IMPC: Influence Maximization based on Multi-Neighbor Potential in Community Networks

Jiaxing Shang[a,b,*], Hongchun Wu[a,b], Shangbo Zhou[a,b], Jiang Zhong[a,b], Yong Feng[a,b], Baohua Qiang[c]

[a] *College of Computer Science, Chongqing University, Chongqing 400044, China*
[b] *Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing 400044, China*
[c] *Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China*

Abstract

The study of influence maximization (IM) has attracted many scholars in recent years due to its import practical values. Given a social network, it aims at finding a subset of k individuals as seed nodes which can trigger the maximum influence cascade through the network under a predefined diffusion model. Kempe et al. fist formulated influence maximization as a discrete optimization problem and proved its NP-hardness and submodularity, based on which they further proposed a greedy approach with guaranteed solution accuracy. Unfortunately, the greedy algorithm was also known for its extremely low time efficiency. To solve this problem more efficiently, many research works were proposed in recent years. However, these studies either make sacrifices in solution accuracy or require huge memory consumption. Besides, only a handful of research works can handle mega-scale networks with millions of nodes and edges. To solve this problem both efficiently and effectively, in this paper we propose IMPC: an influence maximization framework based on multi-neighbor potential in community networks. In our approach the influence diffusion process is divided into two phases: (i) multi-neighbor potential based seeds expansion; and (ii) intra-community influence propagation. Based on this framework we derive an objective function to evaluate the overall influence as a combination of the influence during the two phases. We theoretically prove that the objective function is submodular and design an efficient greedy algorithm to find the seed nodes.

We evaluate the performance of our framework on eight real-world networks which scale up to millions of nodes and hundreds of millions of edges. Experimental results show that our approach can significantly outperform other state-of-the-art algorithms in terms of time and space efficiency with no compromise on solution accuracy.

Keywords: Influence maximization, Community structure, Multi-neighbor potential, Diffusion model, Computational complexity

1. Introduction

The study of influence maximization (IM) has draw much attention of scholars in the field of social network analysis and data mining in recent years, due to its many practical applications such as viral marketing, virus controlling, information diffusion etc. [1, 2]. For example, in viral marketing, when a company wants to promote a new product or innovative idea to the population, suppose the company plains to choose some people as volunteers and let them try this product for free, hoping that these people could recommend it to their family or friends. However, for many companies their budget is very limited, so the natural question is: how to select the "best" initial adopters with limited budget so that the product can influence the maximum number of individuals through the "world-of-mouth" [3] effect? Mathematically speaking, it is defined as the problem of finding k seed nodes in a network such that the objective function which evaluates the total influence can be maximized.

1.1. Diffusion model

The research of influence maximization cannot be separated from the underlying diffusion models [4], which play a vital role. Two widely considered models are the linear threshold (LT) model and the independent cascade

[*] Corresponding author. Tel.: +86 17783974208
Email address: shangjx@cqu.edu.cn (Hongchun Wu)

(IC) model [5], both of which are time-discrete. In these diffusion models, a node is associated with two states, namely, *active* or *inactive*. Active means the node has been successfully influenced, e.g., adopted a new product or accepted a new idea. If a node is active, it will propagate the influence to its inactive neighbors. Inactive means the influence has not yet reached the node or the influence has reached but the node refuses to be influenced. At the beginning all nodes in the network are in the *inactive* state, then k seed nodes are activated as diffusion sources and influence starts propagating from them, until the diffusion process finally tops.

IC model: Under the IC model, each directed edge (u, v) of the network is associated with a propagation probability p_{uv}. At step t, for each active vertex u, it will try to activate each of its inactive neighbor, suppose v, and succeed with probability p_{uv}. Once v is successfully activated, it will remain active during the rest of the time, and from the next step $t + 1$, it will pass the influence to its neighbors in the same manner. On the contrary, if u falls to activate v, u will not make any more attempts in the future, i.e., edge (u, v) can be regarded as blocked. If v has multiple active neighbors, their attempts will be independent of each other. The propagation stops when no more individuals can be further activated.

LT model: Under the LT model, wether an individual can be activated depends on its active neighbors in the current step. For an inactive vertex v, each of its incoming edge (u, v) has a weight parameter $b_{u,v}$ indicating how v is influenced by u. The weight parameters should satisfy the constraint $\sum_{u \text{ neighbor of } v} b_{uv} \leq 1$ for each v. Under this setting, each node v is given an activation threshold θ_v, which is usually randomly selected from the interval [0,1] at the beginning of diffusion. At step t, for any inactive node v, once the condition $\sum_{u \text{ active neighbor of } v} b_{uv} \geq \theta_v$ is achieved, v will be activated and remain active from step $t + 1$. The propagation stops until no more individuals can be further activated.

1.2. Influence maximization

The influence maximization problem was firstly formulated by Kempe et al. [5]. Given a network $G(V, E)$, influence maximization aims to find a subset S of $k = |S|$ vertices such that the diffusion orients from S can trigger the maximum scale of influence propagation (defined as $\sigma(S)$) through a given diffusion model, i.e., $S^* = \arg_S \max \sigma(S)$. Kempe et al. proved that under both linear threshold (LT) and independent cascade (IC) diffusion models, this problem is NP-hard and the objective function is submodular. A function $f(\cdot)$ that maps a subset to a non-negative real number, i.e., $f : 2^V \rightarrow R$, is called submodular if it has the following two properties: (i) monotone increasing, i.e., $f(S) \leq f(T), \forall S \subseteq T \subseteq V$, and (ii) diminishing returns, i.e., $\forall v \in V, S \subseteq T$, we have $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$. The diminishing returns property means that the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. Based on submodularity property [6, 7] of the objective function, Kempe et al. provided a "hill-climbing" greedy algorithm which iteratively selects the new seed node with the maximum marginal gain. They theoretically proved that the solution provides a factor of $(1 - 1/e - \epsilon)$ (about 63%) performance guarantee to the optimal value, where e is the base of the natural logarithm and ϵ is any positive real number. The second item $1/e$ orients from the submodular function itself while the third item ϵ is the error of approximating the objective function using Monte-Carlo simulations. Kempe et al. showed in real experiments that the solution provided by the greedy algorithm is quite close to the optimal solution. However, the greedy algorithm is also known for its extremely low time efficiency, since it requires tens of thousands of Monte-Carlo simulations to accurately approximate the objective function, which prohibits its applications on large networks, even middle-sized ones.

To improve the efficiency of the traditional greedy algorithm, many research works have been proposed by researchers in recent years. Some are based on influence simulation, such as the CELF [8] and CELF++ [9] algorithms. Some research works approximate the objective function through shortest paths [10, 11]. There are also algorithms which simply select top k nodes according to some heuristic metrics, such as the degree centrality [12], closeness centrality [13], etc. However, these studies either make sacrifices in solution accuracy (e.g., the node centrality based heuristic algorithms) or require huge memory consumption (e.g., the influence path based algorithms), other approaches may depend on specific diffusion model settings. Besides, only a handful of research works can handle mega-scale networks with millions of nodes and hundreds of millions of edges.

To tackle the influence maximization problem both efficiently and effectively, in this paper we propose **IMPC**: an **I**nfluence **M**aximization framework based on **M**ulti-neighbor **P**otential in **C**ommunity networks. Community structure [14] is characterized as the division of network nodes into subgroups, within which nodes are densely connected, while among which they are sparsely connected. Our work, which is based on the independent cascade model, takes advantage of the connection characteristics of community structure. It can largely reduce the running time as compared with the fastest benchmark proposed very recently [15]. In our framework we divide the influence diffusion process into two phases: (i) multi-neighbor potential based seeds expansion; and (ii) intra-community influence propagation. In the first phase we mainly focus on the influence propagation from seed nodes to their multi-neighbors, i.e., the neighbors and neighbors' neighbors of S, whose influence is approximated using their potential values. We elaborately design the algorithm such that these values can be analytically computed in a very efficient manner. Since this phase allows influence propagation from seed nodes to different communities, we regard it as seeds expansion. In the second phase, we mainly focus on the influence propagation from the newly activated individuals to their neighbors within the same community, namely, we use the intra-community influence propagation to approximate the diffusion on the rest nodes of the network. Based on this framework we derive an objective function to evaluate the overall influence spread as a combination of the influence diffusion through the two phases. We theoretically prove that the objective function is submodular and propose an efficient greedy algorithm to find the seed nodes.

We evaluate the performance of our algorithm on eight real-world networks which scale up to millions of nodes and hundreds of millions of edges. Experimental results show that our approach can significantly outperform other state-of-the-art algorithms in terms of time and space efficiency with no compromise on solution accuracy.

We summarize the contributions of our work as follows:

- We propose IMPC: a multi-neighbor potential based framework to efficiently and effectively solve the influence maximization problem. Our framework takes advantage of the community structure and divide the influence diffusion process into two phases, based on which we derive an objective function to approximate the overall influence.

- We theoretically prove that the derived objective function is submodular and provide an efficient greedy algorithm to find the seed nodes. We further develop several techniques to speed up the IMPC algorithm, including the incremental computation of the marginal gain and the CELF acceleration strategy.

- We theoretically prove the time complexity of our IMPC algorithm is linear to the networks size, i.e., the number of network edges, and our algorithm requires almost no extra memory except for storing the network.

- We conduct extensive experiments on eight real-world networks which scale up to millions of nodes and hundreds of millions of edges to show its superiority over other state-of-the-art algorithms in terms of time efficiency, solution accuracy and memory usage.

The rest of this paper is organized as follows. Section 2 review the literature on influence maximization. Section 3 gives some preliminaries about the influence maximization problem and social graph definition. Section 4 introduces our IMPC framework and the proposed algorithm. Section 5 presents the evaluation framework, including the datasets, evaluation metrics, baseline methods, and experimental procedure. Experimental results are presented in Section 6. Section 7 concludes this paper.

2. Literature review

2.1. Influence Maximization

Since the influence maximization problem was firstly formulated as an optimization problem by Kempe et al. [5], numerous research works have been proposed in recent years to solve the problem. We roughly categorize these studies into five groups: (1) simulation-based algorithms; (2) node centrality-based heuristic algorithms; (3) influence path-based algorithms; (4) RIS (Reverse Influence Sampling) based algorithms; and (5) community based algorithms.

