now become powerful information dissemination and marketing platforms to allow information, ideas, fads, and political opinions to spread to a large population economically via the so called "word-of-mouth" exchanges [1,2,15,23,25,31,32,34,35]. For example, suppose that one company would like to market a new product and hopes the product will be adopted by a large fraction of users in a social network. To market the product economically, the company initially targets a few "influential" users in the network by giving them free product samples. These users then recommend the product to their friends, and some of their friends will accept the product and recommend the product to their friends,

☆ This document is a collaborative effort.

* Corresponding author. Tel.: +86 13880337745.
E-mail addresses: wenzheng.xu3@gmail.com (W. Xu), wliang@cs.anu.edu.au (W. Liang), linxl@mail.sysu.edu.cn (X. Lin), yu@se.cuhk.edu.hk (J.X. Yu).

http://dx.doi.org/10.1016/j.ins.2016.03.029
0020-0255/

and so on. As a result, this triggers a cascade of influence propagation and many users in the network will accept the product ultimately [21]. This marketing strategy usually is referred to as *viral marketing*. One well-known real story of viral marketing is the commercial success of the Hotmail company in the early 1990s, which made the company become the number-one e-mail provider within only 18 months [19]. Thus, a fundamental research topic in social network sciences is to effectively and efficiently identify a very few influential individuals in a large-scale social network for information diffusion.

In their seminal paper, Kempe et al. [21] formalized information diffusion in a large social network as the *influence maximization problem*, which is defined as follows. Given an integer k (as the budget), the problem is to find k "seeds" (i.e., source nodes) in the network such that the expected number of activated nodes eventually by these k seeds is maximized, assuming that an influence propagation model is given, where the activation of a node means that the node accepts the recommendation. Following their work, several studies have been conducted in the past several years [3–10,16–18,21,22,30,33]. Most of these studies adopt the two popular influence propagation models: *the independent cascade model* and *the linear threshold model* [8,21]. One common property of these two models is that for a given user, the more his/her friends recommend a product to the user, the more likely the user will accept the product and recommend the product to his/her friends. This influence propagation is similar to the epidemic diseases spread, i.e., the probability of a user being influenced monotonically grows with the number of his/her friends whom have already been affected. However, these two models have recently been challenged by Ugander et al. [28] to be over simplified and thus inaccurate, as influence propagation process typically is more complex, and the social decision of a user depends more subtly on the network structure, rather than how many his/her influenced friends. Instead, they studied two influence propagation processes in Facebook: the process whereby a person joins Facebook in response to an invitation email from an existing Facebook user; and the process that a user becomes an engaged user after joining Facebook. They found through empirical analysis that the chance of a user accepting a recommendation is positively correlated with the number of connected components in the induced graph by the neighbors, rather than the number of neighbors, of the user, where each connected component represents a distinct social context of the user in the network and the multiplicity of social contexts is referred to as *the structural diversity*. They showed that a person is more likely to join Facebook if he/she receives more invitation emails from his/her friends with distinct social contexts, e.g., families, workmates, classmates, etc. On the other hand, surprisingly, they also demonstrated that once the number of connected components is controlled (or fixed), a person is less likely, rather than more likely, to join Facebook if more friends invite him/her. They concluded neither the number of friends inviting the user nor the number of connections among his/her friends will determine his/her acceptance probability. Instead, it is the number of connected components (structural diversity) derived by his/her neighbors that captures his/her acceptance probability. We term this model as the structural diversity model, which has been empirically demonstrated to be able to accurately capture the propagation progress of influence.

Motivated by the seminal work of Ugander et al. [28], in this paper we study the influence maximization problem under the structural diversity model, where the decision of a user depends on different groups of neighbors, rather than the number of neighbors. It thus poses a challenging problem, that is, how to incorporate the structural diversity of each user into the influence maximization problem. Existing algorithms for the problem under the two widely adopted influence propagation models thus are not applicable, and new algorithm is desperately needed. In this paper, we propose a probabilistic approximation algorithm for the problem under the structural diversity model. Interestingly, we show that the problem is equivalent to a slightly different influence maximization problem in an auxiliary graph under the independent cascade model. The contributions of the paper can be summarized as follows.

- We first formulate a novel influence maximization problem under the structural diversity model, in which a user is more likely to accept a recommendation if more his/her friends with distinct social contexts make the recommendation to the user.
- We then devise a $(1-\frac{1}{e}-\epsilon)$-approximation algorithm with probability $1-\alpha$ for the problem, which takes $O(\epsilon^{-2}\alpha^{-1}n^2(m+n)k^3\log k)$ time, improving the time complexity $O(\epsilon^{-2}n^3(m+n)k^3\log\frac{nk}{\alpha})$ of the state-of-the-art approximation algorithm by a factor of αn [8] (pp.43), where e is the base of the natural logarithm, both ϵ and α are given constants with $0<\epsilon<1$ and $0<\alpha<1$, and n and m are the number of nodes and links in the social network, respectively.
- We finally evaluate the effectiveness of the proposed algorithm by extensive experimental simulations, using different real datasets. Experimental results show that the proposed algorithm outperforms existing algorithms.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces preliminaries and formulates the influence maximization problem under the structural diversity model. Section 4 devises a probabilistic approximation algorithm for the problem, and analyzes its time complexity, approximation ratio, and success probability. Section 5 empirically evaluates the proposed algorithm, and Section 6 concludes the paper and points out potential future work.

2. Related work

In this section, we review related work on the influence maximization problem under different influence propagation models. Domingos et al. [13,26] were the first to study the influence maximization problem as an algorithmic problem, and proposed an influence propagation model which specifies a joint distribution over all nodes' behavior globally. Kempe

et al. [21] proposed two popular stochastic influence propagation models that can explicitly model the step-by-step dynamics of influence propagation progress: the Independent Cascade Model and the Linear Threshold Model. For a given social network, under the independent cascade model, each user v is activated by one of his/her friends u with an activation probability $p(u, v)$. Given k chosen users in the social network, the influence propagation process proceeds as follows. Initially, each of the k users accepts a product at time 0. Once user u accepts the product but his/her friend v has not yet accepted the product at time t, then user u has a chance to recommend the product to user v, and user v will accept the product with a certain probability $p(u, v)$ at time $t + 1$. Thus, the more friends of user v recommend the product to the user, the more likely he/she will accept the product. On the other hand, under the linear threshold model, each user v is activated by one of his/her friends u with an influence weight $w(u, v) \in [0, 1]$ so that $\sum_{u:\, friend\ of\ v} w(u, v) \leq 1$. Furthermore, each user v chooses a threshold θ_v uniformly from the interval [0, 1]. If a user v has not accepted the product at time t, and the sum of the influence weights of his/her friends who have accepted the product so far is greater than the threshold θ_v, then user v accepts the product at time $t + 1$. Again, we can see that the more friends of user v have accepted the product, the more likely that the sum of the influence weights of these friends is greater than the threshold θ_v of user v, and therefore the more likely user v accepts the product. Chen et al. [9] extended the independent cascade model by taking the propagation deadline constraint into consideration, that is, a user will not recommend the product to his/her friends when $t \geq T$, where T is a given deadline constraint. Wang et al. [33] investigated the problem of selecting seeds to maximize their influence within a given budget, assuming that the costs of persuading different users to accept a product are different. Borg et al. [3] recently proposed a near-optimal time algorithm for the influence maximization problem under the independent cascade model, and Tang et al. [27] further incorporated heuristics to improve the time complexity without compromising the performance guarantee. On the other hand, there is a consensus among social scientists that a person who plays a bridge role between different communities can acquire more potential resources from these communities and has more control over the information that is being transmitted, and the person who develops relations with people from multiple communities are referred to as a *structural hole spanner* [25]. Rezvani et al. recently studied the problem of identifying top-k structural hole spanners in large-scale social networks [25]. Despite that much progress has been made to address the influence maximization problem under the independent cascade model or the linear threshold model, the proposed algorithms in [3,27,33] are inapplicable to the problem under the structural diversity model, since under which the decision of a user depends on the social structures of different groups of neighbors, rather than the number of neighbors.

The mentioned influence propagation models only consider the influence propagation progress of a single product, there are other influence propagation models for multiple competitive products [5,18] (e.g., two brands of smartphones from two companies), where the influence propagation of each product will interfere with the influence propagations of the other products and each user will only accept at most one product at the end of their influence propagation processes. Budak et al. [5] proposed a multi-campaign independent cascade model by extending the independent cascade model, while He et al. [18] proposed a competitive linear threshold model by extending the linear threshold model.

3. Preliminaries

In this section, we first introduce the network model, we then propose an influence propagation structural diversity model and define the influence maximization problem precisely, under the structural diversity model. We finally introduce submodular functions, the independent cascade model, and the live-edge graph model, which will be used later.

3.1. Network model

We model a social network as an undirected graph $G = (V, E)$, where V is the set of nodes representing users and E is the set of edges representing the friendships between users in the network. For each node $u \in V$, denote by $N^G(u)$ the set of its neighbors in G, i.e., $N^G(u) = \{v \mid v \in V, (u, v) \in E\}$. In the rest of the paper, we abbreviate $N^G(u)$ to $N(u)$ if no ambiguity arises.

Following the study by Ugander et al. [28], the probability of accepting a recommendation by a user monotonically grows with the number of connected components in the induced graph by his/her neighbors, rather than the number of his/her neighbors, where each connected component represents a distinct social context of the user, e.g., families, workmates, classmates, etc. For each node $u \in V$, let $G[N(u)]$ be the induced subgraph of G by the neighbor set $N(u)$ of u, i.e., $G[N(u)] = (N(u), E_u)$, where $E_u = \{(v, w) \mid v, w \in N(u), (v, w) \in E\}$. Assume that graph $G[N(u)]$ consists of n_u connected components. Denote by $N_j(u)$ the node set of the jth connected component in $G[N(u)]$, $1 \leq j \leq n_u$. The neighbor set $N(u)$ of node u thus is partitioned into n_u disjoint subsets $N_1(u), N_2(u), \ldots, N_{n_u}(u)$.

3.2. The structural diversity model and problem definition

Consider the spread of an idea in a social network G, each node of G is in either one of two modes: *active* or *inactive*, which corresponds whether the node adopts the idea or not. Similar to one of the two influence propagation models in the seminal paper [21]: the Independent Cascade Model, we here consider *a Structural Diversity Model with Independent Cascade* (abbreviated to the Structural Diversity Model) as follows. Denote by A_t the set of active nodes in G at time t, $t = 0, 1, 2, \ldots$. Given an initial set of active nodes $A_0 \subseteq V$ at time 0, the influence propagates in discrete time, according to the following

randomized rule. At time $t+1$, let $A_{t+1} = A_t$ initially, which means that all active nodes before time $t+1$ still are active at time $t+1$. For each inactive node $u \in V \backslash A_t$ and each connected component $N_j(u)$ in the induced graph $G[N(u)]$, node u is activated with probability $p(N_j(u), u)$, if there is a node in $N_j(u)$ that was activated at time t (i.e., $N_j(u) \cap (A_t \setminus A_{t-1}) \neq \emptyset$) and no other nodes in $N_j(u)$ became active before time t (i.e., $N_j(u) \cap A_{t-1} = \emptyset$), where p is a given influence propagation probability function p: $2^V \times V \mapsto [0, 1]$. If node u is successfully activated, u is added into A_{t+1}. Note that only the very first activated nodes (multiple nodes can be activated at the same time) in $N_j(u)$ have a single chance to activate node u and the other later activated nodes in $N_j(u)$ do not have such a chance, since they are in the same connected component $N_j(u)$ with these first activated nodes, thus have a similar social context. This influence propagation process continues until no inactive nodes become active.

