Diffusing information for mobile social networks under consideration of dynamic influence

Dong Jing and Ting Liu

Abstract

As the developments of new techniques, mobile social networks have been built wildly. To obtain and spread information over mobile social networks efficiently, the influence maximization problem is to find a seed nodes set with limited size such that it can influence as many nodes as possible. Previous works ignore the dynamic influence phenomenon of diffusing information on mobile social networks. In this article, we propose a new model to express the procedure of diffusing information under the existence of dynamic influence. Theoretical analysis shows that the influence maximization problem under new model is non-deterministic polynomial-time hard, and efficient approximation algorithm is proposed. Experimental studies on real data sets show that the new model can process dynamic influence well in the diffusing information procedure, and the proposed algorithms can solve the influence maximization problem on new model efficiently.

Keywords

Mobile social networks, information diffusion, dynamic influence

Date received: 19 December 2016; accepted: 10 March 2017

Academic Editor: Zhipeng Cai

Introduction

Recently, as the developments of techniques of communications and computing, modern smart phones have huge increase and rapid popularization in the whole world. According to the reports, until 2015, the size of mobile phone users in the world has reached 4.45 billion and 42.9% of them are using smart phones (about 2 billion). By the reports from China, the size of mobile smart phone users has reached 5 billion in 2014. In addition, as the emergence of more devices, such as tablet and smart watch, more and more smart mobile devices are popularized now. Because of the developments of embedded computing, sensing, and communicating techniques, the smart mobile devices have more and more powerful abilities and they are not only tools for communications any more. In many applications, they have performed to be powerful tools for mobile sensing, computing, and so on. For example, iPhone 6 has integrated at least eight types of sensors such as ALS (ambient light sensor), PS (proximity sensor), and GPS (Global Positioning System), and it takes a 2.6 GHz 64-bit processor, besides common modules for telecommunication, and it also has powerful WiFi and bluetooth devices.

As the size and power of smart mobile devices increase, more and more online social applications are being used in mobile environments, and the traditional social networks have evolved to be mobile social networks. The typical applications of mobile social networks include Facebook, Twitter, Google+, and Sina Weibo. According to the reports, until 2014, the...
number of month average users (MAU for short) of Facebook is about 1.3 billion, and in 2015, the MAU size for Sina Weibo has reached 212 million only in China. As more and more mobile applications are utilized, more and more natural mobile social networks are being created.

The emergence of mobile social networks changes the way of information diffusion and provides opportunities for viral marketing as shown in the work by Chen et al. Different from traditional methods for marketing, viral marketing can utilize the “word-of-mouth” advantages of mobile social networks and diffuse advertising information more efficiently. It has attracted lots of research interests from both mobile and social computing areas. Influence maximization problem is one of the most popular topics in the area of mobile social network. It has been formally investigated by Kempe et al. However, there are still important challenges not solved in real applications of influence maximization problem when facing more complex scenarios. One of them is that the influence ability between nodes in real world may usually dynamically change, which is also the motivation of this article, while most of current research efforts on influence maximization problem always assume that the influence ability is static.

For general social networks, they are at least composed of nodes and edges like general networks. Usually, each node represents one social actor (e.g. one person in physical world and one user of Facebook) and the edges represent the social interactions between social actors (e.g. two persons are friends and one user is following the other one on Facebook). When considering the influence maximization problem on mobile social networks, usually, the edges have special meanings related to influence (e.g. one user accepts the advice of another one). The most popular methods proposed by previous works are as follows. First, a model is built to describe the information diffusion procedure on mobile social networks; then, the algorithms for finding special seed set with maximized influence are designed. In classic models, whether user A will be influenced by user B depends on the influence probability between them. In detail, if B has accepted some advice and the influence probability between B and A is 0.9, A will accept the same advice with probability 0.9 in next step. The influence probability is a simple and clear method to describe the ability of influence between two nodes, and it is usually assumed to be a constant once if a special social network is considered. However, in mobile social networks, the influence probability may be dynamically changed.

Let us consider an example in real life to show the dynamic influence phenomenon during the process of diffusing information which has been also observed by only few previous works. Assume that there is one user A of Twitter, which can be treated to be a directed graph, before lunch A is browsing his following twitters. Using GPS or WiFi devices of A’s smart phone, the app of Twitter may know the location of A and push one advertisement about restaurant X nearby. After seeing that, A may have his lunch in X and post twitters to recommend X to his friends. Assume A has two friends B and C, B is also near from X but C is not. Then, B is more likely than C to view the advertisement and go for a taste, while A may have the same influence probabilities for both B and C on general information. Let us imagine and compare two similar scenarios. In the first case, A receives one posted tweet from his friend B; here, B is the original author of that tweet. In the second case, A still receives the tweet from B, and the difference is that B reposted that from C. Obviously, the influence probability from B to A in the first case should be higher than the one in the second case. In the two examples above, the influence probability is not static for two given nodes, and it can change during the procedure of information diffusion. As known by us, there are only few previous works considering such challenges and none of them considers the influence maximization problem under dynamic influence directly.