2.1.1. Simulation-based algorithms

The general idea behind simulation-based methods is to design new tricks to reduce the number of MC simulations without compromise on solution accuracy.

Leskovec et al. [8] proposed CELF algorithm to improve the time efficiency of the traditional greedy algorithm through a lazy forward strategy. In their approach, they maintained the marginal gains of individuals with respect to the current seed set and partially update their marginal gains (through MC simulations) only when it is necessary. Inspired by similar idea, Goyal et al. [9] proposed CELF++, an improved version of CELF. Compared to CELF, CELF++ not only stores the marginal gains of individuals corresponding to the current seed set, but also maintains their marginal gains with respect to a super set of the seed set, which further reduces the unnecessary checking of candidates. Zhou et al. [16] proposed a new upper bond of the influence spread under IC and LT models. The new upper bound was then utilized to prone the MC simulations of the greedy algorithm. Experimental results show their approach achieves about $2 \sim 10$ times speedup as compared with the CELF algorithm.

A common disadvantage shared by the simulation-based algorithms is that they all require a large number of MC simulations to obtain a close approximation to the objective function, making their worst-case time complexity as much as $O(knmR)$, thus cannot scale to large networks.

2.1.2. Node centrality based algorithms

Node centrality-based algorithms mine the influential seed nodes according to some centrality measures defined on a node or a set of nodes.

Chen et al. [12] proposed the SD (Single Discount) algorithm by taking into consideration the impact of previously chosen seeds on current candidates. Specifically, the algorithm iteratively selects a new seed node with the maximum degree. When a node is chosen, the degree of each of its neighbor will decreased (discounted) by a unit, in order to avoid influence overlap. Liu et al. [17] proposed a Group-PageRank based algorithm, where the influence from seed set S to a node v is modeled as the combination of influence of its neighbors, which can be recursively computed. Recently, Ma et al. [18] proposed a hybrid degree centrality-based algorithm. Their approach is based on the observation that the performance of different centrality measures may change with different spreading probabilities, so they proposed a local centrality measure and then integrated it with degree centrality by considering the spreading probabilities. Zhu et al. [19] proposed SHIM, a structure hole-based influence maximization algorithm which utilizes structure hole as the centrality measurement. They first identified structure hole spanners whose structure hole value is above the given threshold, followed by computing the influence capability of each structure hole spanner. After that, they selected the top-k seeds by combining the structure hole value and influence value. Liu et al. [13] proposed a closeness centrality-based algorithm, where they defined a generalized closeness centrality (GCC) index by generalizing the closeness centrality index from a single node to a set of nodes. Riquelme et al. [20] mainly considered the centrality measures under the LT diffusion model, and proposed a new centrality measure named LTR (Linear Threshold Rank) to rank the users of the network.

Generally, most centrality based algorithms have linear time complexity, meaning that they can find the seed nodes in a very efficient manner. For example, the time complexity of the naive degree centrality-based algorithm is $O(m)$. However, since they take little consideration of the diffusion process, they usually generate inaccurate solutions, making their results unreliable.

2.1.3. Influence path based algorithms

The general idea behind influence path-based algorithms is to reduce the randomness of diffusion models by restricting influence propagation on specific paths, such that the objective function can be efficiently calculated without MC simulations.

Chen et al. studied influence path-based methods under both IC and LT models. For the IC model, they proposed MIA (Maximum Influence Arborescence) and PMIA (Prefix excluding MIA) [10], both of which are based on arborescence, a local tree structure defined for a node to approximate its influence area. For the LT model, they proposed LDAG (Local Directed Acyclic Graphs) [21], which is a local graph structure centered around a node to represent its local influence area. One limitation of Chen et al.'s approach is that they only considered the influence paths with the maximum propagation probabilities, which may affect the solution accuracy on some datasets. To address this issue, Kim et al. [11] proposed IPA (Independent Path Algorithm), an influence path-based method which simultaneously considers multiple influence paths and assumes that their propagations are independent of each other. Liu et al. [22] proposed influence path-based algorithms to solve the time-constrained influence maximization problem. In their approach they first construct influence spreading paths, then some of the paths are pruned to reduce the computational overload. Ko et al. [23, 24] considered the IC model and proposed a path-based influence approximation method which can remedy the drawback of existing path-based algorithms. Unlike previous path-based methods which utilize influence path to approximate the amount of influence a seed node can exert to non-seed nodes, their approach focus on the amount of influence a non-seed node can receive from a given seed set.

Compared with simulation and centrality-based algorithms, influence path-based algorithms reach a tradeoff between running time and solution quality. However, a critical deficiency of these methods is that they usually require a huge amount of memory to maintain the influence paths, which restricts their space efficiency.

2.1.4. Reverse influence sampling based algorithms

The idea behind RIS (Reverse Influence Sampling) [25] is to use "node-centric" sampling strategies to approximate the objective function. The general framework is: select a node v uniformly at random, and determine the set of nodes that would have influenced v. If this process is repeated multiple times, and a certain node u appears often as an "influencer", then u is likely a good candidate for the most influential node. It can be proved that the probability a node u appears in a set of influencers is proportional to its expected influence on the network.

Tang et al. [26] proposed the TIM+ (Two-phase Influence Maximization) algorithm which generalize the RIS framework from IC model to both IC and LT models. Their approach consists of two phases where the first phases estimates an approximation parameter θ while the second phase generate θ samples to derive to best seed set. In [27] they further proposed the improved version of TIM+, i.e., IMM (Influence Maximization via Martingales) algorithm, which takes advantage of martingales to improve its time and memory efficiency. Based on the idea of RIS, Nguyen et al. [28] proposed the SSA (Stop and Stare Algorithm) and D-SSA (Dynamic SSA) algorithms which focus on characterizing the minimum number of RIS samples needed to achieve $(1 - e - \epsilon)$ approximation. The algorithm keeps generating samples and *stop*s at exponential check points to verify (*stare*) if there is adequate statistical evidence on the solution quality for termination. In a recent research, Huang et al. [29] conducted a rigorous theoretical and experimental analysis of SSA and D-SSA, and proposed a revised version. Recently, Wang et al. [30] proposed a novel bottom-k sketch based RIS framework, namely BKRIS, which brings the order of samples into the RIS framework to improve its time efficiency.

Although the RIS based algorithms exhibit competitive performance in terms of effectiveness and efficiency, they still have their inherent drawbacks. Despite that these algorithms can find a seed set S with approximation guarantee, they cannot make sure that the subsets of S will also provide the same approximation guarantee.

2.1.5. Community based algorithms

Community based influence maximization algorithms usually use the influence of a node within its own community to approximate its influence on the whole network. Benefit from the fact that a community's size is usually much smaller than the whole network size, the influence of a node within its own community can be computed more efficiently.

Cao et al. [31] proposed the first community based influence maximization algorithm OASNET (Optimal Allocation in a Social NETwork). They assumed that different communities are independent of each other and influence cannot spread across different communities. The selection of seed nodes contains two phrases. In the first phase, from each community the algorithm selects k nodes using traditional greedy algorithm, resulting in a total number of $C \cdot k$ candidate nodes. In the second phase the algorithm selects k nodes as the seed set S from the $C \cdot k$ candidates using dynamic programming. Zhang et al. [32] worked on identifying the influential nodes on networks with community structure. The authors firstly constructed an information transfer probability matrix from the weighted network. Then they applied the k-medoid clustering algorithm to identify the k seed (influential) nodes. Chen et al. [33] studied the community based influence maximization problem using the HD (heat diffusion) model and proposed the CIM (Community based Influence Maximization) algorithm to find the seed nodes through a three-step approach. Li et al. [34] considered the node conformity and proposed the community based influence maximization algorithm CINEMA (Conformity-aware INfluEnce MAximization), where the propagation probabilities were determined by both influence index and conformity index of individuals. Shang et al. [15] proposed CoFIM (Community based Framework for Influence Maximization) method which is based on the assumption that influence first propagates from seed nodes to their neighbors and then from these neighbors to other nodes within the same community. Under the weighted cascade model the authors derived a simple evaluation form of the influence objective function which is submodular and can be efficiently computed.

From the above introduction, we see that community based influence maximization algorithms are generally faster than traditional greedy algorithms. Moreover, since they usually make the assumption that different communities are isolated, these algorithms naturally support parallelization. However, the disadvantages of current community based algorithms are also obvious. First, evaluating the marginal gain of a node within its own community also needs Monte-Carlo simulations, which is time consuming and limits the application of these algorithms on large-scale networks. Second, since the algorithms use the influence of a node within its own community to approximate its influence on the whole network, the algorithm's accuracy will severely rely on the underlying community structure. Third, some community based algorithms rely on specific diffusion model, it is not sure whether these algorithms will perform well under classis IC and LT models.

2.1.6. Other influence maximization methods

Besides the above introduced methods, there are also many other influence maximization methods, such as the multiple selectors combination based algorithm [35], linear time algorithm [36], genetic algorithm based influence maximization [37], location-based influence maximization [38, 39], influence feature learning [40], etc.

2.2. Community detection

Community structure [14] is defined as the partition of network nodes into groups, within which nodes are densely connected while between which they are sparsely connected. Community detection is an important task in social network analysis and mining and it has been extensively studied in recent years. Since community detection is not our focus in this article, here we only introduce a small part of related works.

According to the type of community structure, community detection methods can be divided into two classes: *non-overlapping community detection* and *overlapping community detection*. Non-overlapping community detection methods assume that there is no overlap among different communities, i.e., each node can only be assigned to a unique community. Representatives include modularity-based algorithms [41, 42], label propagation-based algorithm [43], spectral clustering-based algorithm [44], incremental algorithms [45, 46], etc. Overlapping community detection methods allow overlap among different communities, i.e., each node can be assigned to more than one community. Examples include clique-based algorithm [47], label propagation-based algorithm [48], random walk-based algorithm [49], model-based algorithms [50, 51], etc.

2.3. Information diffusion

Information diffusion is a hot research topic in the fields of both complex networks and social networks analysis. Since information diffusion is beyond our focus, here we only introduce a few works which are more relevant to our work.