Having the structural diversity model, we now define an influence maximization problem as follows. Given a social network $G = (V, E)$, an integer k (as the budget), and an influence propagation probability function p: $2^V \times V \mapsto [0, 1]$, *the influence maximization problem in G under the structural diversity model* is to identify k seed nodes such that the expected number of activated nodes eventually by the k nodes is maximized. The problem is NP-hard as it can be reduced from the well-known NP-hard set cover problem in polynomial time [21].

3.3. Submodular functions

In this subsection, we here introduce the notion of submodular functions, since the influence maximization problem can be cast as a submodular function maximization problem. Let U be a finite set and z be a real-valued function: $z : 2^U \mapsto \mathcal{R}^{\geq 0}$, function z is a non-decreasing submodular function if and only if it has the following three properties.

(1) $z(\emptyset) = 0$;
(2) Non-decreasement: $z(S) \leq z(T)$ for any two sets S, $T \subseteq U$ with $S \subseteq T$;
(3) Diminishing return property (submodularity): $z(S \cup \{v\}) - z(S) \geq z(T \cup \{v\}) - z(T)$ for any two sets S and T with $S \subseteq T \subset U$ and $v \in U \setminus T$.

3.4. The independent cascade model and the live-edge graph model

We first describe *the independent cascade model* [21]. Given a *directed* graph $G' = (V', E')$, an activation probability function: p': $E' \mapsto [0, 1]$, and a seed set $S \subset V'$, the influence of set S propagates as follows. Let $A_0 = S$. At time $t+1$, let $A_{t+1} = A_t$ initially. Then, for each inactive node $v \in V' \setminus A_t$ and each node $u \in N_{in}(v) \cap (A_t \setminus A_{t-1})$, node u will activate node v with a probability $p'(u, v)$, where $N_{in}(v)$ is the incoming neighbor set of node v and $(A_t \setminus A_{t-1})$ is the set of nodes activated at time t. If v is activated by node u, it is added into A_{t+1}. This process continues until $A_{t+1} = A_t$ for some time t.

We then describe *the live-edge graph model* [8] (pp.11–13). Given a directed graph $G' = (V', E')$, an activation probability function: p': $E' \mapsto [0, 1]$, and an initial seed set $S \subset V'$, a subgraph $H = (V', E'')$ of G' is randomly constructed. Specifically, for each edge $e' \in E'$, edge e' is in E'' with a probability $p'(e')$. Note that none of the edges in H is assigned an activation probability. Denote by $d_H(u, v)$ the distance of the shortest path in H from node u to node v, assuming that the length of each edge in H is one. If there is no any paths in H from node u to node v, set $d_H(u, v) = \infty$. Let $d_H(S, v)$ be shortest distance from nodes in S to node v in graph H, i.e., $d_H(S, v) = \min_{u \in S}\{d_H(u, v)\}$. Obviously, $d_H(S, v) = 0$ if a node $v \in S$. Denote by $R_H(S, t)$ the set of reachable nodes from nodes in S in graph H within t hops, i.e., $R_H(S, t) = \{v \in V' \mid d_H(S, v) \leq t\}$. Given an initial seed set $S \subset V'$, the active node set under the live-edge graph model at time t is defined as $R_H(S, t)$, where $t = 0, 1, \ldots$.

Chen et al. [8] (pp.13–14) showed the two influence propagation models introduced are equivalent through the following lemma.

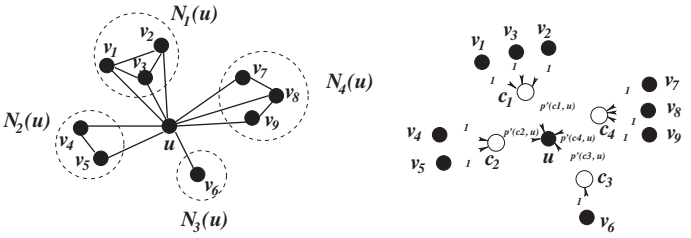
Lemma 1 ([8]). *Given a directed social network $G' = (V', E'; p')$, the independent cascade model is equivalent to the live-edge graph model. Specifically, given a seed set S, for each time t with $t = 0, 1, 2, \ldots$, if $A_0 = R_H(S, 0), A_1 = R_H(S, 1), \ldots, A_t = R_H(S, t)$, then for each inactive node $u \in V' \backslash A_t$, the probability of node u being activated at time $t+1$ under the independent cascade model is equal to the probability of node u reached at the $(t+1)$th hop under the live-edge graph model, i.e., $P\{u \in A_{t+1}\} = P\{u \in R_H(S, t+1)\}, \forall u \in V' \setminus A_t$.*

4. Approximation algorithm

In this section, we first propose a probabilistic approximation algorithm for the influence maximization problem under the proposed structural diversity model. We then analyze the time complexity, approximation ratio, and success probability of the proposed algorithm.

4.1. Algorithm

Our basic strategy is to first reduce the influence maximization problem in an undirected social network G under the structural diversity model to a slightly different influence maximization problem in another directed graph G' under the independent cascade model. We then show that an approximate solution to the latter in turn returns an approximate solution to the former.



(a) A node u and its neighbor set $N(u) = \{v_1, v_2, \ldots, v_9\}$ in G

(b) The converted subgraph in G' from node u and the nodes in $N(u)$

Fig. 1. The conversion of a node u and its neighbors in $N(u)$ in graph G.

Given a social network $G = (V, E)$ that is an undirected graph, the influence propagation probability function p: $2^V \times V \mapsto [0, 1]$, and the structural diversity model, we construct an auxiliary directed graph $G' = (V \cup C, E'; p')$ as follows. For each node $u \in V$ of G, recall that $N(u)$ is the neighbor set of node u in G and the induced graph $G[N(u)]$ by $N(u)$ consists of n_u connected components $N_1(u), N_2(u), \ldots, N_{n_u}(u)$, where connected component $N_j(u)$ can activate node u with a probability $p(N_j(u), u)$, $1 \le j \le n_u$, illustrated in Fig. 1 (a). We add n_u *component nodes* $c_1, c_2, \ldots, c_{n_u}$ to the node set C in graph G', where node c_j represents the connected component $N_j(u)$. There is a directed edge in E' from node c_j to node u with activation probability $p'(c_j, u) = p(N_j(u), u)$, $1 \le j \le n_u$. For a node $v_i \in N(u)$ contained in $N_j(u)$, we add a directed edge from node v_i to component node c_j to E' with activation probability one. Fig. 1 (b) illustrates such a construction. Thus, the node set $V \cup C$ of G' consists of two types of nodes: *individual nodes* in V representing individual users and *component nodes* in C representing the connected components induced by the neighbor sets of the nodes in G. Given a seed set $S \subset V$, denote by A_t the set of active nodes in G under the structural diversity model at time t, and denote by A'_t the set of active nodes in G' under the independent cascade model at time t, where $t = 0, 1, 2, \ldots$.

There is an interesting phenomenon about the influence propagation progress in G' under the independent cascade model. That is, individual nodes in G' can be activated only by component nodes, not by individual nodes, since there are no edges in G' between individual nodes, and vice versa. Given an initial seed set $S \subset V$, recall that A'_t is the set of active nodes in G' under the independent cascade model at time t, where $t = 0, 1, 2, \ldots$. Initially, $A'_0 = S$. We observe that only individual nodes are activated at time $t = 0, 2, 4, \ldots$, while only component nodes are activated at time $t = 1, 3, 5, \ldots$, i.e., $(A'_t \setminus A'_{t-1}) \subseteq V$ if t is an even number, and $(A'_t \setminus A'_{t-1}) \subseteq C$ otherwise.

Having graphs $G = (V, E;\ p : 2^V \times V \mapsto [0, 1])$ and $G' = (V \cup C, E';\ p' : E' \mapsto [0, 1])$, we show that the structural diversity model in G is equivalent to the independent cascade model in G' by the following theorem.

Theorem 1. *The structural diversity model in G is equivalent to the independent cascade model in G'. Specifically, given a seed set $S \subset V$, for each time $t = 0, 1, 2, \ldots$, if $A_j = A'_{2j} \cap V$, where $A'_{2j} \cap V$ is the set of active individual nodes in G' at time $2j$ and $0 \le j \le t$, then for each inactive node $u \in V \setminus A_t$, the probability of node u in G being activated under the structural diversity model at time $t+1$ is equal to the probability of node u in G' being activated under the independent cascade model at time $2(t+1)$, i.e., $P\{u \in A_{t+1}\} = P\{u \in A'_{2(t+1)}\}, \forall u \in V \setminus A_t$.*

Proof. Consider any inactive node $u \in V \setminus A_t$, we calculate $P\{u \in A_{t+1}\}$ as follows. Recall that the induced graph $G[N(u)]$ by the neighbor set $N^G(u)$ of node u consists of n_u connected components $N_1^G(u), N_2^G(u), \ldots, N_{n_u}^G(u)$. Let $\mathcal{N}^G$ be the set of these n_u connected components. We partition the n_u connected components into three groups $\mathcal{N}_O^G, \mathcal{N}_F^G, \mathcal{N}_I^G$ (letters 'O', 'F', and 'I' represent words 'Outdated', 'Fresh', and 'Inactive', respectively), where a connected component $N_j^G(u)$ is in group $\mathcal{N}_O^G$ if some nodes in $N_j^G(u)$ were activated before time t (i.e., $\mathcal{N}_O^G = \{N_j^G(u) \mid N_j^G(u) \cap A_{t-1} \neq \emptyset\}$), $N_j^G(u)$ is in group $\mathcal{N}_F^G$ if none nodes in $N_j^G(u)$ were activated before time t and some nodes in $N_j^G(u)$ are activated at time t (i.e., $\mathcal{N}_F^G = \{N_j^G(u) \mid N_j^G(u) \cap A_{t-1} = \emptyset, N_j^G(u) \cap A_t \neq \emptyset\}$), and $N_j^G(u)$ is in group $\mathcal{N}_I^G$ if every node in $N_j^G(u)$ is inactive at time t (i.e., $\mathcal{N}_I^G = \{N_j^G(u) \mid N_j^G(u) \cap A_t = \emptyset\}$). Following the structural diversity model, for each connected component $N_j^G(u)$, only the first activated nodes in it have a chance to activate node u with a probability $p(N_j^G(u), u)$. Thus, only the connected components in $\mathcal{N}_F^G$ can activate node u at time $t+1$. Therefore, the probability of node u being activated at time $t+1$ is $P\{u \in A_{t+1}\} = 1 - \prod_{N_j^G(u) \in \mathcal{N}_F^G} (1 - p(N_j^G(u), u))$.