Actually, the examples above show two important sources of dynamic influence, locations and time delay. In a mobile social network, if two nodes have same or similar locations, the influence probability between them tends to become larger. If one information has long time delay or has been reposted several times by users sequentially, after receiving such information, the influence probability between two users tends to become smaller. Here, the location need not be a geographic position, and it can be any profile information of nodes which can enhance the social interactions between nodes. For example, when the nodes represent academic authors, the affiliation or research area information can be the “location.” The time need not be a real time either, and it can be any value about the information which can affect the interests for the information of nodes. For example, we can use the times that one user has received the same information to measure the “time.” Zhou et al. considered only the “location” factor but not the “time” factor. Only few works focus on dynamic influence caused by those factors. Goyal et al. and Leskovec et al. have observed the dynamic influence cases caused by “time” factor. Leskovec et al. summarized a series of interesting scenarios including the dynamic influence in social networks by experiments on real data sets. Goyal et al. studied how to learn such dynamic influence efficiently. Neither of them directly focuses on the influence maximization problems under dynamic influence. Moreover, when
considering dynamic influence, the algorithms designed for classical influence maximization problem are not able to be applied. This article studies the influence maximization problem under dynamic influence caused by the location and time factors.

In this article, we address the problem of diffusing information under dynamic influence in mobile social networks. Following the methods of previous works, we also use influence maximization problem to describe the procedure of diffusing information over mobile social networks. To overcome the challenge of dynamic influence, we modify the classic models to support describing the change in influence under considerations of location and time factors. To solve the influence maximization problem efficiently, we study its computational complexities and design efficient algorithms. The main contributions can be summarized as follows:

1. We identify the dynamic influence as new challenges of information diffusion on mobile social networks. To overcome them, we propose new information diffusion (ID) models to support dynamic influence and formulate the new influence maximization problem based on new model.
2. We show the hardness for influence maximization problem under dynamic influence. It is achieved by proving that classic influence maximization problems is a special case of the new problem.
3. We design efficient approximation algorithms for the new influence maximization problem. By showing the monotone and submodular properties of the new problem, a $(1 - 1/e)$ approximation algorithm can be obtained.
4. More efficient approximation algorithms for the new influence maximization problem are designed. Also, several optimizing strategies are discussed.
5. The experimental results on real data sets show that the proposed method can solve the information diffusion problem with dynamic influence on mobile social networks.

The rest parts of the article are organized as follows. In section “Related work,” the related works are discussed. Then, some preliminaries and new definitions will be introduced in section “Notations and definitions.” In section “Approximation algorithm for IMD problem,” theoretical analysis and approximation algorithms for influence maximization problems are introduced. Also, an improved algorithm is given in section “Improving Algorithm GREEDYIMD.” Experimental results are shown in section “Experimental evaluation.” Finally, it is the conclusion.

**Related work**

The influence maximization is an important problem in the research area of online social networking, which has many applications such as viral marketing and computational advertising. It is first studied by Domingos and Richardson,\textsuperscript{9,10} and the formalized definitions and comprehensive theoretical analysis are given by Kempe et al.\textsuperscript{2} The standard formal definition of influence maximization can be explained as follows: given the constraint that at most $k$ nodes can be selected, the input is a graph which represents the “influence” relationships between nodes, the problem is to compute a set of $k$ nodes such that the number of nodes influenced by the $k$ nodes is maximum. Different models have been formally defined to simulate the information propagation processes with different characteristics, and the two most popular models are the independent cascade (IC for short) and linear threshold (LT for short) models. In the work by Kempe et al.,\textsuperscript{2} the influence maximization problems under both IC and LT models are shown to be NP-hard (non-deterministic polynomial-time hard) problems, and the problem of computing the exact influence of given nodes set is shown to be $\# P$-hard problem in the work by Chen et al.\textsuperscript{4}