Some studies investigated information diffusion using the epidemic model. For example, Shang et al. [52, 53] studied how overlapping and non-overlapping community structure affects epidemic spreading on complex networks. Through simulation results on both synthetic and real-world datasets, they found that individuals located at the overlapping region of two communities play an important role in epidemic spreading. Some studies focus on how to identify the influential spreaders from a network based on the epidemic model. In [54] Guo et al. proposed an improved distance-based coloring method to identify multiple influential spreaders and evaluated their approach by simulating the SIR epidemic model on both synthetic and real-world complex networks. In [55] Bao et al. proposed a heuristic clustering (HC) algorithm to identify multiple influential spreaders based on the SIR model. Their first used the clustering algorithm to classify nodes into different clusters, then they chose the center nodes from different clusters to ensure that multiple influential spreaders are dispersively distributed in the network. Some recent works focus on studying social contagions. In [56] Wang et al. proposed a non-Markovian social contagion model in multiplex networks with inter-layer degree correlations to study the spreading behavior and developed an edge-based compartmental theory to describe the model. Through simulation results the authors found that the inter-layer degree correlations can affect the final behavior adoption size. Wang et al. further used the non-Markovian model to investigate how the heterogeneity of credibility levels affects social contagions within a population [57] and how time-delays affects social contagions [58].

3. Preliminaries

3.1. Influence maximization problem

Kempe et al. [5] first formulated influence maximization as an optimization problem, and proved its NP-hardness and submodularity. Under the two diffusion models, let S denote the seed set, i.e., the set of nodes activated at step $t = 0$, and $S(t)$ as the set of nodes activated at step t. Suppose the diffusion process ends at step T, then it is easy to see that $S(0) = S$ and $S(T) = \Phi$. So the total number of influenced individuals, i.e., overall number of activated nodes through the diffusion process is $\sum_{t=0}^{T} |S(t)|$, which is the object we aim to maximize. However, since the diffusion models are stochastic, Kempe et al. used the expected number of overall activated nodes, namely $\sigma(S)$ as the objective function. The influence maximization problem is defined as follows:

Definition 1. (Influence Maximization [5]) Given a network and a diffusion model, the influence maximization problem aims to find a subset S of k nodes ($|S| = k$), such that the expected number of overall activated nodes $\sigma(S)$ is maximized:

$$S^* = \arg_S \max \sigma(S) \tag{1}$$

Kempe et al. [5] proved that under IC and LT models, the influence maximization problem defined above is NP-hard and the objective function $\sigma(S)$ is submodular. A nonnegative set function $f : 2^V \rightarrow R$ is submodular if it has the diminishing returns property:

$$f(S \cup v) - f(S) \geq f(T \cup v) - f(T), \forall v \in V, S \subseteq T \tag{2}$$

Based on the submodualrity, Kempe et al. further proposed a “hill-climbing” greedy algorithm (as shown in Algorithm 1) and proved that the algorithm provides a factor of $(1 - 1/e - \epsilon)$ approximation guarantee to the optimal solution, as shown in Theorem 1.

Theorem 1. *(Kempe et al. [5])For influence maximization problem defined in Definition 1, the seed set generate by Algorithm 1 provides a factor of $(1-1/e-\epsilon)$ approximation to the optimal solution, i.e., $f(S) \geq (1-1/e-\epsilon)f(S^*)$, where S^* is the optimal solution.*

Algorithm 1 The hill-climbing greedy algorithm (Kempe et al. [5])

```
Input:
    Network $G$, number of seed nodes $k$
Output:
    Seed set $S$
Initialize: Let $S \leftarrow \Phi$.
for $i = 1$ to $k$ do
    $v = \arg_u \max\{\sigma(S \cup \{u\}) - \sigma(S)\}$
    // $\sigma(*)$ is approximated using Monte-Carlo simulations
    $S \leftarrow S \cup \{v\}$
end for
return S
```

However, as Chen et al. had shown in [12], the time complexity of the simple greedy algorithm is as high as $O(knRm)$, where k is the number of seed nodes, n is the number of network nodes, m is the number of network edges, and R is the number of Monte-Carlo simulations needed to approximate $\sigma(S)$. Chen et al. [10, 21] also proved that under both IC and LT models, computing the exact value of $\sigma(S)$ is #P-hard. To obtain a good approximation to $\sigma(S)$, the simple greedy algorithm needs tens of thousands of Monte-Carlo simulations, which severely affects its time efficiency.

3.2. Network preliminaries

Without loss of generality, we consider an directed unweighted[1] network $G = (V, E)$, where V is the set of nodes, E is the set of edges. G has adjacency matrix A, where:

$$A_{i,j} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

For any node u, its outgoing neighbor set is:

$$N^+(u) = \{v | A_{u,v} > 0\} \tag{4}$$

[1] Our framework is also applicable for undirected or weighted networks

Similarly, we define the incoming neighbor set of u as $N^-(u) = \{v|A_{v,u} > 0\}$. Let $S \subseteq V$ denotes any node set, then the outgoing neighbor set of S is:

$$N^+(S) = \bigcup_{u \in S} N(u) \tag{5}$$

Most real world networks exhibit some degree of community structure. The community structure of G is defined as the partition of its nodes into subgroups, i.e., $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$, where the i-th community C_i contains multiple nodes (in this paper we only consider non-overlapping community structure, so there is no overlap across different communities). For any node v, let $C(u)$ denotes the community of u, i.e., $u \in C(u)$. Then we further have the neighbor community set of u as:

$$NC(u) = \{C(v)|v \in N^+(u)\} \tag{6}$$

Similarly, the neighbor community set of S is:

$$NC(S) = \{C(v)|v \in N^+(S)\} \tag{7}$$

4. IMPC framework

In this paper we mainly focus on the IC model, where each edge (u, v) is associated with a propagation probability p_{uv}, indicating the opportunity that u can successfully activate its neighbor v when u is active. Our approach is based on the observation of the IC unfolding process: influence first propagates from seed nodes S to their neighbors $N^+(S)$, and then to their neighbors's neighbors, i.e., $N^+(N^+(S))$, and finally to the rest nodes of the network. It has been shown in previous studies that community structure can be helpful in solving the influence maximization problem. However, unlike traditional community based studies which restrict influence diffusion within communities at the very beginning, our framework considers both the intra- and inter-community diffusions in different phases. In our framework the influence diffusion process is divided into two phases: (i) multi-neighbor potential based seeds expansion and (ii) intra-community influence propagation. In the following subsections we will give detailed illustration of the two phases.

4.1. Multi-neighbor potential based seeds expansion

In the first phase, we focus on two steps of influence propagation, i.e., the influence first propagates from S to $N^+(S)$, which will produce a set of active neighbors $S_1 \subseteq N^+(S)$. Then the influence will continue propagating from S_1 to $S_2 \subseteq N^+(S_1)$. In this phase, we allow influence to propagate across different communities, so that the seed nodes can be "expanded" to other communities. The key problem in this phase is: how to analytically compute the influence spread from: $S \to S_1$ and $S_1 \to S_2$?

During the first step of propagation, for each node $v \in N^+(S)$, the probability that v gets activated during propagation $S \to S_1$ is:

$$\mathcal{P}_{S \to v} = \mathcal{PS}(v) = 1 - \prod_{u \in S \cap N^-(v)} (1 - p_{uv}) \tag{8}$$

If we concentrate on a specific node u and only consider one step of influence propagation, then we can take advantage of the node potential introduced by Wang et al. [59] to evaluate the influence of node u. For each node $u \in S$, its influence potential is defined as:

$$\mathcal{PT}(u) = \sum_{v \in N^+(u)} p_{uv} \tag{9}$$

So the total influence after the first step of influence propagation can be approximated by:

$$f_1(S) = |S| + \sum_{u \in S} \mathcal{PT}(u) \tag{10}$$

where $|S|$ indicates the influence of the seed nodes on themselves. We should note that the potential of different seeds in S may overlap, so Eq. 10 is an approximation to the real influence of one step propagation. Obviously, considering only one step of influence propagation is inadequate for accurate influence computation. So we further consider the influence propagation from $S_1 \rightarrow S_2$ and approximate the influence using potential values of the nodes in $S_1 \subseteq N^+(S)$ which are activated after the first step of influence propagation. Considering two steps of influence propagation, the total influence spread is:

$$f_1(S) = |S| + \sum_{u \in S} \mathcal{PT}(u) + \sum_{v \in S_1} \mathcal{PT}(v) \tag{11}$$

We should note in Eq. 11 that the computation of $f_1(S)$ relies on S_1, which is not known to us and cannot be analytically obtained due to the stochastic property of the diffusion model. However, we know the probability that a node $v \in N^+(S)$ belongs to S_1, which is $\mathcal{PS}(v)$, as defined in Eq. 8. So the total influence spread through the first phase is:

$$f_1(S) = |S| + \sum_{u \in S} \mathcal{PT}(u) + \sum_{v \in N^+(S)} \mathcal{PS}(v)\mathcal{PT}(v) \tag{12}$$

4.2. Intro-community influence propagation

The second phase evaluates the influence spread from S_2 to the rest of network. Specifically, we take advantage of community structure and assume that influence from nodes in S_2 to the rest of network can only spread within their own communities. Our model is based on the observation of previous studies that strong community structure may restrict epidemic spreading within its own community [53]. For community $C_i \in \mathcal{C}$, it is intuitive to see that the influence spread mainly depends on two factors: (i) the size of the community $|C_i|$, and (ii) the number of active nodes located in that community, i.e., $|S_2 \cap C_i|$. To have a reasonable evaluation form of the influence spread, the design of the objective function should follow some principles:

- The larger the $|C_i|$ and $|S_2 \cap C_i|$ are, the higher the influence spread value should be.
- If the community size is much larger than the size of $S_2 \cap C_i$, i.e., $|C_i| \gg |S_2 \cap C_i|$, then compared to $|C_i|$, $|S_2 \cap C_i|$ should play a more important role in the final influence spread value.
- If the two sizes are similar, i.e., $|C_i| \approx |S_2 \cap C_i|$, then the influence spread value should be restricted by the community size $|C_i|$.
- The evaluation function $g(S_2, C_i)$ should be submodular about S_2.