The rest is to calculate the probability of node u in G' being activated under the independent cascade model at time $2(t+1)$. Let $N_{in}^{G'}(u)$ be the incoming neighbor set of node u in G'. Following the construction of G', each node in $N_{in}^{G'}(u)$ in G' is a component node and each component node c_j corresponds to a connected component $N_j^G(u)$ in G, see Fig. 1 (b). It is obvious that $|N_{in}^{G'}(u)| = n_u$. For each component node $c_j \in N_{in}^{G'}(u)$, let $N_{in}^{G'}(c_j)$ be the incoming neighbor set of node c_j in G'. Then, each node in $N_{in}^{G'}(c_j)$ is an individual node. Similarly, we further partition the set $N_{in}^{G'}(u)$ into three subsets $N_O^{G'}, N_F^{G'}$, and $N_I^{G'}$, where a component node $c_j \in N_{in}^{G'}(u)$ is in set $N_O^{G'}$ if some nodes in $N_{in}^{G'}(c_j)$ were activated before time $2t$ (i.e.,

$N_O^{G'} = \{c_j \mid N_{in}^{G'}(c_j) \cap A'_{2t-1} \neq \emptyset\}$), c_j is in set $N_F^{G'}$ if none nodes in $N_{in}^{G'}(c_j)$ were activated before time $2t$ and some nodes in $N_{in}^{G'}(c_j)$ are activated at time $2t$ (i.e., $N_F^{G'} = \{c_j \mid N_{in}^{G'}(c_j) \cap A'_{2t-1} = \emptyset, N_{in}^{G'}(c_j) \cap A'_{2t} \neq \emptyset\}$), and c_j is in set $N_I^{G'}$ if every individual node in $N_{in}^{G'}(c_j)$ is inactive at time $2t$ (i.e., $N_I^{G'} = \{c_j \mid N_{in}^{G'}(c_j) \cap A'_{2t} = \emptyset\}$). Following the construction of G' and the assumption that $A_j = A'_{2j} \cap V$ for every j with $0 \leq j \leq t$, we can see that each connected component $N_j^G(u)$ in $\mathcal{N}_O^G$ (or $\mathcal{N}_F^G$, or $\mathcal{N}_I^G$) in G corresponds to a component node c_j in $N_O^{G'}$ (or $N_F^{G'}$, or $N_I^{G'}$) in G', and vice versa. In the following we consider the influence propagation process in G' under the independent cascade model at time $2t+1$ and time $2(t+1)$, respectively.

We first consider the process at time $2t+1$ as follows. We observe that at time $2t$, each component node in $N_O^{G'}$ is active while each component node in $N_I^{G'}$ or $N_F^{G'}$ is inactive, by following the definitions of $N_O^{G'}$, $N_I^{G'}$ and $N_F^{G'}$. Note that at time $2t+1$, only component nodes are activated by some individual nodes in G'. Following the independent cascade model, each component node c_j in $N_O^{G'}$ is still active at time $2t+1$ since c_j is active at time $2t$, each component node c_j in $N_F^{G'}$ will become active at time $2t+1$ since $N_{in}^{G'}(c_j) \cap A'_{2t} \neq \emptyset$ and each individual node in $N_{in}^{G'}(c_j) \cap A'_{2t}$ will activate node c_j with probability one at time $2t+1$, and each component node c_j in $N_I^{G'}$ is still inactive at time $2t+1$ since $N_{in}^{G'}(c_j) \cap A'_{2t} = \emptyset$. In other words, $N_O^{G'} \subseteq A'_{2t}$, $N_F^{G'} \subseteq A'_{2t+1} \setminus A'_{2t}$, and $N_I^{G'} \cap A'_{2t+1} = \emptyset$.

We then study the process at time $2(t+1)$ and calculate $P\{u \in A'_{2(t+1)}\}$ as follows. Following the independent cascade model, only nodes in $A'_{2t+1} \setminus A'_{2t}$ can activate node u at time $2(t+1)$. Since each node in $N_{in}^{G'}(u)$ is a component node in G', the probability of node u being activated under the independent cascade model at time $2(t+1)$ is $P\{u \in A'_{2(t+1)}\} = 1 - \prod_{c_j \in N_{in}^{G'}(u), c_j \in A'_{2t+1} \setminus A'_{2t}} (1 - p'(c_j, u)) = 1 - \prod_{c_j \in N_F^{G'}} (1 - p'(c_j, u)) = 1 - \prod_{N_j^G(u) \in \mathcal{N}_F^G} (1 - p(N_j^G(u), u))$, since $p'(c_j, u) = p(N_j(u), u)$. The theorem then follows. □

Since the structural diversity model in G is equivalent to the independent cascade model in G', for each given seed set $S \subset V$, S has the same influence in both G and G', i.e., the expected number of nodes activated eventually by S in graph G under the structural diversity model is equal to the expected number of individual nodes activated eventually by S in G' under the independent cascade model.

In graph $G' = (V \cup C, E'; p')$, denote by $f(S)$ the expected number of individual nodes that are activated eventually by a seed set $S \subset V$ in G' under the independent cascade model. We now consider the problem of finding a seed set $S \subset V$ in G' with $|S| \leq k$ such that $f(S)$ is maximized. It must be mentioned that the considered problem in G' is different from the traditional influence maximization problem in a graph G'' under the independent cascade model in [21], since the problem in [21] is to identify a k-seed set S'' in G'', such that the expected number of nodes in G'' activated eventually by the seed set S'' is maximized, while in our case the graph G' contains two types of different nodes (i.e., individual nodes and component nodes) and our problem is to find a k-seed set S that contains only individual nodes so that the expected number of individual nodes in G' activated eventually by the seed set S is maximized and the expected number of activated component nodes will not be counted in terms of the influence.

Assume that function $f(S)$ is a non-decreasing submodular function. To identify a k-seed set S in G' such that $f(S)$ is maximized, Nemhauser et al. [24] proposed a greedy algorithm with an approximation ratio of $(1 - 1/e)$ as follows, where e is the base of the natural logarithm. The algorithm starts with an empty set ($S = \emptyset$), and iteratively adds an individual node u_{max} in G' into S that leads to the maximum marginal gain at each iteration, i.e., $f(S \cup \{u_{max}\}) - f(S) = \max_{u \in V \setminus S}\{f(S \cup \{u\}) - f(S)\}$. The procedure continues until a seed set with k nodes is found.

Notice that the value computation of function $f(\cdot)$ is very difficult, as the computation of the influence $f(S)$ of S is #P-hard for a set S. This implies that the computing of $f(S)$ is at least hard as NP-hardness [29]. By adopting the strategy in [21], we use Monte Carlo simulations of the influence propagation process to estimate the influence $f(S)$. Specifically, given a seed set S, we simulate the influence propagation process in G' for L times. Each time we count the number of active individual nodes after the propagation process, and then take the average of these counts over the L times. With a sufficiently large L, the obtained average count will be an approximation to $f(S)$ with high probability. For a given threshold $\gamma > 0$, we say that an estimate $\hat{f}(S)$ is a *γ-error estimate* of $f(S)$, if $|\hat{f}(S) - f(S)| \leq \gamma \cdot f(S)$.

Given an error ratio ϵ with $0 < \epsilon < 1$ and a probability $1 - \alpha$ with $0 < \alpha < 1$, we devise an approximation algorithm for finding a k-seed set S, and the algorithm achieves an approximation ratio of $(1 - 1/e - \epsilon)$ with probability $1 - \alpha$. we refer to this algorithm as Algorithm 1, which proceeds as follows. Initially, $S = \emptyset$. It then performs k iterations to find a k-seed set. Within iteration i, it invokes a subroutine Algorithm 2 to compute the approximate influence $\hat{f}(S \cup \{u\})$ for each individual node $u \in V \setminus S$, by performing Monte Carlo simulations $L_i = \lceil \frac{\epsilon^{-2}\alpha^{-1}k(2k+\epsilon)^2 n}{2i} \rceil$ times. It then selects a node u_{max} with the maximum approximate influence $\hat{f}(S \cup \{u\})$ among nodes in $V \setminus S$, and add node u_{max} to S.

4.2. Identifying the next seed

Let S be the selected seed set so far with $|S| < k$, we show how to efficiently calculate the approximate influence $\hat{f}(S \cup \{u\})$ for each individual node $u \in V \setminus S$, using Monte Carlo simulations L times, and then add a node $u_{max} \in V \setminus S$ with the maximum approximate influence $\hat{f}(S \cup \{u_{max}\})$ to set S. In the following, we start with a simple strategy for the computation of $\hat{f}(S \cup \{u\})$ for each node $u \in V \setminus S$, and we then present a novel algorithm for the calculation.

Algorithm 1 Greedy algorithm under the SD model.

Input: $G = (V, E)$, influence propagation probability $p : 2^V \times V \mapsto [0, 1]$, an integer $k > 0$,an error ratio ϵ with $0 < \epsilon < 1$, and a probability $1 - \alpha$ with $0 < \alpha < 1$.
Output: A k-seed set $S \subset V$ such that S is a $(1 - 1/e - \epsilon)$ approximate solution with probability $1 - \alpha$.
1: Obtain a directed graph $G' = (V \cup C, E'; p')$ from G;
2: $S \leftarrow \emptyset$;
3: **for** $i \leftarrow 1$ to k **do**
4: $L_i \leftarrow \lceil \epsilon^{-2}\alpha^{-1}k(2k+\epsilon)^2 n/(2i)\rceil$; /* the times of Monte Carlo simulations */
5: Select the node u_{max} with the maximum approximate influence $\max_{u\in V\setminus S}\{\hat{f}(S\cup\{u\})\}$ by performing L_i Monte Carlo simulations, by calling `Algorithm 2`;
6: $S \leftarrow S \cup \{u_{max}\}$;
7: **end for**
8: **return** S.

Algorithm 2 Identify the next seed.

Input: $G' = (V \cup C, E'; p')$, a seed set $S \subset V$, and times of Monte Carlo Simulations L
Output: Choose a node $u \in V \setminus S$ with the maximum approximate influence $\hat{f}(S \cup \{u\})$
1: Let $r_u = 0$ for each $u \in V \setminus S$; /*expected number of reachable individual nodes*/
2: **for** $j \leftarrow 1$ to L **do**
3: Construct graph $H_j = (V \cup C, E''_j)$ from G';
4: Find the set $R_{H_j}(S)$ of reachable nodes from the nodes in S in H_j, let $R^I_{H_j}(S) = R_{H_j}(S) \cap V$;
5: Obtain a graph H'_j by removing the nodes in $R_{H_j}(S)$ and their incident edges from H_j;
6: Find all SCCs in graph H'_j;
7: Construct a DAG $H''_j = (V^*_j, E^*_j; \rho)$ from H'_j by collapsing each SCC into a node in H''_j;
8: For each node $u^* \in V^*_j$ with weight $\rho(u^*) > 0$, calculate $r^*_{u^*} = \sum_{v^*\in R_{H''_j}(u^*)} \rho(v^*)$;
9: For each node $u \in R^I_{H_j}(S) \setminus S$, let $r_u \leftarrow r_u + |R^I_{H_j}(S)|$;
10: For each node $u \in V \setminus R^I_{H_j}(S)$, assume u^* is the collapsed node in H''_j of u, let $r_u \leftarrow r_u + |R^I_{H_j}(S)| + r^*_{u^*}$;
11: **end for**
12: **return** node u_{max} such that $r_{u_{max}} = \max_{u\in V\setminus S}\{r_u\}$.

We present the simple strategy first. For each Monte Carlo simulation among the L times, we simulate the influence propagation process in G' under the live-edge graph model, rather than the independent cascade model, due to their equivalence by Lemma 1. Given a graph $G' = (V \cup C, E'; p')$, and a seed set $S \cup \{u\} \subset V$, Kempe et al. [21] showed that we can simulate the influence propagation process in G' under the live-edge graph model as follows. We first construct a directed graph $H = (V \cup C, E'')$. For each edge $e' \in E'$ with an activation probability $p'(e')$, we add edge e' in E'' with probability $p'(e')$. Having the constructed graph H, we obtain a sample value of $f(S\cup\{u\})$ by calculating the number of reachable individual nodes in H from nodes in $S\cup\{u\}$. Denote by $R_H(S\cup\{u\})$ and $R^I_H(S\cup\{u\})$ the set of reachable nodes (including individual nodes and component nodes) and reachable individual nodes in H from the nodes in $S\cup\{u\}$, respectively. i.e., $R_H(S\cup\{u\}) = \{v \mid v \in V \cup C,$ there is a directed path in H from a node $w \in S\cup\{u\}$ to $v\}$ and $R^I_H(S\cup\{u\}) = \{v \mid v \in V,$ there is a directed path in H from a node $w \in S\cup\{u\}$ to $v\}$. The calculation of $R_H(S\cup\{u\})$ can be done by performing a Depth-First Search (DFS) in H. Then, $R^I_H(S\cup\{u\}) = R_H(S\cup\{u\}) \cap V$. To estimate the value of $f(S\cup\{u\})$, we independently construct L auxiliary graphs $H_1, H_2, \ldots, H_L$ like constructing the graph H, and compute $R^I_{H_j}(S\cup\{u\})$ in each graph. We approximate $f(S\cup\{u\})$ with $\hat{f}(S\cup\{u\}) = \frac{\sum_{j=1}^{L}|R^I_{H_j}(S\cup\{u\})|}{L}$. The total running time of calculating $\hat{f}(S\cup\{u\})$s for nodes in $V\setminus S$ is $|V\setminus S|\cdot L\cdot(n_{G'} + m_{G'}) = O(nL(n_{G'} + m_{G'}))$, where $|V\setminus S| \le |V| = n$.