Many research efforts have been made for the problem of finding the node set with maximum influence. Kempe proposed an algorithm for influence maximization based on greedy ideas which has constant approximation ratio $(1 - 1/e)$. The time complexity of the greedy approximation algorithm of influence maximization is $O(n^2(m + n))$, which is too high to be applied in large-scale social networks. To overcome the shortcomings of greedy-based algorithms, many researchers focus on the problem of influence maximization. By studying the submodularity characteristics of influence functions, Leskovec et al.\textsuperscript{11} proposed CELF (cost-effective lazy-forward) algorithm. CELF can improve the performance of greedy-based algorithms for influence maximization by reducing the times of evaluations of influence set of given seed set; however, its performance on large-scale data is still not satisfying. Using the similar ideas, CELF++ is proposed to improve the performance of algorithms for solving influence maximization by Goyal et al.\textsuperscript{12} In the work by Chen et al.,\textsuperscript{3} degree-discount algorithm is proposed to improve the performance of greedy-based influence maximization algorithms. By assuming all influence probabilities are same in IC models, Chen et al.\textsuperscript{3} reduce the complexities of influence maximization problems and give better algorithms based on the new models. Utilizing the structural properties of communities in social networks, Chen et al.\textsuperscript{13} proposed new algorithms by merging similar nodes and reduce the cost of computing influence set. Goyal et al.\textsuperscript{14} proposed SIMPATH algorithm in LT model which improves the performance of greedy-based
influence maximization algorithm in LT model. Jiang et al.\textsuperscript{15} proposed simulated annealing-based influence maximization algorithms.

Kimura and Saito\textsuperscript{16} proposed new models of information propagation based on the idea of finding shortest paths, which assume that the information is mainly transferred through shortest paths, and designed new heuristic algorithms for influence maximization problems. Using this model, Chen et al.\textsuperscript{1} proposed heuristic algorithms based on maximum broadcast paths, which assume that the information propagated on the network is not transferred by shortest path but maximum broadcast paths. Based on the influence probabilities between users, for each single node, an influence tree is built by computing the maximum broadcast paths, which can be used to estimate the influence range of each user. By assigning threshold for each user, the influence tree can be controlled to ignore nodes which contribute little for the computation of influence set and reduce the size of nodes computed by the influence computation. Also, Chen et al.\textsuperscript{1} proved the submodularity of influence functions defined based on maximum broadcast paths and designed approximation algorithms with $1 - 1/e$ approximation ratio. In the work by Han et al.,\textsuperscript{17} timeliness networks with opportunistic selection are investigated and the information maximization model is extended to those applications. In the work by Shi et al.\textsuperscript{18} maximal time bound is considered to limit the abilities of diffusing information in social networks, and efficient algorithms for influence maximization problem for computing maximal time-bounded positive influence set are proposed. In the work by Chen et al.\textsuperscript{13} similarities of nodes of communities in social networks are utilized to reduce the number of nodes involved in the influence computation. Kim et al.\textsuperscript{19} proposed efficient influence maximization algorithms in parallel computing settings. Cai et al.\textsuperscript{20} try to extend the information maximization models to the applications of crowd-sourced data-based social networks. Han et al.\textsuperscript{21} consider the communities in social networks and study the influence maximization problem over such networks.

There are also many works which try to extend the classic influence maximization methods to other application settings. Li et al.\textsuperscript{22} study the problem of influence maximization under location-based social networks. In those networks, one node can be influenced by the other node if and only if they are neighbors according to their location information, and Li et al.\textsuperscript{22} focus on the problem of finding $k$ users which can influence maximum users in the location-based social network. Tang et al.\textsuperscript{23} identify the relation types during propagating the information and formally define the problem of influence maximization by considering different types of relationships between nodes. A key idea is that given certain information which needs to be propagated, the influence set of some node set can be computed more efficiently by reducing those edges belonging to some certain types. Chen et al.\textsuperscript{24} study the problem of influence maximization under topic-aware applications. Cai et al.\textsuperscript{25} use the idea of information diffusion to prevent sensitive information in social networks. More related work on applications in social networks can be found in the works of Han et al.\textsuperscript{26} and Bi et al.\textsuperscript{27} Although the problem of extending classic influence maximization methods has been studied by many research works as shown above, we are not aware of any efforts on influence maximization on dynamic setting.

Obviously, the influence models are usually defined based on several parameters which are utilized to describe the key properties in real applications. The parameter selection problem is essential to the influence maximization methods. Tang et al.\textsuperscript{5} propose topic factor graph (TFG) models to determine different parameters between users and topics. Liu et al.\textsuperscript{4} determine different influence parameters among users using probabilistic models to analyze the relationships of distributions between topics of users and influence relations of users. Weng et al.\textsuperscript{28} utilize the latent Dirichlet allocation (LDA) models to describe the topic distributions of user topics and propose TwitterRank methods to determine the influence probabilities between users and topics.

**Notations and definitions**

In this section, classical information diffusing models are introduced first; then, to integrate the two challenge aspects of diffusing information on mobile social networks, new model is proposed by modifying the classical one. Finally, based on the new ID model, we give the new definition of influence maximization problem. In fact, the problem of influence maximization depends on the definition of diffusing models. For all diffusing models, the related influence maximization problems are based on the same idea, but they are different on the aspect of computational hardness.