Based on the above considerations, we define the influence spread function from nodes in S_2 to their own communities as:

$$g(S_2, C_i) = \alpha(1 - \beta \cdot e^{-\frac{2|C_i|\cdot|S_2 \cap C_i|}{|C_i| + |S_2 \cap C_i|}}) \tag{13}$$

where α, β are non-negative parameters. The total influence spread through the second phase is:

$$f_2(S_2) = \sum_{C_i \in \mathcal{C}} \alpha(1 - \beta \cdot e^{-\frac{2|C_i|\cdot|S_2 \cap C_i|}{|C_i| + |S_2 \cap C_i|}}) \tag{14}$$

However, similar to S_1, we cannot analytically obtain S_2 due to the stochastic nature of the diffusion model. To calculate the intra-community influence analytically and efficiently, we may have two alternative heuristic approaches, i.e., to use $N^+(S)$ or $N^+(N^+(S))$ to approximate S_2. Intuitively, $N^+(N^+(S))$ would be a better choice, however, in real-world applications, the propagation probabilities are usually very small, as a result, most of the nodes in $N^+(N^+(S))$ will remain inactive after two steps of influence propagation. On the contrary, $N^+(S)$ will include a large proportion of active individuals after two steps of propagation, making itself a better choice. Another reason for choosing $N^+(S)$ is that it can be more efficiently computed as compared to $N^+(N^+(S))$. So in our approach we use $N^+(S)$ to approximate S_2 and consider the influence propagation from $N^+(S)$ to the rest of the network as an approximation to the real influence of the second phase. So the overall influence through the second phase is:

$$f_2(S) = \sum_{C_i \in \mathcal{C}} \alpha(1 - \beta \cdot e^{-\frac{2|C_i|\cdot|N^+(S) \cap C_i|}{|C_i| + |N^+(S) \cap C_i|}}) \tag{15}$$

Putting the influence spread in the two phases together, we get the overall influence spread in the entire network as:

$$
\begin{aligned}
f(S) &= f_1(S) + f_2(S) \\
&= |S| + \sum_{u \in S} \mathcal{PT}(u) + \sum_{v \in N^+(S)} \mathcal{PS}(v)\mathcal{PT}(v) \\
&+ \sum_{C_i \in \mathcal{C}} \alpha(1 - \beta \cdot e^{-\frac{2|C_i|\cdot|N^+(S)\cap C_i|}{|C_i|+|N^+(S)\cap C_i|}})
\end{aligned}
\tag{16}
$$

4.3. Submodularity of the objective function

In this subsection, we show the submodularity of the derived objective function as defined in 16. Before proving the submodularity, we first prove the following lemma:

Lemma 2. *If $f(S)$ is submodular about S, $t(x)$ is a monotone increasing concave function, then $g(S) = t(f(S))$ is also submodular about S.*

Proof. (1) We first prove the monotone increasing property of $g(S)$, i.e.,

$$\forall S \subseteq T \subseteq V, g(S) \leq g(T)$$

Since $f(S)$ is submodular, we have $f(S) \leq f(T)$. Because $t(x)$ is a monotone increasing function, we have $t(f(S)) \leq t(f(T))$, i.e., $g(S) \leq g(T)$, therefore the $g(S)$ satisfies the monotone increasing property.

(2) We then prove the "diminishing returns" property of $g(S)$, i.e., $\forall v \in V, S \subseteq T \subseteq V$,

$$g(S \cup \{v\}) - g(S) \geq g(T \cup \{v\}) - g(T)$$

To prove this, we first prove that $\forall x \leq y, a \geq 0$,

$$t(x + a) - t(x) \geq t(y + a) - t(y)$$

The idea is based on calculus as follows,

$$
\begin{aligned}
t(x + a) - t(x) &= \int_x^{x+a} t'(u)du \\
t(y + a) - t(y) &= \int_y^{y+a} t'(u)du
\end{aligned}
$$

Since $y \geq x$, with out generality, let $\delta = y - x \geq 0$ then

$$t(y + a) - t(y) = \int_x^{x+a} t'(u + \delta)du$$

Since $t(x)$ is concave, we have $t''(x) \leq 0$, i.e., $t'(x)$ is monotone decreasing about x, so we have

$$\int_x^{x+a} t'(u)du \geq \int_x^{x+a} t'(u + \delta)du$$

i.e.,

$$t(x + a) - t(x) \geq t(y + a) - t(y), \forall x \leq y, a \geq 0$$

$\forall S \subseteq T \subseteq V, v \in V$, let

$$
\begin{aligned}
f(S) = x, f(S \cup \{v\}) = x + a \\
f(T) = y, f(T \cup \{v\}) = y + b
\end{aligned}
$$

Since $f(S)$ is submodular, we have $a \geq b$. Since $t(x)$ is monotone increasing, we have $t(y+a) \geq t(y+b)$, so we further have

$$t(x+a) - t(x) \geq t(y+a) - t(y) \geq t(y+b) - t(y)$$

Substituting $x, x+a, y, y+b$ with $f(S), f(S \cup \{v\}), f(T), f(T \cup \{v\})$ respectively, we have

$$t(f(S \cup \{v\})) - t(f(S)) \geq t(f(T \cup \{v\})) - t(f(T))$$

i.e.,

$$g(S \cup \{v\}) - g(S) \geq g(T \cup \{v\}) - g(T))$$

So $g(S)$ is submodular about S.

□

Based on Lemma 2, we now show the submodularity of the derived objective function as defined in Eq. 16.

Theorem 3. *The influence function $f(S)$ defined in Eq. 16 is submodular.*

Proof. (1) Fist, we prove that $f_1(S) = |S| + \sum_{u \in S} \mathcal{PT}(u) + \sum_{v \in N^+(S)} \mathcal{PS}(v)\mathcal{PT}(v)$ is submodular. It is easy to see that both $|S|$ and $\sum_{u \in S} \mathcal{PT}(u)$ are submodular functions about S, so the key is to prove that:

$$g(S) = \sum_{v \in N^+(S)} \mathcal{PS}(v)\mathcal{PT}(v) = \sum_{v \in N^+(S)} \mathcal{P}_{S \to v}\mathcal{PT}(v)$$

is submodular, i.e., $\forall S \subseteq T \subseteq V, x \in V \setminus T$,

$$g(S \cup \{x\}) - g(S) \geq g(T \cup \{x\}) - g(T) \tag{17}$$

From Eq. 8 we can easily prove that

$$\mathcal{P}_{S \to v} \leq \mathcal{P}_{T \to v}, \forall S \subseteq T, v \in V$$

Consider seed set S, if we add a new seed x to S, then for any node $v \in N^+(x)$, we have:

$$1 - \mathcal{P}_{S \cup \{x\} \to v} = (1 - p_{xv})(1 - \mathcal{P}_{S \to v})$$

Then we further have the marginal gain as:

$$\mathcal{P}_{S \cup \{x\} \to v} - \mathcal{P}_{S \to v} = p_{xv}(1 - \mathcal{P}_{S \to v}) \tag{18}$$

Similarly, for seed set T, we have:

$$\mathcal{P}_{T \cup \{x\} \to v} - \mathcal{P}_{T \to v} = p_{xv}(1 - \mathcal{P}_{T \to v})$$

Since $\mathcal{P}_{S \to v} \leq \mathcal{P}_{T \to v}$, we have:

$$\mathcal{P}_{S \cup \{x\} \to v} - \mathcal{P}_{S \to v} \geq \mathcal{P}_{T \cup \{x\} \to v} - \mathcal{P}_{T \to v}$$

Since the above inequation holds for all $v \in V$, and $g(S) = \sum_{v \in N^+(S)} \mathcal{P}_{S \to v}\mathcal{PT}(v)$, we can directly get the formula as defined in Eq. 17, so $f_1(S)$ is submodular about S.

(2) Second, we prove that $f_2(S)$ is a submodular function about S.

It has been proved by Shang et al. [15] that $|N^+(S)|$ is submodular about S, so $\forall C_i \in \mathcal{C}$, $|N^+(S) \cap C_i|$ is also submodular about S.

Let $t(x) = \alpha(1 - \beta e^{-\frac{2cx}{c+x}})$, then we have:

$$t'(x) = 2\alpha\beta \frac{c^2}{(c+x)^2} e^{-\frac{2cx}{c+x}} \geq 0$$

$$t''(x) = -4\alpha\beta c^2 \frac{(x+c+c^2)}{(c+x)^4} e^{-\frac{2cx}{c+x}} \leq 0$$

It is easy to see that $t(x)$ is a monotone increasing concave function. Since $|N^+(S) \cap C_i|$ is submodular about S, from Lemma 2, we can directly prove that

$$g(C_i, S) = \alpha\big(1 - \beta \cdot e^{-\frac{2|C_i|\cdot|N^+(S)\cap C_i|}{|C_i|+|N^+(S)\cap C_i|}}\big)$$

is also submodular about S. So $f_2(S) = \sum_{C_i \in \mathcal{C}} g(C_i, S)$ is submodular about S.

In sum, $f(S) = f_1(S) + f_2(S)$ is submodular about S. □

4.4. IMPC algorithm

Unlike traditional greedy algorithms which use Monte-Carlo simulations to approximate the real influence, our IMPC algorithm maximizes the derived objective function as defined in Eq. 16. Moreover, we use the following two techniques to speed up our IMPC algorithm.

- **Incremental computation of marginal gain**: As defined in Eq. 16, to compute the marginal gain for node $v \in V$ with respect to seed set S, we have to traverse all the neighbors of S, i.e., $N^+(S)$, to get the $\mathcal{PT}(v)$, $\mathcal{PS}(v)$, and $N^+(S) \cap C_i$ values, which is not cost-effective. If we take a closer look at the objective function, we will find that only those nodes $w \in N^+(v)$ have to be considered in computing the marginal gain of v with respect to S. To incrementally update the marginal gain, we first compute the $\mathcal{PT}(v)$ values for all of the nodes $v \in V$ using Eq. 9 and store these potential values in an array $\mathcal{PT}$. We then define another array $\mathcal{RS}$, where for each node $v \in V$, $\mathcal{RS}(v) = 1 - \mathcal{P}_{S \to v}$. Intuitively, $\mathcal{RS}(v)$ indicates the probability that node v is not influenced by the current seed set S after one step of influence propagation. It is clear to see that $\mathcal{RS}(v) = 1, \forall v \in V$ when $S = \Phi$, and $\mathcal{RS}(v) = 0, \forall v \in S$ when $S \neq \Phi$. With the help of these data structures, we can incrementally compute the marginal gain, as shown in Algorithm 2.