We then show how to improve the computational efficiency of the simple strategy, by adopting a batch estimation technique. We consider the L graphs $H_1, H_2, \ldots, H_L$ one by one. For each graph $H_j = (V \cup C, E''_j)$, we compute the number of reachable nodes $|R^I_{H_j}(S\cup\{u_1\})|, |R^I_{H_j}(S\cup\{u_2\})|, \ldots, |R^I_{H_j}(S\cup\{u_{n-s}\})|$ from the sets $S\cup\{u_1\}, S\cup\{u_2\}, \ldots, S\cup\{u_{n-s}\}$, respectively, in a batch way, where $u_l \in V\setminus S$, $1 \le l \le n-s$, and $n - s = |V\setminus S|$. For graph $H_j = (V \cup C, E''_j)$, we first perform a DFS traversal on H_j to find the set of reachable nodes $R_{H_j}(S)$ from seed set S. Then, $R^I_{H_j}(S) = R_{H_j}(S) \cap V$. We remove the nodes in $R_{H_j}(S)$ and their incident edges from graph H_j and obtain a residual graph H'_j. We note that

$$|R^I_{H_j}(S\cup\{u\})| = \begin{cases} |R^I_{H_j}(S)|, & \forall u \in R^I_{H_j}(S) \\ |R^I_{H_j}(S)| + |R^I_{H'_j}(u)|, & \forall u \in V \setminus R^I_{H_j}(S) \end{cases} \quad (1)$$

We thus only need to calculate $|R^l_{H'_j}(u)|$ for each node $u \in V \setminus R^l_{H_j}(S)$ in graph H'_j as follows.

We first find all strongly connected components (SCCs) in H'_j. We then collapse each SCC into a supernode with a node weight $\rho(u^*)$ being the number of individual nodes in it. As a result, we obtain a directed acyclic graph (DAG) $H''_j = (V^*_j, E^*_j; \rho)$. Notice that the size of the collapsed graph H''_j usually is much smaller than that of the original graph H'_j. Chen et al. [10] showed one important property relating to graphs H'_j and H''_j that, if each node $u \in H'_j$ has been collapsed to a node $u^* \in H''_j$, then

$$|R^l_{H'_j}(u)| = \sum_{v^* \in R_{H''_j}(u^*)} \rho(v^*). \tag{2}$$

Note that we only need to calculate $\sum_{v^* \in R_{H''_j}(u^*)} \rho(v^*)$ for each node u^* in H''_j with a positive weight, since each individual node u in H'_j must be collapsed into a node u^* in H''_j with weight $\rho(u^*) > 0$. We finally calculate the value $\sum_{v^* \in R_{H''_j}(u^*)} \rho(v^*)$ for each node u^* in H''_j by performing a DFS starting from node u^*. We refer to this algorithm as Algorithm 2 whose description is in the following.

4.3. Algorithm Analysis

In the following we analyze the performance of Algorithm 1, including its time complexity, approximation ratio, and success probability.

4.3.1. Analysis of the time complexity

Given a social network $G = (V, E)$ with $n = |V|$ and $m = |E|$, recall that $G' = (V \cup C, E')$ is the directed graph derived from G. Let $n_{G'} = |V \cup C|$ and $m_{G'} = |E'|$. We start by the following lemma.

Lemma 2. *The number of nodes and the number of edges in graph G' are no more than $n + 2m$ and $4m$, respectively. Moreover, graph G' can be constructed in time $O(d_{max} \cdot m)$, where d_{max} is the maximum node degree among the nodes in G.*

Proof. Let d_u be the node degree of a node u in G. It is obvious that $\sum_{u \in V} d_u = 2m$. For each node $u \in V$, recall that the induced subgraph $G[N(u)]$ by the neighbor set $N(u)$ of node u contains n_u connected components. Clearly, $n_u \le d_u$. Since we add n_u component nodes into C, the number of nodes in $V \cup C$ is $|V \cup C| = |V| + |C| = n + \sum_{u \in V} n_u \le n + \sum_{u \in V} d_u = n + 2m$. The number of edges in E' is $|E'| = \sum_{u \in V}(n_u + d_u) \le 2\sum_{u \in V} d_u = 4m$.

The time complexity of the construction of graph G' is as follows. Let $d_{max} = \max_{u \in V}\{d_u\}$. For each node $u \in V$ of G, the induced subgraph $G[N(u)]$ by the neighbor set $N(u)$ of node u can be constructed in time $O(d_u \cdot d_{max})$ while the n_u connected components can be found in time $O(d_u \cdot d_{max})$, by performing DFS traversals on $G[N(u)]$. Therefore, G' can be constructed in time $\sum_{u \in V} O(d_u \cdot d_{max}) = O(d_{max} \cdot m)$. Note that $d_{max} \ll n$ in most real-world social networks. □

We then show the time complexity of Algorithm 1 by the following lemma.

Lemma 3. *Given a social network $G = (V, E)$, an activation probability function p: $2^V \times V \mapsto [0, 1]$, an integer k, and two constants ϵ and α with $0 < \epsilon < 1$ and $0 < \alpha < 1$, the time complexity of Algorithm 1 for the influence maximization problem in G is $O(\epsilon^{-2}\alpha^{-1}n^2(m+n)k^3 \log k)$, which improves by a factor of αn of the time complexity $O(\epsilon^{-2}n^3(m+n)k^3 \log(nk/\alpha))$ of the state-of-the-art approximation algorithm[8] (pp.43), where $n = |V|$ and $m = |E|$.*

Proof. We first show that the time complexity of Algorithm 2 is $O(Ln(n_{G'} + m_{G'}))$. Since Algorithm 2 performs L times of Monte Carlo simulations, we only analyze the running time of one Monte Carlo simulation. Within iteration j, the total running time of constructing graph $H_j = (V \cup C, E''_j)$, finding the set $R_{H_j}(S)$, constructing graph H'_j from H_j, finding all SCCs in H'_j, and constructing graphing $H''_j = (V^*_j, E^*_j; \rho)$ from H'_j, is $O(n_{G'} + m_{G'})$. Then, for each node u^*_l in V^*_j with weight $\rho(u^*_l) > 0$, we perform a DFS traversal, starting from node u^*_l. Since $\rho(u^*_l) > 0$ means that as least one individual node in H'_j has been collapsed to node u^*_l in H''_j, the number of nodes in H''_j with weight $\rho(u^*_l) > 0$ is no more than the number of individual nodes in V. We thus perform no more than $|V| = n$ DFS traversals on graph H''_j. In summary, the time complexity of Algorithm 2 is $L(O(n_{G'} + m_{G'}) + n(O(n_{G'} + m_{G'}))) = O(Ln(n_{G'} + m_{G'}))$.

The time complexity of Algorithm 1 thus is $O(d_{max} \cdot m) + \sum_{i=1}^{k} O(L_i n(n_{G'} + m_{G'})) = n(n_{G'} + m_{G'})O(\sum_{i=1}^{k} \epsilon^{-2}\alpha^{-1}k(2k + \epsilon)^2 n/(2i)) = O(\epsilon^{-2}\alpha^{-1}n^2(m+n)k^3 \log k)$, since $O(n_{G'} + m_{G'}) = O(n + m)$ and $\sum_{i=1}^{k} 1/i = O(\log k)$ by [11]. □

4.3.2. Analysis of the approximation ratio

We now show that the approximation ratio of Algorithm 1 is $(1 - 1/e - \epsilon)$ with probability $1 - \alpha$, where $0 < \alpha < 1$. Given graph $G' = (V \cup C, E'; p')$, we first show that the influence function $f(S)$ is a non-decreasing submodular function, we then show the approximation ratio.

Lemma 4. *Given a directed graph $G' = (V \cup C, E'; p')$, define the value of function $f(S)$ as the expected number of activated individual nodes eventually in G' at the end of influence propagation process by a seed set $S \subseteq V$ under the independent cascade model. Then, function $f(S)$ is a non-decreasing submodular function.*

Proof. It is obvious that $f(\emptyset) = 0$ and $f(S) \leq f(T)$ for any two seed sets $S, T \subseteq V$ with $S \subseteq T$. We only need to show that $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ for any two seed sets $S, T \subseteq V$ with $S \subseteq T$ and each node $u \in V \backslash T$.

We first calculate the influence $f(S)$ of a seed set $S \subseteq V$ as follows. Let $\mathcal{G}' = \{G'_1, G'_2, \ldots, G'_q\}$ be the set of different subgraphs of G' with the same node set $V \cup C$, where $1 \leq j \leq q$, $q = 2^{m_{G'}}$ and $m_{G'} = |E'|$. Following the live-edge graph model, we construct an auxiliary directed graph $H = (V \cup C, E'')$ from $G' = (V \cup C, E'; p')$ as follows. For each edge $e' \in E'$ with an activation probability $p'(e')$, we add an edge e' in E'' with probability $p'(e')$. Note that H must be a graph in $\mathcal{G}'$. Define $P\{H = G'_j\}$ as the probability of $H = G'_j$, where $1 \leq j \leq q$. Denote by $R_{G'_j}(S)$ and $R^I_{G'_j}(S)$ the set of reachable nodes and reachable individual nodes from the nodes in S in graph G'_j, respectively. Let $f'(S)$ be the expected number of active nodes (including component nodes and individual nodes) eventually in G'_j by seed set S under the independent cascade model. Due to the equivalence of the independent cascade model and the live-edge graph model, Kempe et al. [21] showed that $f'(S) = \sum_{G'_j \in \mathcal{G}'} P\{H = G'_j\} \cdot |R_{G'_j}(S)|$. Similarly, we have $f(S) = \sum_{G'_j \in \mathcal{G}'} P\{H = G'_j\} \cdot |R^I_{G'_j}(S)|$.