Algorithm 2 MARGINAL_GAIN

```
Input:
    Network $G(V, E)$, community structure $\mathcal{C}$, current seed set $S$, candidate node $v$
Output:
    Marginal gain: $\Delta = f(S \cup \{v\}) - f(S)$
Initialize: $\Delta \leftarrow 1 + \mathcal{PT}(v)$
if $v \in N^+(S)$ then
    $\Delta \leftarrow \Delta - (1 - \mathcal{RS}(v))\mathcal{PT}(v)$
end if
for each $w \in N^+(v)$ do
    $\Delta \leftarrow \Delta + p_{vw}\mathcal{RS}(w)$ //According to Eq. 18
    if $w \notin N^+(S)$ then
        $\Delta \leftarrow \Delta + \beta(e^{-\frac{2|C(w)|\cdot(|N^+(S)\cap C(w)|+1)}{|C(w)|+|N^+(S)\cap C(w)|+1}} - e^{-\frac{2|C(w)|\cdot|N^+(S)\cap C(w)|}{|C(w)|+|N^+(S)\cap C(w)|}})$
        $|N^+(S) \cap C(w)| \leftarrow |N^+(S) \cap C(w)| + 1$
    end if
end for
return $\Delta$
```

- **CELF strategy**: Since the objective function is submodular (as shown in Theorem 3), we further take advantage of the CELF [8] strategy proposed by Leskovec et al. to accelerate the algorithm.

Algorithm 3 shows the pseudo-code of the IMPC algorithm. Lines 1-5 initialize the seed S and $\mathcal{RS}$, $\mathcal{PT}$ arrays. Line 7 finds the next seed node with the maximum marginal gain, which is incrementally computed using the subroutine 2. The finding of new seed node is accelerated by the CELF [8] strategy. After a new seed node is found, lines 8-12 update the seed set S and $\mathcal{RS}$ array.

4.5. Complexity analysis

In this subsection we analysis the time and space complexity of the IMPC algorithm.

Algorithm 3 The IMPC algorithm

```
Input:
    Network $G(V, E)$, community structure $\mathcal{C}$, number of seed nodes $k$
Output:
    Seed set $S$
 1: Initialize: $S \leftarrow \Phi$.
 2: for each $u \in V$ do
 3:     $\mathcal{RS}(u) = 1$
 4:     $\mathcal{PT}(u) = \sum_{v \in N^+(u)} p_{uv}$
 5: end for
 6: for $i = 1$ to $k$ do
 7:     $v = \arg_{u \in V \setminus S} \max \text{MARGINAL\_GAIN}(G, \mathcal{C}, S, u)$
 8:     $S \leftarrow S \cup \{v\}$
 9:     $\mathcal{RS}(v) \leftarrow 0$
10:     for $w \in N^+(v)$ do
11:         $\mathcal{RS}(w) \leftarrow (1 - p_{vw})\mathcal{RS}(w)$
12:     end for
13: end for
14: return S
```

Theorem 4. *(Time complexity) The worst-case time complexity of the IMPC algorithm (as shown in Algorithm 3) is $O(n + m + k(m + k_{max}))$, where $k = |S|$ is the number of seed nodes to be found, $m = |E|$ is the number of edges of the network, $k_{max} = \max_{v \in V} |N^+(v)|$ is the maximum node degree.*

Proof. As shown in Algorithm 3, initializing the $\mathcal{RS}$ and $\mathcal{PT}$ arrays requires $O(n)$ and $O(m)$ time respectively. In line 7, computing the marginal gain for any given S and v using subroutine 2 needs $O(|N^+(v)|)$ time. In the worst case, we have to traverse all the candidates to find the next seed, so the worst-case time complexity of finding the next seed is $O(\sum_{u \in V \setminus S} |N^+(u)|) \approx O(m)$. After a new seed node, suppose v, is found, we further need $O(|N^+(v)|) \leq O(k_{max})$ time to update the $\mathcal{RS}$ array. So the worst-case time complexity of finding one seed node is $O(m + k_{max})$. Since the algorithm iteratively finds k seed nodes, the overall worst-case time complexity of the IMPC algorithm is $O(n + m + k(m + k_{max}))$, which is essentially $O(km)$ when $m \gg \max\{k_{max}, n\}$. □

Theorem 5. *(Space complexity) If we use adjacency list to represent the network, then the space complexity of the IMPC algorithm will be $O(m + n)$, where n is the number of nodes, m is the number of edges.*

Proof. To store the network with adjacency list, we need $O(m)$ space. To store the $\mathcal{PT}$ and $\mathcal{RS}$ arrays, we need $O(n)$ space. Since the IMPC algorithm requires no extra space, the overall space complexity is $O(m + n)$. □

From the complexity analysis we see that the IMPC algorithm runs in linear time and requires almost no extra memory except for storing the network graph, making itself very efficient in handling large-scale networks.

5. Evaluation

5.1. Datasets

We evaluate the performance of our multi-neighbor potential based influence maximization algorithm on eight real-world datasets with different sizes and categories, including scholar cooperation networks, customer review networks, products co-selling networks, etc. We give a brief introduction about these datasets as follows:

NetHEPT: NetHEPT [12] is a scholar cooperation dataset from the arXiv electronic preprint platform[2]. The network consists of 15K nodes and 31K edges, where nodes represent the scholars in the High Energy Physics Theory area while edges represent the co-authorships among different scholars.

[2] http://www.arxiv.org/

NetPHY: Similar to NetHEPT, NetPHY is another scholar cooperation dataset from the arXiv platform consisting of the co-authorships among researchers in the field of Physics. The dataset has 37K authors and 174K edges.

Epinions: Epinions [60] is a who-trust-whom social network from the customer reviewer site Epinions.com. If a customer trust the review of another one, an edge will be established between them. The network consists of 76K vertices and 406K edges, after all repeated edges were combined.

Amazon: The Amazon dataset was collected by Yang et al. [61] from Amazon, an online shopping site in America. If a product is frequently co-purchased by customers with another product, then an undirected edge is established between them. The network has 335K vertices and 926K links.

DBLP: DBLP is another dataset collected by Yang el. [61] from the DBLP platform and provides co-authorship among scholars in the field of computer science. The network has 317K vertices and 1M edges where each edge indicates that the corresponding two scholars (vertices) have coauthored at least one paper.

Pokec: The Pokec dataset represents the popular on-line social network in Slovakia. It has been established for more than 10 years and connects more than 1.6 million people. The dataset contains the whole social network among Slovakia users, including about 1.6M nodes and 22.3M connections, provided by Takac et al. [62].

LiveJournal: LiveJournal is a free online social platform where users can keep a blog, journal or diary. The users can also declare friendships with each other, forming the edges of the network. The dataset was collected by Yang et al. [61] and has about 4M nodes and 35M edges.

Orkut: Orkut is another free online social platform which was previously operated by Google. It aims to help users maintain existing relations with old friends and establish new relations with new friends. This network, which consists of 3.1M users and 117M relations, was collected by Yang et al. [61] and is the largest one in our experiments.

Most of our datasets are obtained form the Stanford large network dataset collection platform (SNAP)[3]. For simplicity, we treat all the network graphs as undirected ones. The statistical properties of these datasets are summarized in Table 1, together with the parameter values (α, β) of our IMPC algorithm. We experiment with different parameter values, and those with good performance in influence spread are chosen and shown in Table 1.

Table 1: Summary of eight real world datasets

Dataset	NetHEPT	NetPHY	Epinions	Amazon	DBLP	Pokec	LiveJournal	Orkut
# Ndoes	15K	37K	76K	335K	317K	1.6M	4M	3.1M
# Edges	31K	174K	406K	926K	1M	22.3M	35M	117M
Max. Degree	64	178	3,044	290	343	14,854	14,724	33,313
Avg. Degree	4.12	9.38	10.69	4.34	6.62	27.22	17.01	76.17
# Communities	2,262	8,034	10,307	12,326	11,933	2,520	116,288	5,118
Max. Com. Size	661	88	11,220	1,762	13,913	209,159	299,915	764,325
Min. Com. Size	2	1	1	1	1	1	1	1
Avg. Com. Size	6.73	4.62	7.36	27.17	26.57	647.94	34.48	600.32
Modularity	0.836	0.808	0.727	0.84	0.763	0.771	0.724	0.856
Parameter α	0.5	0.2	0.5	1	1	5	1	2
Parameter β	0.5	0.2	0.5	0.2	0.2	0.5	0.2	0.2
Com. Detect. Time (s)	0.09	0.17	0.27	2.1	2.7	40	133	338

5.2. *Baseline algorithms*

We compare the performance of our IMPC algorithm with six baseline algorithms, which cover the main strategies of current studies, including the community based algorithms, node centrality based algorithms, influence path based algorithms, as well as reverse influence sampling (RIS) based algorithms. The parameters of the IMPC algorithm are also included in Table 1. The simulation-based algorithms (e.g., CELF) are not included due to their extremely low time efficiency in handling large-scale networks.

- **CoFIM**: The CoFIM (Community based Framework for Influence Maximization) is a community based influence maximization framework proposed by Shang et al. [15], which assumes that influence first propagates

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

from seed nodes to their neighbors and then from these neighbors to other nodes within the same community. Under the weighted cascade model the authors derived a simple evaluation form of the influence objective function which is submodular and can be efficiently computed.

- **IPA**: IPA [11] (Independent Influence Path Algorithm) is a path based influence maximization algorithm mainly designed for IC model. It approximates the influence function through independent influence paths whose propagation probability is higher than a threshold parameter θ. The propagation probability of an influence path is defined as the product of propagation probabilities along that path. As the authors did in their paper, we set $\theta = 1/320$.

- **TIM+**: TIM+ [26] (Two-phase Influence Maximization) is a reverse influence sampling based algorithm which generalizes the work of Borgs et al. [25] to support both IC and LT models. The method consists of two phase where the first phases estimates an approximation parameter θ while the second phase generate θ samples to derive to best seed set. TIM+ runs in near-linear time and can return a $(1-1/e-\varepsilon)$-approximation solution.