We then show that $|R^I_{G'_j}(S)|$ is a submodular function in each subgraph G'_j. For any two seed sets $S, T \subset V$ with $S \subseteq T$ and each node $u \in V \backslash T$, consider any node $v \in R^I_{G'_j}(T \cup \{u\}) \setminus R^I_{G'_j}(T)$, i.e., node v is reachable from node u but not reachable from the nodes in T. Since S is a subset of T, then node v is reachable from node u but not reachable from the nodes in S. We thus have $v \in R^I_{G'_j}(S \cup \{u\}) \setminus R^I_{G'_j}(S)$. Therefore, $|R^I_{G'_j}(T \cup \{u\}) \setminus R^I_{G'_j}(T)| \leq |R^I_{G'_j}(S \cup \{u\}) \setminus R^I_{G'_j}(S)|$, i.e., $|R^I_{G'_j}(T \cup \{u\})| - |R^I_{G'_j}(T)| \leq |R^I_{G'_j}(S \cup \{u\})| - |R^I_{G'_j}(S)|$, and the function $|R^I_{G'_j}(S)|$ thus is a submodular function. Since $f(S)$ is a non-negative linear combination of submodular functions, $f(S)$ is a submodular function, too. □

The analysis of the approximation ratio of Algorithm 1 proceeds as follows. Let $S = \{u_1, u_2, \ldots, u_k\}$ be the solution delivered by Algorithm 1 and individual node u_i is chosen at iteration i by the algorithm. Let $S_0 = \emptyset$ and $S_i = \{u_1, u_2, \ldots, u_i\}$, $1 \leq i \leq k$. Following Algorithm 1, we have $u_i = \arg\max_{v \in V \backslash S_{i-1}} \{\hat{f}(S_{i-1} \cup \{v\})\}$. Let $\bar{u}_i = \arg\max_{v \in V \backslash S_{i-1}} \{f(S_{i-1} \cup \{v\})\}$, $1 \leq i \leq k$. Note that nodes u_i and $\bar{u}_i$ may be the same node. Given an error ratio ϵ with $0 < \epsilon < 1$, let $\delta = \epsilon/(2k + \epsilon)$. Assume that both values $\hat{f}(S_{i-1} \cup \{u_i\})$ and $\hat{f}(S_{i-1} \cup \{\bar{u}_i\})$ are δ-error estimates of the values $f(S_{i-1} \cup \{u_i\})$ and $f(S_{i-1} \cup \{\bar{u}_i\})$, respectively, i.e., $|\hat{f}(S_{i-1} \cup \{u_i\}) - f(S_{i-1} \cup \{u_i\})| \leq \delta \cdot f(S_{i-1} \cup \{u_i\})$ and $|\hat{f}(S_{i-1} \cup \{\bar{u}_i\}) - f(S_{i-1} \cup \{\bar{u}_i\})| \leq \delta \cdot f(S_{i-1} \cup \{\bar{u}_i\})$ for every i with $1 \leq i \leq k$. Under this assumption, Chen et al. [8] (pp.41–43) showed that the approximation ratio of Algorithm 1 is $(1 - 1/e - \epsilon)$, by the following lemma.

Lemma 5 *[8]. Given the graph $G' = (V \cup C, E'; p')$ derived from G, an integer k, and an error ratio ϵ with $0 < \epsilon < 1$, let $\delta = \epsilon/(2k + \epsilon)$. If both approximate influences $\hat{f}(S_{i-1} \cup \{u_i\})$ and $\hat{f}(S_{i-1} \cup \{\bar{u}_i\})$ of $f(S_{i-1} \cup \{u_i\})$ and $f(S_{i-1} \cup \{\bar{u}_i\})$ are δ-error estimates for each i with $1 \leq i \leq k$, then Algorithm 1 for the influence maximization problem in G' delivers an approximate solution with an approximation ratio of $(1 - 1/e - \epsilon)$ under the independent cascade model.*

4.3.3. Analysis of the success probability

Following Lemma 5, the claim that Algorithm 1 is a $(1 - 1/e - \epsilon)$ approximation algorithm is based on the assumption of that the values $\hat{f}(S_{i-1} \cup \{u_i\})$ and $\hat{f}(S_{i-1} \cup \{\bar{u}_i\})$ are δ-error estimates of the values $f(S_{i-1} \cup \{u_i\})$ and $f(S_{i-1} \cup \{\bar{u}_i\})$ for every i with $1 \leq i \leq k$. In the following we derive a probabilistic algorithm with the same approximation ratio by removing this assumption.

Let Z_i (or $\bar{Z}_i$) be a random variable with $Z_i = 1$ (or $\bar{Z}_i = 1$) if and only if the value $\hat{f}(S_{i-1} \cup \{u_i\})$ (or $\hat{f}(S_{i-1} \cup \{\bar{u}_i\})$) is a $\delta(= \epsilon/(2k + \epsilon))$-error estimate of the value $f(S_{i-1} \cup \{u_i\})$ (or $f(S_{i-1} \cup \{\bar{u}_i\})$); otherwise $Z_i = 0$ (or $\bar{Z}_i = 0$) for every i with $1 \leq i \leq k$. We show that the probability $P\{Z_1 = 1, \bar{Z}_1 = 1, \ldots, Z_k = 1, \bar{Z}_k = 1\}$ is no less than $1 - \alpha$. Then, Algorithm 1 achieves an approximation ratio of $(1 - 1/e - \epsilon)$ with probability $1 - \alpha$ by Lemma 5.

We show that $P\{Z_i = 1\} \geq 1 - \beta$ and $P\{\bar{Z}_i = 1\} \geq 1 - \beta$ for every i with $1 \leq i \leq k$, where $\beta = \alpha/2k$. Following Algorithm 1, within iteration i, it computes the approximate influence $\hat{f}(S_{i-1} \cup \{u\})$ for every node $u \in V \setminus S_{i-1}$, including the nodes u_i and $\bar{u}_i$, by performing $L_i = \lceil \epsilon^{-2} \alpha^{-1} k(2k + \epsilon)^2 n/(2i) \rceil$ Monte Carlo simulations. Recall that $\delta = \epsilon/(2k + \epsilon)$ and $\beta = \alpha/2k$. We thus have $L_i \geq \epsilon^{-2} \alpha^{-1} k(2k + \epsilon)^2 n/(2i) = n/(4i\delta^2\beta)$. Note that $|S_{i-1} \cup \{u_i\}| = |S_{i-1} \cup \{\bar{u}_i\}| = i$. We show that $P\{Z_i = 1\} \geq 1 - \beta$ and $P\{\bar{Z}_i = 1\} \geq 1 - \beta$ by the following lemma.

Lemma 6. *Given graph $G' = (V \cup C, E'; p')$, a seed set $S \subset V$ with $S \neq \emptyset$ and $|S| = s$, an error ratio δ with $0 < \delta < 1$, and a probability $1 - \beta$ with $0 < \beta < 1$, we conduct L times of independent Monte Carlo simulations in G' to simulate the influence propagation process under the independent cascade model with the same seed set S. Let Y_j be the number of active individual nodes in G' in the jth Monte Carlo simulation, where $1 \leq j \leq L$. Let $\bar{Y}_L = (\sum_{j=1}^{L} Y_j)/L$. Assume that $\mu = E[Y_1] = E[Y_2] = \cdots = E[Y_L]$ is the expected value of the L random variables. If $L \geq n/(4s\delta^2\beta)$, then the probability of the absolute difference between $\bar{Y}_L$ and μ being no more than $\delta\mu$ is no less than $1 - \beta$, i.e.,*

$$P\{|\bar{Y}_L - \mu| \leq \delta\mu\} \geq 1 - \beta. \tag{3}$$

Proof. In the following, we first calculate the expectation and variance of random variable $\bar{Y}_L$, we then show that $P\{|\bar{Y}_L - \mu| \leq \delta\mu\} \geq 1 - \beta$. We observe that

$$E[\bar{Y}_L] = E\left[\frac{\sum_{j=1}^{L} Y_j}{L}\right] = \frac{\sum_{j=1}^{L} E[Y_j]}{L} = \frac{L \cdot \mu}{L} = \mu. \tag{4}$$

Assume that the variance of random variable Y_1 is σ^2, i.e., $\sigma^2 = Var[Y_1]$. It is obvious that $\sigma^2 = Var[Y_1] = Var[Y_2] = \cdots = Var[Y_L]$. The variance of random variable $\bar{Y}_L$ is

$$Var[\bar{Y}_L] = Var\left[\frac{\sum_{j=1}^{L} Y_j}{L}\right] = \frac{\sum_{j=1}^{L} Var[Y_j]}{L^2} = \frac{L \cdot \sigma^2}{L^2} = \frac{\sigma^2}{L}, \tag{5}$$

since random variables $Y_1, Y_2, \ldots, Y_L$ are independently and identically distributed.

To show that $P\{|\bar{Y}_L - \mu| \leq \delta\mu\} \geq 1 - \beta$, it is sufficient to show that $P\{|\bar{Y}_L - \mu| \geq \delta\mu\} \leq \beta$. We show that the latter inequality holds by the Chebyshev inequality. According to the Chebyshev inequality, we have

$$\begin{aligned} & P\{|\bar{Y}_L - \mu| \geq \delta\mu\} = P\{|\bar{Y}_L - E[\bar{Y}_L]| \geq \delta\mu\} \\ & \leq Var[\bar{Y}_L]/(\delta\mu)^2 = (1/L\delta^2) \cdot (\sigma^2/\mu^2) \\ & \leq (1/L\delta^2) \cdot (n/4s) \end{aligned} \tag{6}$$

$$\begin{aligned} & \leq (1/\delta^2) \cdot (4s\delta^2\beta/n) \cdot (n/4s) \quad (as\ L \geq n/4s\delta^2\beta) \\ & = \beta, \end{aligned} \tag{7}$$

where the inequality (6) holds by Lemma 7 (see Appendix), which shows that $\sigma^2/\mu^2 \leq n/4s$. The lemma then follows. □

By Lemmas 3, 5, and 6, we have the following theorem.

Theorem 2. *Given a social network $G = (V, E)$, a positive integer k, an activation probability function p: $2^V \times V \mapsto [0, 1]$, an error ratio ϵ with $0 < \epsilon < 1$, and a success probability $1 - \alpha$ with $0 < \alpha < 1$, there is a $(1 - 1/e - \epsilon)$-approximation algorithm with a probability no less than $1 - \alpha$ for the influence maximization problem in G under the structural diversity model. The algorithm takes $O(\epsilon^{-2}\alpha^{-1}n^2(m + n)k^3 \log k)$ time, where both ϵ and α are constants with $0 < \epsilon < 1$ and $0 < \alpha < 1$, $n = |V|$, and $m = |E|$.*

Proof. We have shown the time complexity and approximation ratio of Algorithm 1 by Lemmas 3 and 5, respectively. We now show its success probability as follows.

By Lemma 6, we know that $P\{Z_i = 1\} \geq 1 - \beta$ and $P\{\bar{Z}_i = 1\} \geq 1 - \beta$ for every i with $1 \leq i \leq k$. Then, $P\{Z_i = 0\} \leq \beta$ and $P\{\bar{Z}_i = 0\} \leq \beta$. The success probability of Algorithm 1 is

$$\begin{aligned} & P\{Z_1 = 1, \bar{Z}_1 = 1, \ldots, Z_k = 1, \bar{Z}_k = 1\} \\ & = 1 - P\{Z_1 = 0\ or\ \bar{Z}_1 = 0\ or\ \cdots\ or\ Z_k = 0\ or \bar{Z}_k = 0\} \\ & \geq 1 - \sum_{i=1}^{k} (P\{Z_i = 0\} + P\{\bar{Z}_i = 0\}) \end{aligned} \tag{8}$$

$$\geq 1 - \sum_{i=1}^{k} (\beta + \beta) = 1 - \alpha, \text{ as } \beta = \alpha/2k. \tag{9}$$

The theorem then follows. □

5. Algorithm evaluation

In this section, we evaluate the performance of the proposed algorithm using different real datasets. We also compare the algorithm with other heuristics for finding highly influential users in social networks.

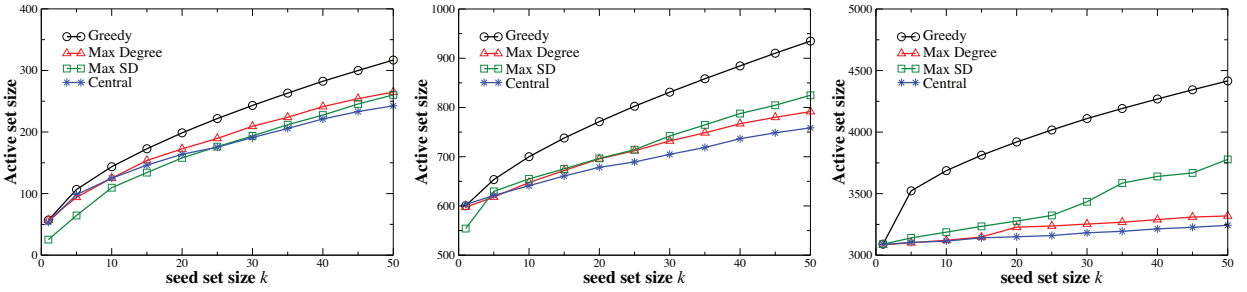
5.1. Experimental environment setting

We use the datasets of four real-world social networks, which are listed in Table 1, where the first network is an academic collaboration network NetHEPT obtained from the "High Energy Physics-Theory" section of the e-print arXiv, in which each node represents an author of a paper and for a paper with two or more authors, an edge is added for each pair of authors. The second network come from the full paper list of the "Physics" section, denoted by NetPHY. The datasets of the first two networks are available on the web [1]. The third network is an online social network–Facebook [2], where each node

[1] http://research.microsoft.com/en-us/people/weic/graphdata.zip.
[2] http://socialnetworks.mpi-sws.org/data-wosn2009.html.

Table 1
Statistics of four real-world networks.

Dataset	NetHEPT	NetPHY	Facebook	DBLP
# of nodes	15,233	37,154	63,731	654,628
# of edges	58,853	231,584	817,090	1,990,159
Avg. degree	3.86	6.23	12.82	3.04



(a) uniform setting ($p_c = 0.1$) (b) uniform setting ($p_c = 0.2$) (c) weighted setting

Fig. 2. Expected numbers of active users by different algorithms in the collaboration network NetHEPT.

represents a user and each edge represents the friendship between two users. The final network is a much larger collaboration network from the DBLP Computer Science Bibliography [3]. The datasets are widely adopted in existing works, such as [10,12,21], and [29].

Following existing works [21,29], we consider two types of influence propagation probability settings: the uniform setting and the weighted setting. Recall that the induced graph $G[N(u)]$ of the neighbor set $N(u)$ of node u consists of n_u connected components $N_1(u), N_2(u), \ldots, N_{n_u}(u)$. In the uniform setting, the influence propagation probability $p(N_j(u), u)$ of each connected component $N_j(u)$ is assigned a uniform probability p_c, i.e., $p(N_j(u), u) = p_c$, where p_c is a constant with $0 < p_c < 1$ and $1 \leq j \leq n_u$. Intuitively, only a very few users will be eventually activated if p_c is very small (e.g., 0.01) while a large proportion of users will be influenced if p_c is quite large (e.g, 0.8) [21]. For the former (i.e., p_c is very small), it is not worthwhile to consider the influence maximization problem since only a very few users will be influenced by the initial seeds. For the latter (i.e., p_c is very large), it is not realistic in a real social network that every social context of a user has high influence probability to the user. We thus consider that p_c is set with 0.1 and 0.2 in this paper so that the value of p_c is neither too small nor too large. In the weighted setting, $p(N_j(u), u)$ is assigned $1/n_u$ so that the expected number of connected components which would succeed in activating node u is one [21].

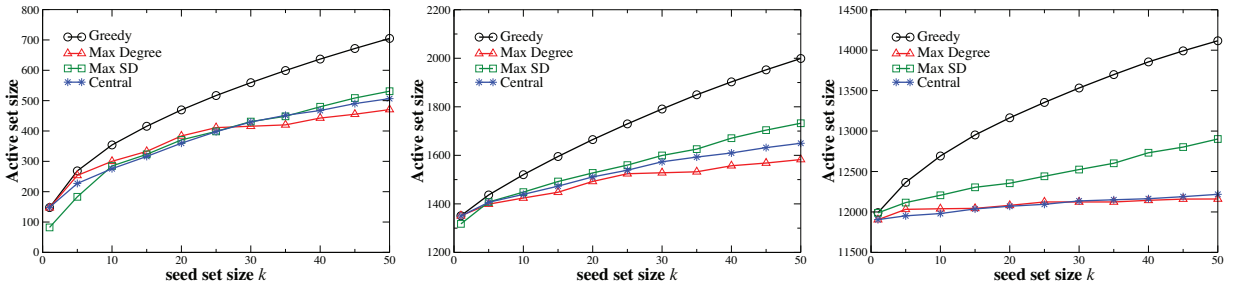
We compare the proposed algorithm `Greedy` (i.e., Algorithm 1) with other three heuristics: (1) Algorithm max degree. The max degree heuristic chooses k seed nodes with the top-k degrees [21]; (2) Algorithm max structural diversities (max SD). Similar to the max degree heuristic, the max structural diversities heuristic selects k seed nodes with the top-k structural diversities, where the structural diversity of a node u in G is the number of connected components in the induced graph $G[N(u)]$; and (3) Algorithm distance centrality (central). The distance centrality heuristic finds k seed nodes with the smallest average shortest-path distances to other nodes [10,21]. The distance between two unreachable nodes is set as n, where n is the number of nodes in G. Note that there are some other heuristics, such as the ones in [10,29], which will not be compared with since they are only applicable to the independent cascade model or the linear threshold model.

Following the similar setting as the works in [10,29], the number k of seed nodes is chosen between 1 and 50. The default error ratio ϵ setting is 0.03, and the error probability α is 5%. The proposed approximation algorithm has an approximation ratio of $1 - 1/e - \epsilon \approx 0.6$ with probability $1 - \alpha = 95\%$ following its theoretical analysis. All the experiments are performed on a desktop with Intel(R) Core(TM) i7-4790 CPU (3.6 GHz), 4 GB RAM, and the operating system of Windows 8.1 enterprise. All the mentioned algorithms are implemented in C++ in the integrated development environment (IDE) of Microsoft Visual Studio 2013.

5.2. Performance on influence

In the following we evaluate the performance of the proposed algorithm in networks NetHEPT, NetPHY, Facebook, and DBLP, respectively, in terms of influence (i.e., expected numbers of active users) of the found seeds. Fig. 2 shows the influence of the seeds found by different algorithms under the uniform setting with $p_c = 0.1$ and $p_c = 0.2$, and the weighted setting in the collaboration network NetHEPT, by varying the seed set size k from 1 to 50. Fig. 2 (a) clearly shows that the seed set delivered by the proposed algorithm `Greedy` has larger influence than that of the seed sets by other mentioned

[3] http://snap.stanford.edu/data/index.html.



(a) uniform setting ($p_c = 0.1$) (b) uniform setting ($p_c = 0.2$) (c) weighted setting

Fig. 3. Expected numbers of active users by different algorithms in the collaboration network NetPHY.

algorithms, and the performance gap between them grows bigger and bigger with the growth of the value of k. Specifically, the influence of the seed set found by algorithm `Greedy` is about 20%, 20%, and 30% larger than that of the seed sets identified by algorithms `Max_Degree`, `Max_SD`, and `Central`, respectively, when $k = 50$. The rationale behind the outperformance of the proposed algorithm `Greedy` over the existing algorithms is as follows. Algorithm `Greedy` finds the top-k most influential users by not only considering how users are accurately influenced (i.e., under the structural diversity model) but also taking into account the influence among the users chosen (i.e., there is only a little influence among the selected users). In contrast, algorithms `Max_Degree` and `Central` do not consider the influence propagation process of users and only select the top-k influential users by their degrees and average distances to other users in the network, respectively, while algorithm `Max_SD` may choose k seeds that have large influence to each other. Fig. 2 (b) and (c) also demonstrate that the proposed algorithm `Greedy` can find much more influential seed sets than that by the other three mentioned algorithms. For example, under the weighted setting with $k = 50$, the expected number of activated individuals by the 50 seeds found by algorithm `Greedy` can reach up to 4400, which takes about 30% ($\approx 4400/15,233$) of the total number of individuals in network NetHEPT, while the influences of the seed sets found by algorithms `Max_Degree`, `Max_SD`, and `Central` are only about 3300, 3800, and 3250, respectively.

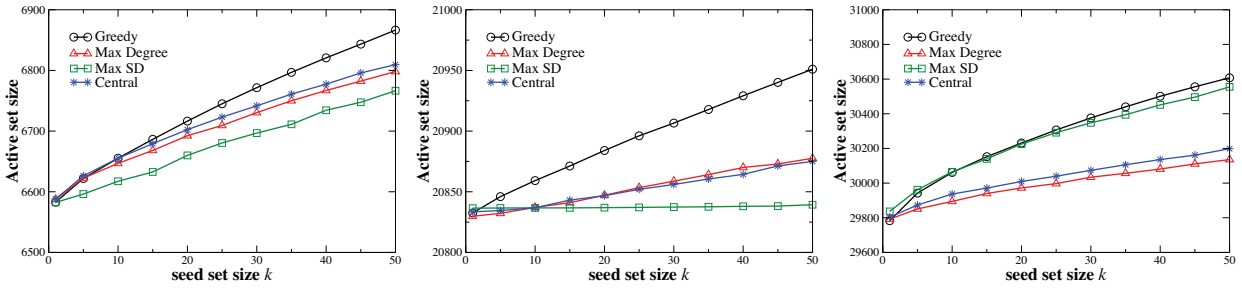
Fig. 2 shows an interesting phenomenon that although the influence of the seed set discovered by algorithm `Max_SD` is no better even worse than that by algorithms `Max_Degree` and `Central` when the seed set size k is small, the influence of the seed set found by it is larger than that by the two mentioned algorithms when k is large. For example, algorithm `Max_SD` finds a seed set with an approximate influence of 3800 under the weighted setting when $k = 50$, which is about 14% and 17% larger than that by `Max_Degree` and `Central`, respectively. The rationale behind the phenomenon is that algorithm `Max_SD` finds the top-k nodes by the number of connected components in the induced graph $G[N(u)]$ of the neighbor set $N(u)$ of each node u, i.e., it considers that the neighbors in the same connected component have the similar social context, while algorithms `Max_Degree` and `Central` fail to distinguish the the neighbors of a node from the same connected component, and only find the top-k nodes by node degrees (`Max_Degree`) or average shortest-path distances to other nodes (`Central`).

We then investigate the performance of different algorithms in network NetPHY whose size is larger than that of network NetHEPT. Fig. 3 plots the influence of the seed sets found by the four mentioned algorithms. Similar to the performance of the algorithms in network NetHEPT, the seed set identified by algorithm `Greedy` has much larger influence than that found by the other three algorithms `Max_Degree`, `Max_SD`, and `Central` in network NetPHY. Moreover, the advantage of algorithm `Greedy` in network NetPHY is even bigger than that in network NetHEPT. For example, Fig. 3 (a) demonstrates that the influence of the seed set found by algorithm `Greedy` in network NetPHY is 50%, 32%, and 39% larger than that of the seed sets found by the other three algorithms under the uniform setting with $p_c = 0.1$ and $k = 50$, while its advantage over the three algorithms is about 20%, 20%, and 30% in network NetHEPT (see Fig. 2 (a)).

We also study the performance of the algorithms in social network Facebook. Fig. 4 shows that the seed set found by algorithm `Greedy` has larger influence than that by the other three algorithms. Furthermore, the influence of the seed set identified by algorithm `Max_SD` is smaller than algorithms `Max_Degree` and `Central` under the uniform setting (Fig. 4 (a) and (b)) whereas the influence of the seed set found by it is as large as that of the seed set discovered by algorithm `Greedy` under the weighted setting (Fig. 4 (c)).