- **IMM**: IMM [27] (Influence Maximization via Martingales) is an extension of TIM+, which provides the same $(1 - 1/e - \varepsilon)$-approximation solution while significantly improves time efficiency by utilizing martingales. Compared to TIM+'s two-phase approach, IMM adds an extra intermediate step to heuristically refine θ into a tighter lower bound. Both TIM+ and IMM have a parameter ε balancing accuracy and efficiency. In this paper we set $\epsilon = 0.1$ as the authors recommended in their papers.

- **SD**: The SD [12] (Single Discount) algorithm is a node centrality-based method which selects seed nodes according to node degree. Specifically, it iteratively selects a new seed node with the maximum degree. When a node is chosen, the degree of each of its neighbor will decreased (discounted) by a unit, in order to avoid influence overlap.

- **Degree**: The Degree algorithm is also a node centrality based method which simply selects top k seed nodes with the maximum degree values.

5.3. Evaluation metrics

To test the effectiveness and efficiency of our proposed approach, we utilize three measures, which were widely used in many previous studies [10, 11, 15].

Influence spread: influence spread is defined as the expected number of overall individuals which can be successfully activated through influence propagation. It is a widely used metric to evaluate the accuracy and effectiveness of an influence maximization algorithm. Since we have previously shown that calculating the exact value of influence spread is #P-hard, an alternative way is through Monte-Carlo simulations. In this article we calculate the influence spread by averaging over 10,000 MC simulations.

Running time & Memory usage: Running time and memory usage are widely used measures to evaluate the time and space efficiency of an influence maximization algorithm. In this article we focus on the overall time and memory consumptions of different algorithms in mining $k = 50$ seed nodes and use them to as time and space measures.

5.4. Experimental procedure

Community detection: Since our community based algorithm relies on the community, to obtain the community structure of real-world networks, we apply the Louvain [41] algorithm which has been proved to have good performance in community detection [46]. The Louvain algorithm is based on *modularity* optimization, where *modularity* is a metric to evaluate the quality of a community structure. It is proposed by Newman et al. [63], and usually denoted by Q, which varies between -1 and 1. Intuitively, a high Q value indicates more connections within communities and less connections among communities. Table 1 also exports the community information and running time of Louvain algorithm on the eight real world datasets.

Diffusion model: The diffusion model we use in this paper is the WC (weighted cascade) model, i.e., the independent cascade model with propagation probability from u to v defined by $p_{uv} = 1/|N^{-}(v)|$, where $N^{-}(v)$ is the set of incoming neighbors of v.

Experimental environment: Our experiments are conducted on a workstation with Intel Xeon E3 3.5GHz CPU and 32GB memory, running the Ubuntu Linux operating system. Our codes [4] are programmed in C/C++ and the programs are in single process and single thread. The source codes of all the baseline algorithms provided by other authors are also written in C/C++.

6. Results

6.1. Overall results of different algorithms

6.1.1. Influence spread

We first compare the influence spread of different algorithms on the eight real world datasets, as shown in Figure 1, where x-axis represents the number of seed nodes to be found while y-axis represents the overall influence spread computed through Monte-Carlo simulations. From the results on the eight real-world datasets, we see that our IMPC algorithm is always among the ones (together with CoFIM, TIM+ and IMM) providing the highest influence spread values. On several datasets (NetHEPT, Epinions, and Orkut) the IMPC algorithm exhibits the best overall performance in terms of influence spread, as shown in Figure 1(a),(c),(h). The IPA algorithm shows the worst performance on all the networks except for the Epinions dataset, as shown in Figure 1(c). The node centrality based algorithms (SD and Degree), though performs well on some datasets, cannot provide any performance guarantee. For example, on two datasets: NetHEPY and NetPHY, their influence spread values, as shown in Figure 1(a),(b), are significantly inferior to that of the IMPC algorithm. On the largest dataset (Orkut), we can also observe a significant gap in influence spread between the node centrality based algorithms and the IMPC algorithm. The difference between node centrality based algorithms and our IMPC algorithm indicates that considering multi-neighbor potential and taking advantage of community structure can significantly improve the accuracy of influence maximization algorithms.

Since the IMM algorithm is an extension of the TIM+ algorithm, both of which are RIS based algorithms providing the same influence guarantee, they exhibit the same influence spread values. Although the influence spread values of TIM+ and IMM algorithms are slightly above that of our IMPC algorithm's with $k = 50$, our algorithm shows its robustness across different values of k. When k is small (e.g., $k \leq 20$), on most datasets (except NetHEPT and DBLP), we see that the influence spread values of TIM+ and IMM algorithms are much lower than our IMPC algorithm. For example, on LiveJournal dataset with $k = 10$, the IMPC algorithm achieves a influence spread value of 45,514, while the values of TIM+ and IMM algorithms are 26,598 and 27,024 respectively, only about half of ours as shown in Figure 1(h).

The CoFIM algorithm shows competitive performance as compared with our IMPC algorithm on some datasets. However, this algorithm relies specifically on the WC (Weighted Cascade) diffusion model, while the IMPC method is free of the diffusion parameter settings, making our method more general in real world applications.

Overall, from the results on the eight real-world networks, our IMPC algorithm shows its effectiveness and robustness in finding top influential seed nodes as compared with the state-of-the-art algorithms.

6.1.2. Running time and memory usage

Running time is an important metric to evaluate the efficiency of algorithms in handling large-scale networks. Table 2 shows the running time of different algorithms on the eight real-world datasets. Here we use the time for selecting $k = 50$ seed nodes to evaluate the time efficiency. Since the Degree and SD algorithms directly and naively select seed nodes based on node degree, the comparison of their running time will make little sense, so we eliminate them from the table.

From the results we see that our IMPC algorithm is much more efficient than all the baseline methods, while the TIM+ algorithm exhibits the worst time efficiency. On all of the eight datasets, including the largest one with more than one million nodes and one hundred million edges, the IMPC algorithm requires less than 15 seconds to find the seed nodes. As compared to the fastest benchmark, i.e., the CoFIM algorithm, our algorithm is up to one order of magnitude faster, as shown on the Epinions dataset. On the other datasets (except Amazon), our algorithm is more than two times faster than the CoFIM algorithm. The most significant improvement of our algorithm over the baseline methods is shown on the Epinions dataset, where the IMPC algorithm is about 10, 400, 500, and 80 times faster than the CoFIM, IPA, TIM+, IMM algorithms, respectively. The running time of our algorithm

[4] The source code of our IMPC algorithm is available on: https://github.com/sjx19871109/IMPC

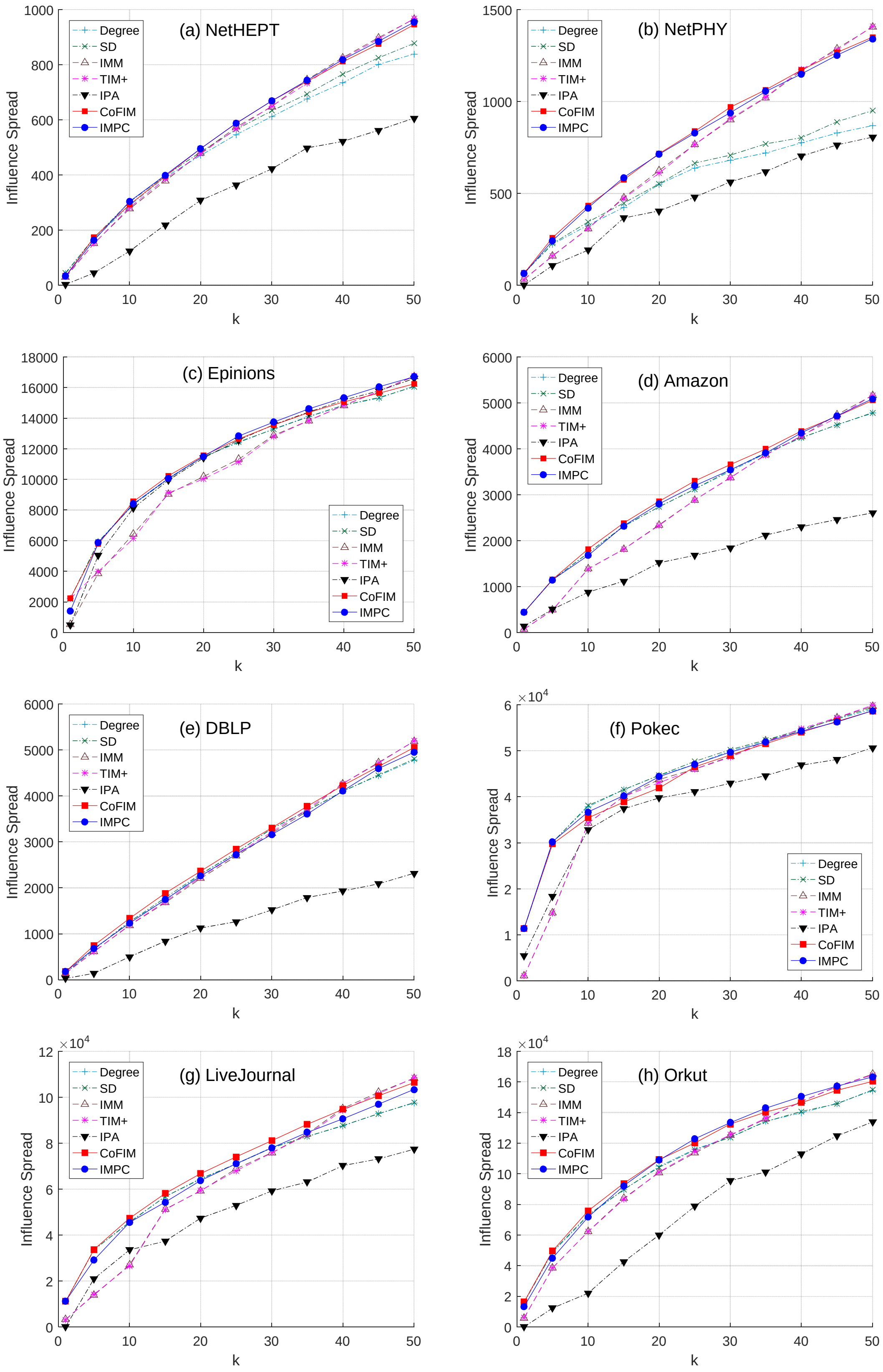

Figure 1: Influence spread on eight real-world networks

on the largest dataset, i.e., Orkut, is slightly over ten seconds, only about 1/6, 1/50, 1/160, and 1/25 to that of the CoFIM, IPA, TIM+ and IMM algorithms. From Table 2 we notice an interesting phenomenon that the IPA algorithm performs better on the Orkut dataset than on the LiveJournal dataset. This is mainly because IPA takes a "node-centric" strategy to build the influence paths, and LiveJournal has more nodes (4 million) than Orkut (3.1 million). In sum, the high time efficiency of our IMPC algorithm makes it possible to handle the mega-scale networks.

Table 2: Running time of different algorithms on eight real world datasets ($k = 50$)

Run Time(s)	NetHEPT	NetPHY	Epinions	Amazon	DBLP	Pokec	LiveJournal	Orkut
IMPC	**0.012**	**0.016**	**0.050**	**0.157**	**0.146**	**2.67**	**4.96**	**12.4**
CoFIM	0.072	0.091	0.65	0.24	0.31	8.75	14.87	73.33
IPA	3	2	20	5	69	430	1154	605
TIM+	3.63	12.36	27.14	71.70	79.24	494.31	484.71	1939.45
IMM	0.85	2.69	4.38	14.81	15.20	84.32	82.72	290.36

The memory usage is another issue concerned by researchers in designing influence maximization algorithms. Table 3 shows the memory usage of different algorithms on the eight real-world datasets. From the results we see that our IMPC algorithm again exhibits its high scalability in dealing with mega-scale networks. Among the baseline methods, the IPA algorithm requires the maximum memory usage because it has to maintain a large amount of influence paths. As we have proved in Theorem 5, the IMPC algorithm requires almost no extra memory except for storing the network, so its space efficiency is much more efficient than other algorithms on all the datasets. For example, on the NetHEPT dataset, the memory usage of our IMPC algorithm is 6M, only about 2.3%, 1.3% and 6.3% to that of the IPA, TIM+ and IMM algorithms respectively. The superiority of our IMPC algorithm is much more obvious on middle-size networks than on mega-scale networks. This is because the time complexity of our algorithm mainly depends on the network size, while the time complexities of the IPA, TIM+ and IMM methods rely on both networks size and the number of seed nodes to be mined, i.e., k. Since k is fixed across different datasets, when the network size becomes very large, the space complexity caused by k will become less significant. Similar to Table 2, IPA algorithm performs better on the Orkut dataset than on the LiveJournal dataset, though the former is a larger network.

Table 3: Memory usage of different algorithms on eight real world datasets ($k = 50$)

Memory	NetHEPT	NetPHY	Epinions	Amazon	DBLP	Pokec	LiveJournal	Orkut
IMPC	**6M**	**13M**	**26M**	**66M**	**70M**	**0.9G**	**1.5G**	**4.2G**
CoFIM	7M	22M	47M	119M	126M	2.2G	3.5G	11G
IPA	263M	284M	1.2G	257M	3.7G	14.2G	31.3G	22.5G
TIM+	450M	861M	318M	3.1G	3.0G	3.9G	6.2G	11G
IMM	95M	187M	84M	712M	642M	1.9G	2.4G	5.5G

In sum, experimental results on eight real-world datastes show that our approach can significantly outperform other state-of-the-art algorithms in terms of time and space efficiency with no compromise on solution accuracy.

6.2. Impact of algorithm parameters

As shown in Eq. 16, the IMPC algorithm depends on two parameters: α and β, which weight the importance of community information in selecting seed nodes. Larger α indicates that greater importance is put on community information. Figure 2 shows how the algorithm parameters affect the influence spread on the eight real-world datasets. The parameter α is set from $\{0, 0.2, 0.5, 1, 2, 5\}$ while β is set from $\{0.2, 0.5, 0.8\}$. Generally, from the results on most of the datasets, we see that with the increase of α (more importance is put on community information), the influence spread value gradually grows to a peak, and then fall back. This indicates that by utilizing the community structure reasonably, we can improve the solution accuracy of the influence maximization algorithm. However, as shown in the results, the parameters should be carefully chosen. Putting too much weight on the community information can even worsen the algorithm performance.

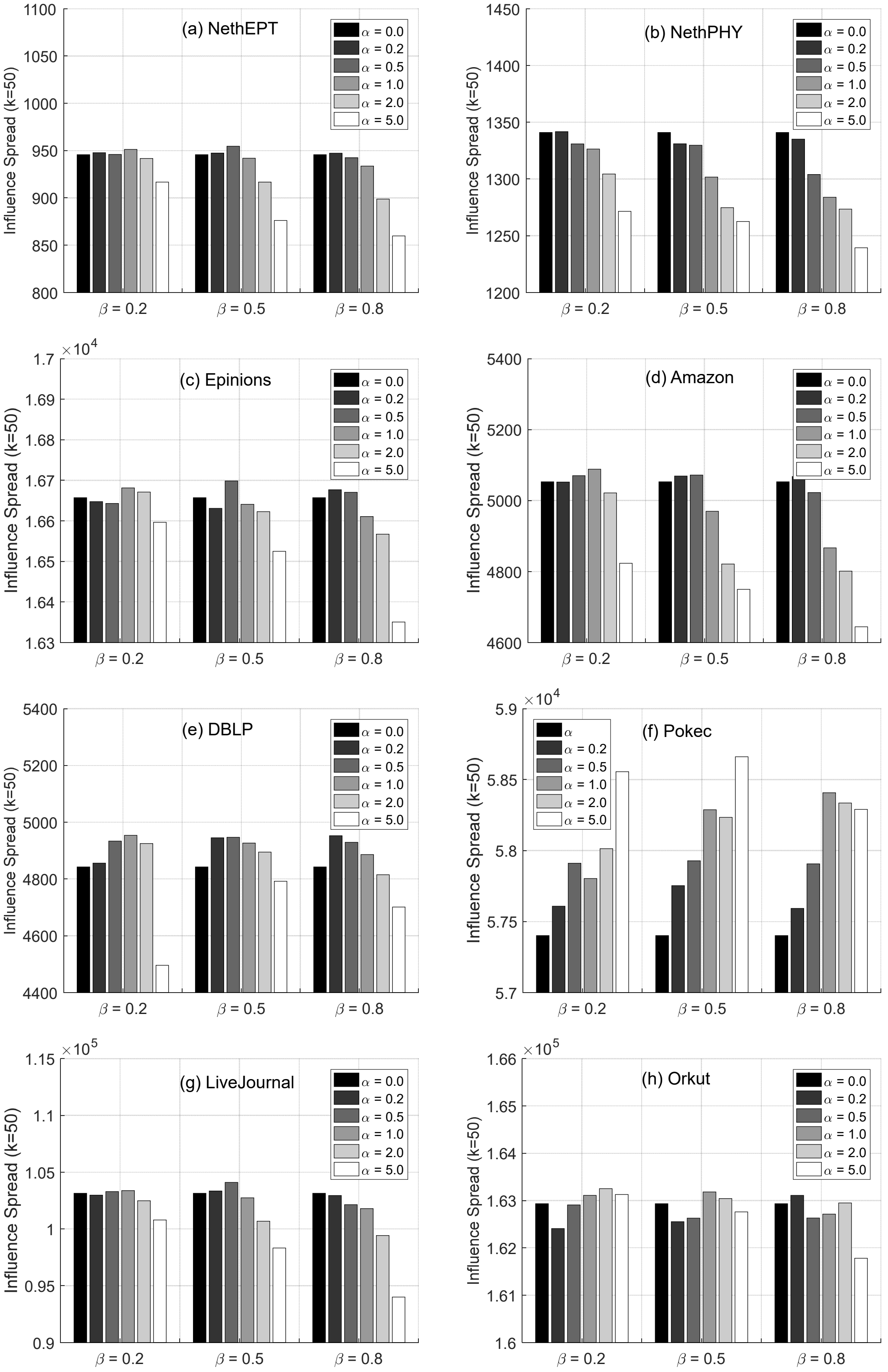

Figure 2: The impact of parameter α and β on influence spread in eight real world datasets

7. Conclusion and future work

In this paper we propose the IMPC framework to solve the influence maximization problem on community networks. In our framework, we take advantage of the connection properties of community structure and divide the influence diffusion process into two phases: (i) multi-neighbor potential based seeds expansion, and (ii) intra-community influence propagation. Based on our model we derive an objective function to evaluate the influence spread as a combination of influence diffusion through the two phases. We theoretically prove that the objective function is submodular and propose an efficient greedy algorithm to find the seed nodes. We also develop several techniques to speed up our algorithm. We conduct extensive experiments on eight real-world datasets to show its superiority over other state-of-the-art methods. Results show that our algorithm can significantly outperform the baseline algorithms in terms of time efficiency with no compromise on influence spread and memory efficiency. We also show how the parameters which weight the importance of community information impact the algorithm performance. Our work provides new solutions to the influence maximization problem on mega-scale networks.

Our research is not immune from its limitations, which can be addressed in future studies. First, our algorithm contains two parameters, which are selected through extensive experiments. In the future, we will improve our framework such that the number of parameters can be reduced and the parameter values can be automatically determined. Second, the IMPC algorithm works on networks with non-overlapping community structure, but real-world networks often show overlapping communities. In the future we will generalize our work to the overlapping community networks. Third, in this study we only consider the IC model, in the future we will extend the IMPC algorithm to other diffusion models, such as the LT model, voter model, etc.

Acknowledgements

This work was supported in part by: National Key Research and Development Program of China (No. 2017YFB1402400), National Natural Science Foundation of China (No. 61702059), Frontier and Application Foundation Research Program of Chongqing City (No. cstc2018jcyjAX0340), Chongqing Industrial Generic Technology Innovation Program (No. cstc2017zdcy-zdzxX0010), China Postdoctoral Science Foundation (No. 2017M612913), Chongqing Postdoctoral Foundation (No. Xm2017125), Guangxi Key Laboratory of Trusted Software (No. kx201702).