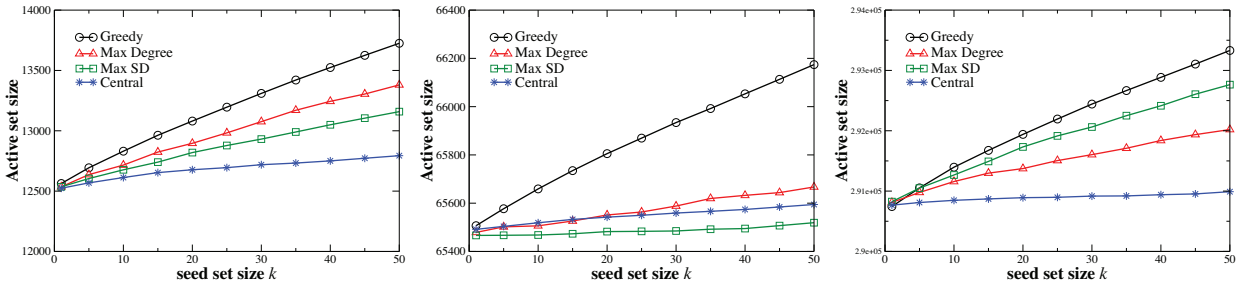
We finally investigate the performance of the algorithms in network DBLP. Again, Fig. 5 shows that the performance algorithm `Greedy` is the best among the four algorithms. Furthermore, the influence of the seed set identified by algorithm `Max_Degree` is higher than algorithms `Max_SD` and `Central` under the uniform setting (Fig. 5 (a) and (b)) whereas the influence of the seed set found by it is however worse than that by algorithm `Max_SD` under the weighted setting (Fig. 5 (c)).

In summary, it can been seen that the seed set found by algorithm `Greedy` has the highest influence compared with the seeds found by the other existing algorithms for all mentioned social networks with all different influence propagation probability settings, while the performances of existing algorithms `Max_Degree`, `Central`, and `Max_SD` depend on not only the structure of the social networks but also the influence probability settings.



(a) uniform setting ($p_c = 0.1$) (b) uniform setting ($p_c = 0.2$) (c) weighted setting

Fig. 4. Expected numbers of active users by different algorithms in the social network Facebook.



(a) uniform setting ($p_c = 0.1$) (b) uniform setting ($p_c = 0.2$) (c) weighted setting

Fig. 5. Expected numbers of active users by different algorithms in the network DBLP.

5.3. Performance on running time

The rest is to investigate the running times of the four mentioned algorithms for different networks including NetHEPT, NetPHY, Facebook, and DBLP, respectively. Fig. 6(a) plots the running times of algorithms `Greedy`, `Max_Degree`, `Max_SD`, and `Central` in network NetHEPT which consists of 15,233 nodes and 58,853 edges (i.e., $n = 15,233$ and $m = 58,853$), under the uniform setting with $p_c = 0.1$, by varying the seed set size k from 1 to 50. It can be seen from Fig. 6(a) that the running times of algorithms `Max_Degree` and `Max_SD` are much shorter than those of algorithms `Greedy` and `Central`. Specifically, the running times of algorithms `Max_Degree`, `Max_SD`, `Central`, and `Greedy` are 0.03 s, 0.62 s, 76 s, and 1.22 h, respectively, when $k = 50$. Fig. 6(a) also shows that the running time curves of algorithms `Max_Degree`, `Max_SD`, and `Central` are almost flat with the increase of the seed set size k. This is due to the facts that the calculations of node degrees in algorithm `Max_Degree`, the node structural diversities in algorithm `Max_SD`, and the average distance to other nodes in algorithm `Central`, dominate their running times, respectively, while the running time of algorithm `Greedy` grows with the increase of k by Lemma 3. Fig. 6 (b), (c), and (d) plot the running times of different algorithms for three larger networks NetPHY, Facebook, and DBLP, from which it can be seen that the performance behaviors of these algorithms are similar with each other. Also, it can be seen that each of the algorithms takes a much longer time to find the top-k influential users in a larger social network. For example, the running times of algorithm `Greedy` for networks NetPHY, Facebook, and DBLP, are 4.26 h, 18.45 h, and 64.7 h, respectively, when $k = 50$.

Fig. 7 plots the running time of algorithm `Greedy` for network DBLP with $k = 50$, under the uniform setting with $p_c = 0.1$ and $p = 0.2$, as well as the weighted setting. We here omit the running times of algorithms `Max_Degree`, `Max_SD`, and `Central` since their running times are independent with the influence probability setting. Fig. 7 indicates that the running time of algorithm `Greedy` under different influence probability settings does not vary too much. For example, its running time grows from about 65 h under the uniform setting with $p = 0.1$ to approximately 70 h under the weighted setting.

Notice that although the experimental results demonstrate that the running time of the proposed algorithm `Greedy` is longer than those of algorithms `Max_Degree`, `Max_SD`, and `Central`, it is acceptable in practice due to the following three reasons. The first is that the size of activated users by the initial seed users usually is much more important than the running time of the algorithm. For example, a company intends to market one of its new products, e.g., iPhone 6s, by targeting a few “influential” users in a social network through giving the users the product samples. If the solution delivered by the proposed algorithm can bring a 10% improvement of activated users, the improvement will bring enormous profits to the company if the size of the social network is quite large. The company may not really care about the running time of such an algorithm from a couple of hours to several days if it is acceptable (e.g., 64.7 h for the DBLP network), as such an algorithm runs only once when there is any new product to be marketed. The second is that the proposed algorithm

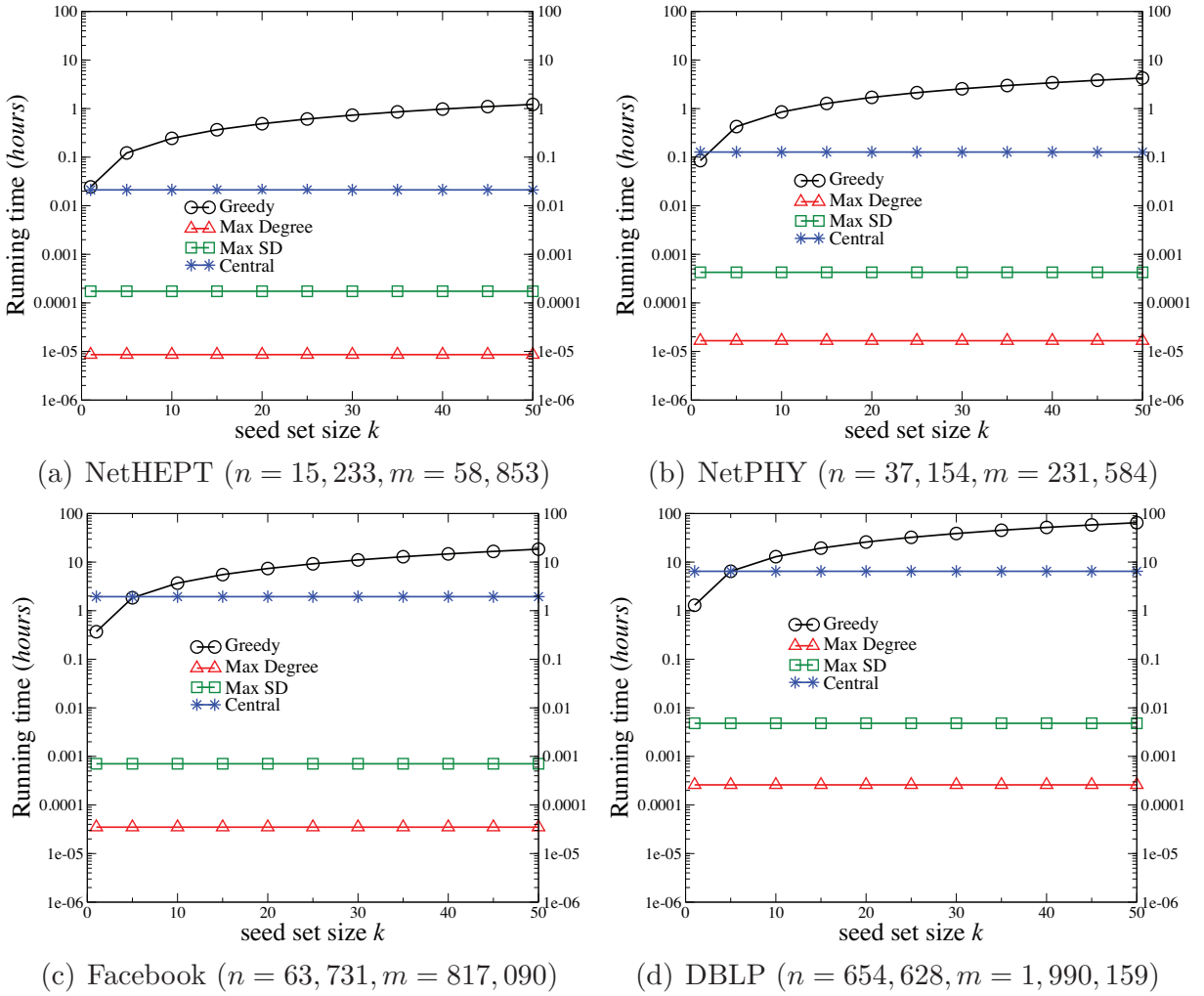


Fig. 6. Running times of different algorithms in networks NetHEPT, NetPHY, Facebook, and DBLP, respectively, under the uniform setting with $p_c = 0.1$.

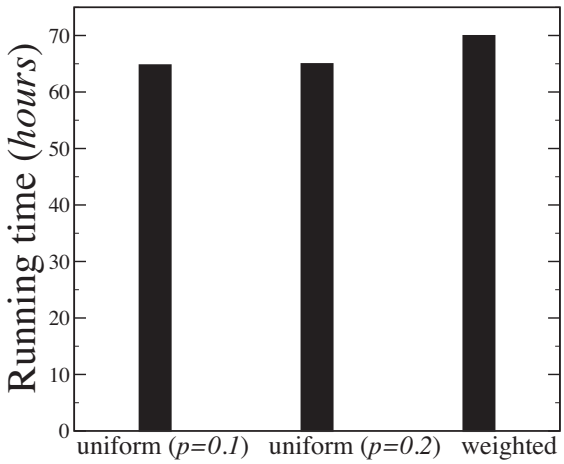


Fig. 7. The running time of algorithm `Greedy` for network DBLP with $k = 50$, under both the uniform setting with $p_c = 0.1$ and $p = 0.2$, and the weighted setting.

Greedy can always deliver a feasible solution with a guaranteed performance in comparison with the optimal solution of the problem, no matter which social networks and what influence propagation probability settings are considered, while the solutions delivered by the other mentioned algorithms do not have such performance guarantees, which means that their solutions may be far from the optimal ones for some special networks. Although the empirical results of the other mentioned algorithms for a given special network are not too bad, it does not imply that they can deliver solutions with a performance guarantee for some other networks or influence probability settings. The final is that the running time of algorithm Greedy, e.g., 64.7 h for network DBLP, can be considerably reduced (e.g., within a few hours), by exploring the parallelism in its implementation in a multi-core cluster of servers.

We would like to mention that the focus of this paper is to identify influential users under the structural diversity model that is totally different from the traditional independent cascade model or linear threshold model, under which the decision of a user depends on the social structures of different groups of its neighbors, rather than the number of its neighbors, the computational complexity of finding the structures is much higher, compared with the calculation of the number of neighbors.

6. Conclusions and future work

Identifying a few influential users to maximize their influence to other users in a social network has attracted plenty of attentions in the past decade. Most existing studies of the influence maximization problem adopt either the independent cascade model or the linear threshold model. One common assumption of these two models is that a user is more likely to be influenced if more his/her friends have already been influenced. This assumption however recently was challenged to be over simplified and inaccurate, as the social decision of a user depends more subtly on the network structure, rather than the number of his/her influenced neighbors. Instead, it is shown that a user is very likely to be influenced by his/her influenced friends with higher "structural diversities". In this paper we first formulated a novel influence maximization problem under this new structural diversity model. We then proposed a $(1-1/e-\epsilon)$-approximation algorithm with a probability $1-\alpha$, where e is the base of the natural logarithm, ϵ and α are given constants with $0<\epsilon<1$ and $0<\alpha<1$. We finally evaluated the effectiveness of the proposed algorithm by extensive experimental simulations, using different real datasets. Experimental results show that the proposed algorithm outperforms existing ones, and the seed set found by it has much larger influence than that delivered by existing algorithms. In our future work, we will focus on reducing the time complexity of the proposed algorithm by exploring other optimization techniques such as execution parallelism, advanced data structures, and so on.

Acknowledgment

We appreciate the anonymous referees and the Editor-in-Chief Professor Witold Pedrycz for their expertise comments and constructive suggestions, which have helped us improve the quality and presentation of the paper greatly. It is also acknowledged that the work by Wenzheng Xu was partially supported by 2016 Basic Research Talent Foundation of Sichuan University in China (Grant no. 2082204194050), and the work by Jeffrey Xu Yu was partially supported by Research Grants Council of the Hong Kong SAR, China (Grant no. 14209314).

Appendix

Lemma 7. *Given graph $G'=(V\cup C,E';p')$, a seed set $S\subset V$ with $S\neq\emptyset$ and $|S|=s$, an error ratio δ with $0<\delta<1$, and a probability $1-\beta$ with $0<\beta<1$, let random variable Y be the number of active individual nodes in G' at the end of the influence propagation process under the independent cascade model with seed set S. Assume that μ is the expected value of Y, i.e., $\mu=E[Y]$, and σ^2 is the variance of Y, i.e., $\sigma^2=Var[Y]$. Then, $\sigma^2/\mu^2\leq n/4s$, where $n=|V|$.*

Proof. For each node $v_i\in V\setminus S$, let X_i be a random variable with $X_i=1$ if and only if node v_i is activated eventually by the seed set S; otherwise, $X_i=0$, where $1\leq i\leq n-s=|V\setminus S|$. Assume that $P\{X_i=1\}=p_i$, $0\leq p_i\leq 1$. Then, $P\{X_i=0\}=1-p_i$. It is obvious that $E[X_i]=p_i$, $1\leq i\leq n-s$. Note that random variables $X_1,X_2,\ldots,X_{n-s}$ are not independent with each other. We represent random variable Y by random variables $X_1,X_2,\ldots,X_{n-s}$ as

$$Y=s+\sum_{i=1}^{n-s}X_i,$$

where s is the number of nodes in seed set S and these s nodes must be activated at the end of the influence propagation process. Notice that

$$\mu=E[Y]=s+\sum_{i=1}^{n-s}E[X_i]=s+\sum_{i=1}^{n-s}p_i,$$

and

$$\begin{aligned}\sigma^2 = Var[Y] = Var\left[\sum_{i=1}^{n-s} X_i\right] &= \sum_{i=1}^{n-s}\sum_{j=1}^{n-s} Cov(X_i, X_j)\\ &= \sum_{i=1}^{n-s}\sum_{j=1}^{n-s}(E[X_i \cdot X_j] - E[X_i]\cdot E[X_j])\\ &\le \sum_{i=1}^{n-s}\sum_{j=1}^{n-s}(p_i - p_i p_j) = \sum_{i=1}^{n-s}\sum_{j=1}^{n-s} p_i(1-p_j),\end{aligned} \tag{A.1}$$

where $Cov(X_i, X_j)$ is the covariance of random variables X_i and X_j, i.e. $Cov(X_i, X_j) = E[X_i \cdot X_j] - E[X_i]\cdot E[X_j]$, and inequality (A.1) holds as $P\{X_i = 1, X_j = 1\} \le P\{X_i = 1\} = p_i$. It is obvious that $\sigma^2/\mu^2 \le (\sum_{i=1}^{n-s}\sum_{j=1}^{n-s} p_i(1-p_j))/(s + \sum_{i=1}^{n-s} p_i)^2$. Let function $g(x_1, x_2, \ldots, x_{n-s}) = (\sum_{i=1}^{n-s}\sum_{j=1}^{n-s} x_i(1-x_j))/(s + \sum_{i=1}^{n-s} x_i)^2$, where $0 \le x_i \le 1$ and $1 \le i \le n-s$. We show that the maximum value of function $g(x_1, x_2, \ldots, x_{n-s})$ is no more than $n/4s$ as follows.

Assume that function $g(x_1, x_2, \ldots, x_{n-s})$ achieves its maximum value at a point $(x_1^*, x_2^*, \ldots, x_{n-s}^*)$. From the symmetric form of variables $x_1, \ldots, x_{n-s}$ in function $g(x_1, \ldots, x_{n-s})$, we know that $x_1^* = x_2^* = \cdots = x_{n-s}^*$. Let function $h(x) = g(x, x, \ldots, x) = \frac{(n-s)^2 x(1-x)}{(s+(n-s)x)^2}$, where $0 \le x \le 1$. It is obvious that functions $g(x_1, x_2, \ldots, x_{n-s})$ and $h(x)$ have the same maximum value. To calculate the maximum value of function $h(x)$, we compute the derivative of $h(x)$ with respect to x. $\frac{\partial h(x)}{\partial x} = \frac{(n-s)^2}{(s+(n-s)x)^3}\cdot(s-(n+s)x)$. Let $\frac{\partial h(x)}{\partial x} = 0$. Then, $x = s/(n+s)$. Therefore, function $h(x)$ reaches its maximum value when $x = s/(n+s)$, and its maximum value is: $\max_{0\le x\le 1}\{h(x)\} = h(s/(n+s)) = (n-s)^2/4ns \le n/4s$.

By the above discussion, we then have

$$\begin{aligned}\frac{\sigma^2}{\mu^2} &\le g(p_1, p_2, \ldots, p_{n-s}) \le \max_{0\le x_i\le 1}\{g(x_1, x_2, \ldots, x_{n-s})\}\\ &= \max_{0\le x\le 1}\{h(x)\} \le n/4s.\end{aligned} \tag{A.2}$$

□

References

[1] N. Agarwal, H. Liu, L. Tang, P.S. Yu, Identifying the influential bloggers in a community, in: Proceedings of the 2008 ACM International Conference on Web Search and Data mining (WSDM), 2008.
[2] E. Bakshy, J.M. Hofman, W.A. Mason, D.J. Watts, Everyone's an influencer: quantifying influence on twitter, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM), 2011.
[3] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2014.
[4] A. Borodin, M. Braverman, B. Lucier, J. Oren, Strategy proof mechanisms for competitive influence in networks, in: Proceedings of the 22nd International Conference on World Wide Web (WWW), 2013.
[5] C. Budak, D. Agrawal, A.E. Abbadi, Limiting the spread of misinformation in social networks, in: Proceedings of the 20th International Conference on World Wide Web (WWW), 2011.
[6] C. Budak, D. Agrawal, A.E. Abbadi, Diffusion of information in social networks: is it all local? in: Proceedings of the IEEE 12th International Conference on Data Mining (ICDM), 2012.
[7] V. Chaoji, S. Ranu, R. Rastogi, R. Bhatt, Recommendations to boost content spread in social networks, in: Proceedings of the 21st International Conference on World Wide Web (WWW), 2012.
[8] W. Chen, L.V.S. Lakshmanan, C. Castillo, Information and Influence Propagation in Social Networks, Morgan & Claypool Publishers, 2013.
[9] W. Chen, W. Lu, N. Zhang, Time-critical influence maximization in social networks with time-delayed diffusion process, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI), 2012.
[10] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2009.
[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, third ed., The MIT Press, 2009.
[12] T.N. Dinh, H. Zhang, D.T. Nguyen, M.T. Thai, Cost-effective viral marketing for time-critical campaigns in large-scale social networks, IEEE/ACM Trans. Netw. 22 (6) (2013) 2001–2011.
[13] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2001.
[14] Facebook reports fourth quarter and full year 2015 results, http://investor.fb.com/releasedetail.cfm?ReleaseID=952040.
[15] M. Gomez-Rodriguez, J. Leskovec, A. Krause, Inferring networks of diffusion and influence, ACM Trans. Knowl. Discov. Data (TKDD) 5 (4) (2012). Article No. 21.
[16] A. Goyal, F. Bonchi, L. Lakshmanan, A data-based approach to social influence maximization, in: Proceedings of the VLDB Endowment, 2011.
[17] A. Goyal, F. Bonchi, L. Lakshmanan, S. Venkatasubramanian, On minimizing budget and time in influence propagation over social networks, Social Netw. Anal. Mining 3 (2) (2013) 179–192.
[18] X. He, G. Song, W. Chen, Q. Jiang, Influence blocking maximization in social networks under the competitive linear threshold model, in: Proceedings of the Twelfth SIAM International Conference on Data Mining (SDM), 2012.
[19] O. Hugo, E. Garnsey, The emergence of electronic messaging and the growth of four entrepreneurial entrants, New Tech. Based Firms New Millenn. 2 (2002) 97–124.
[20] L. Jin, Y. Chen, T. Wang, P. Hui, A.V. Vasilakos, Understanding user behavior in online social networks: a survey, IEEE Commun. Mag. 51 (9) (2013) 144–150.
[21] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2003.
[22] H. Li, S.S. Bhowmick, A. Sun, CINEMA: conformity-aware greedy algorithm for influence maximization in online social Networks, in: Proceedings of the 16th International Conference on Extending Database Technology (EDBT), 2013.

[23] Y. Li, M. Qian, D. Jin, P. Hui, A.V. Vasilakos, Revealing the efficiency of information diffusion in online social networks of microblog, Inf. Sci. 293 (2015) 383–389.
[24] G. Nemhauser, L. Wolsey, M. Fisher, An analysis of the approximations for maximizing submodular set functions–i, Math. Program. 14 (1978) 265–294.
[25] M. Rezvani, W. Liang, W. Xu, C. Liu, Identifying top-k structural hole spanners in large-scale social networks, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM), 2015.
[26] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2002.
[27] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), 2014.
[28] J. Ugander, L. Backstrom, C. Marlow, J. Kleinberg, Structural diversity in social contagion, Proc. Nat. Acad. Sci. (PNAS) 109 (2012) 5962–5966.
[29] C. Wang, W. Chen, Y. Wang, Scalable influence maximization for independent cascade model in large-scale social networks, Data Mining Knowl. Discov. 25 (2012) 545–576.
[30] Y. Wang, G. Cong, G. Song, 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 (SIGKDD), 2010.
[31] Y. Wang, A.V. Vasilakos, J. Ma, N. Xiong, On studying the impact of uncertainty on behavior diffusion in social networks, IEEE Trans. Syst. Man Cybern. Syst. 45 (2) (2015a) 185–197.
[32] Y. Wang, A.V. Vasilakos, J. Ma, VPEF: a simple and effective incentive mechanism in community-based autonomous networks, IEEE Trans. Netw. Serv. Manage. 12 (1) (2015b) 75–86.
[33] Y. Wang, A.V. Vasilakos, Q. Jin, J. Ma, PPRank: economically selecting initial users for influence maximization in social networks, IEEE Syst. J. (2015c) 1–12. To appear.
[34] G. Wei, P. Zhu, A.V. Vasilakos, Y. Mao, J. Luo, Y. Ling, Cooperation dynamics on collaborative social networks of heterogeneous population, IEEE J. Select. Areas Commun. 31 (6) (2013) 1135–1146.
[35] Y. Zhang, X. Li, J. Xu, A.V. Vasilakos, Human interactive patterns in temporal networks, IEEE Trans. Syst. Man Cybern. Syst. 45 (2) (2015) 214–222.