References

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

[2] J. Leskovec, L. A. Adamic, B. A. Huberman, The dynamics of viral marketing, ACM Transactions on the Web (TWEB) 1 (1) (2007) 5.

[3] J. J. Brown, P. H. Reingen, Social ties and word-of-mouth referral behavior, Journal of Consumer Research (1987) 350–362.

[4] V. Mahajan, E. Muller, F. M. Bass, New product diffusion models in marketing: A review and directions for research, The Journal of Marketing (1990) 1–26.

[5] D. Kempe, J. Kleinberg, É. 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, ACM, 2003, pp. 137–146.

[6] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An analysis of approximations for maximizing submodular set functionsi, Mathematical Programming 14 (1) (1978) 265–294.

[7] M. L. Fisher, G. L. Nemhauser, L. A. Wolsey, An analysis of approximations for maximizing submodular set functionsii, Polyhedral combinatorics (1978) 73–87.

[8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2007, pp. 420–429.

[9] A. Goyal, W. Lu, 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, ACM, 2011, pp. 47–48.

[10] W. Chen, C. Wang, 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, ACM, 2010, pp. 1029–1038.

[11] J. Kim, S.-K. Kim, H. Yu, Scalable and parallelizable processing of influence maximization for large-scale social networks, in: Data Engineering (ICDE), 2013 IEEE 29th International Conference on, IEEE, 2013, pp. 266–277.

[12] 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, ACM, 2009, pp. 199–208.

[13] H.-L. Liu, C. Ma, B.-B. Xiang, M. Tang, H.-F. Zhang, Identifying multiple influential spreaders based on generalized closeness centrality, Physica A: Statistical Mechanics and its Applications 492 (2018) 2237–2248.

[14] M. Girvan, M. E. Newman, Community structure in social and biological networks, Proceedings of the national academy of sciences 99 (12) (2002) 7821–7826.

[15] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, Cofim: A community-based framework for influence maximization on large-scale networks, Knowledge-Based Systems 117 (2017) 88–100.

[16] C. Zhou, P. Zhang, W. Zang, L. Guo, On the upper bounds of spread for greedy algorithms in social network influence maximization, IEEE Transactions on Knowledge and Data Engineering 27 (10) (2015) 2770–2783.

[17] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, J. X. Yu, Influence maximization over large-scale social networks: A bounded linear approach, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 171–180.

[18] Q. Ma, J. Ma, Identifying and ranking influential spreaders in complex networks with consideration of spreading probability, Physica A: Statistical Mechanics and its Applications 465 (2017) 312–330.

[19] J. Zhu, Y. Liu, X. Yin, A new structure-hole-based algorithm for influence maximization in large online social networks, IEEE Access 5 (2017) 23405–23412.

[20] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, M. Serna, Centrality measure in social networks based on linear threshold model, Knowledge-Based Systems 140 (2018) 92–102.

[21] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE, 2010, pp. 88–97.

[22] B. Liu, G. Cong, Y. Zeng, D. Xu, Y. M. Chee, Influence spreading path and its application to the time constrained social influence maximization problem and beyond, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2014) 1904–1917.

[23] Y.-Y. Ko, D.-K. Chae, S.-W. Kim, Accurate path-based methods for influence maximization in social networks, in: Proceedings of the 25th International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 59–60.

[24] Y.-Y. Ko, D.-K. Chae, S.-W. Kim, Influence maximisation in social networks: A target-oriented estimation, Journal of Information Science (2018) 0165551517748289.

[25] 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, Society for Industrial and Applied Mathematics, 2014, pp. 946–957.

[26] Y. Tang, X. Xiao, Y. Shi, Influence maximization: Near-optimal time complexity meets practical efficiency, in: Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 2014, pp. 75–86.

[27] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: A martingale approach, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1539–1554.

[28] H. T. Nguyen, M. T. Thai, T. N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, in: Proceedings of the 2016 International Conference on Management of Data, ACM, 2016, pp. 695–710.

[29] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, L. V. Lakshmanan, Revisiting the stop-and-stare algorithms for influence maximization, Proceedings of the VLDB Endowment 10 (9) (2017) 913–924.

[30] X. Wang, Y. Zhang, W. Zhang, X. Lin, C. Chen, Bring order into the samples: A novel scalable method for influence maximization, IEEE Transactions on Knowledge and Data Engineering 29 (2) (2017) 243–256.

[31] T. Cao, X. Wu, S. Wang, X. Hu, Oasnet: an optimal allocation approach to influence maximization in modular social networks, in: ACM Symposium on Applied Computing, ACM, 2010, pp. 1088–1094.

[32] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowledge-Based Systems 42 (2013) 74–84.

[33] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, S.-Y. Lee, Cim: community-based influence maximization in social networks, ACM Transactions on Intelligent Systems and Technology (TIST) 5 (2) (2014) 25.

[34] H. Li, S. S. Bhowmick, A. Sun, J. Cui, Conformity-aware influence maximization in online social networks, The VLDB Journal 24 (1) (2015) 117–141.

[35] J. Shang, H. Wu, S. Zhou, L. Liu, H. Tang, Effective influence maximization based on the combination of multiple selectors, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2017, pp. 572–583.

[36] H. Wu, J. Shang, S. Zhou, Y. Feng, A linear time algorithm for influence maximization in large-scale social networks, in: International Conference on Neural Information Processing, Springer, 2017, pp. 752–761.

[37] K. Zhang, H. Du, M. W. Feldman, Maximizing influence in a social network: Improved results using a genetic algorithm, Physica A: Statistical Mechanics and its Applications 478 (2017) 20–30.

[38] X. Wang, Y. Zhang, W. Zhang, X. Lin, Efficient distance-aware influence maximization in geo-social networks, IEEE transactions on knowledge and data engineering 29 (3) (2017) 599–612.

[39] X. Li, X. Cheng, S. Su, C. Sun, Community-based seeds selection algorithm for location aware influence maximization, Neurocomputing 275 (2018) 1601–1613.

[40] C. Gou, H. Shen, P. Du, D. Wu, Y. Liu, X. Cheng, Learning sequential features for cascade outbreak prediction, Knowledge and Information Systems (2018) 1–19.

[41] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.

[42] T. Chen, P. Singh, K. E. Bassler, Network community detection using modularity density measures, Journal of Statistical Mechanics: Theory and Experiment 2018 (5) (2018) 053406.

[43] X.-K. Zhang, J. Ren, C. Song, J. Jia, Q. Zhang, Label propagation algorithm for community detection based on node importance and label influence, Physics Letters A 381 (33) (2017) 2691–2698.

[44] L. Gulikers, M. Lelarge, L. Massoulié, A spectral method for community detection in moderately sparse degree-corrected stochastic block models, Advances in Applied Probability 49 (3) (2017) 686–721.

[45] J. Shang, L. Liu, F. Xie, Z. Chen, J. Miao, X. Fang, C. Wu, A real-time detecting algorithm for tracking community structure of dynamic networks, in: ACM 6th SNA-KDD Workshop, ACM, 2012.

[46] J. Shang, L. Liu, X. Li, F. Xie, C. Wu, Targeted revision: A learning-based approach for incremental community detection in dynamic networks, Physica A: Statistical Mechanics and its Applications 443 (2016) 70–85.

[47] X. Wen, W.-N. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, J. Zhang, A maximal clique based multiobjective evolutionary algorithm for overlapping community detection, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 363–377.

[48] H. Sun, J. Liu, J. Huang, G. Wang, X. Jia, Q. Song, Linklpa: A link-based label propagation algorithm for overlapping community detection in networks, Computational Intelligence 33 (2) (2017) 308–331.

[49] W. Li, J. Xie, M. Xin, J. Mo, An overlapping network community partition algorithm based on semi-supervised matrix factorization and random walk, Expert Systems with Applications 91 (2018) 277–285.

[50] X. Cao, X. Wang, D. Jin, X. Guo, X. Tang, A stochastic model for detecting overlapping and hierarchical community structure, PloS one 10 (3) (2015) e0119171.

[51] Y. Chen, X. Wang, X. Xiang, B. Tang, Q. Chen, S. Fan, J. Bu, Overlapping community detection in weighted networks via a bayesian approach, Physica A: Statistical Mechanics and its Applications 468 (2017) 790–801.

[52] J. Shang, L. Liu, F. Xie, C. Wu, How overlapping community structure affects epidemic spreading in complex networks, in: IEEE 38th International Computer Software and Applications Conference Workshops (COMPSACW), IEEE, 2014, pp. 240–245.

[53] J. Shang, L. Liu, X. Li, F. Xie, C. Wu, Epidemic spreading on complex networks with overlapping and non-overlapping community structure, Physica A: Statistical Mechanics and its Applications 419 (2015) 171–182.

[54] L. Guo, J.-H. Lin, Q. Guo, J.-G. Liu, Identifying multiple influential spreaders in term of the distance-based coloring, Physics Letters A 380 (7-8) (2016) 837–842.

[55] Z.-K. Bao, J.-G. Liu, H.-F. Zhang, Identifying multiple influential spreaders by a heuristic clustering algorithm, Physics Letters A 381 (11) (2017) 976–983.

[56] W. Wang, M. Cai, M. Zheng, Social contagions on correlated multiplex networks, Physica A: Statistical Mechanics and its Applications.

[57] W. Wang, X.-L. Chen, L.-F. Zhong, Social contagions with heterogeneous credibility, Physica A: Statistical Mechanics and its Applications 503 (2018) 604–610.

[58] W. Wang, H. E. Stanley, L. A. Braunstein, Effects of time-delays in the dynamics of social contagions, New Journal of Physics 20 (1) (2018) 013034.

[59] Y. Wang, X. Feng, A potential-based node selection strategy for influence maximization in a social network, in: International Conference on Advanced Data Mining and Applications, Springer, 2009, pp. 350–361.

[60] M. Richardson, R. Agrawal, P. Domingos, Trust management for the semantic web, in: International semantic Web conference, Springer, 2003, pp. 351–368.

[61] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowledge and Information Systems 42 (1) (2015) 181–213.

[62] L. Takac, M. Zabovsky, Data analysis in public social networks, in: International Scientific Conference and International Workshop Present Day Trends of Innovations, 2012, pp. 1–6.

[63] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2) (2004) 026113.