**Traditional ID models**

In this article, information diffusion can be described as the propagating procedure of information over some network. A network is usually denoted by a graph $G(V, E)$. Here, $V$ is the node set where each node represents one person or entity and $E$ is the edge set where each edge represents the relation (cooperation, friends, enemies, and so on) between two nodes. Each node is associated with active or inactive state. Intuitively, the active state means that the node has been affected. The active set of nodes may affect the nodes in inactive set, and the influence ratio can describe the strength of that affection. If some inactive node is affected by some
active node so much that the inactive becomes active, such a process is called activation. Intuitively, for some node \(v\), the more of the neighbors of \(v\) are activated, more likely \(v\) will be activated. After that, \(v\) will affect more nodes further. As such procedures repeat, more and more nodes will become active. The procedure of activation cannot be reversed: one node can transform from inactive state to active state, but not vice versa. To design proper theoretical model to describe information diffusion in real world, the key is to explain how the interactions between nodes work. Next, we introduce two popular ID models.

**LT model.** Given a network \(G(V, E)\), let \(\mathcal{N}(v)\) be the set \(\{u|(u,v) \in E\}\). For each \((u,v) \in E\), a threshold value \(b_{uv}\) is utilized to represent the degree of influence from \(u\) to \(v\). For each node \(v\), it is satisfied that \(0 \leq \sum_{u \in \mathcal{N}(v)} b_{uv} \leq 1\). During the procedure of information diffusion, another threshold value \(\theta_v\) with respect to each node \(v\) is used to control the diffusion of information. In detail, at some instant time, let \(A(v)\) be the set of \(v\)'s neighbor nodes which has been active. If \(\sum_{u \in A(v)} b_{uv} \geq \theta_v\), \(v\) will become active. In this model, when node \(u\) tries to activate its neighbor \(v\) and fails, the influence \(b_{uv}\) is remembered and will be accumulated in the following activating steps. In other words, the influence from \(u\) to \(v\) will not be ignored, even if the activation is failed. As we will see in the following part, the influence is treated differently in other models. The whole procedure of information diffusion in LT model can be described as follows. First, an initial active node set \(S_0\) will be activated. Then, in the \(i\)th step of information diffusion, based on the active nodes in \(S_{i-1}\), the influence for each node on \(V \setminus S_{i-1}\) will be computed. According to the influence computed and the \(\theta_v\) for each node \(v\), all nodes satisfying \(\sum_{u \in A(v)} b_{uv} \geq \theta_v\) will be put in \(S_i\). Repeat these steps until no more nodes can become active.

**IC model.** IC model is a probabilistic model. Instead of \(b_{uv}\) in LT model, this model uses \(p_{uv}\) to describe the probability that \(u\) can activate \(v\) in a single activation. The whole procedure of information diffusion under IC model can be described as follows. First, an initial node set \(S_0\) will be set to be active. Then, in the \(i\)th step, every node will try to activate their neighbors. In detail, for each node \(u \in S_{i-1}\) and node \(v \in V \setminus S_{i-1}\), if \((u,v) \in E\), \(v\) will be activated once with probability \(p_{uv}\). If \(v\) indeed becomes active, it will be added to \(S_i\) and not be further considered in current step. Repeat this procedure until that no new nodes are added. It should be noted that \(p_{uv}\) is only determined by \(u\) and \(v\) and is independent with other node pairs. In this model, each edge \((u, v)\) will be considered only one time. Once it fails, this edge will never be considered. In the work by Kempe et al., an extended model in which \(p_{uv}\) will be decreased as time goes by is considered.

**New model for diffusing information**

All previous models for information diffusion do not consider dynamic influence; this part will propose a new model integrating both “location” and “time” factors which are major sources of dynamic influence in mobile social network.

**ID model for dynamic influence.** In ID model, the mobile social network can be represented by a graph \(G = (V, E)\). Here, \(V\) is the node set and \(E\) is the directed edge set which represents the influence relationship between nodes in the network. Intuitively, if there is an edge \((u, v) \in E\), it says that \(v\) can be influenced by \(u\). That is, if \(u\) has been influenced, \(v\) also may be influenced through the edge \((u, v)\). We use a function \(P : E \rightarrow [0, 1]\) to represent the influence probability of each edge which can be used to describe how much influence one node has on another one. For edge \((u, v)\), we will represent its value of \(P\) with \(p_{uv}\). The parameters introduced above are same as the IC model, and to describe the dynamic influence during diffusing information, we need to involve more parameters.

To integrate the dynamic influence caused by “location” factors, we use the function \(L : V \times V \rightarrow [0, 1]\) defined over \(V \times V\) to describe the location relationship between nodes. Intuitively, the more likely that two nodes have the same locations, the more likely that they influence each other and the higher the corresponding value of \(L\) is. In the following, we use \(l_{uv}\) to denote the \(L\) value of node pair \((u, v)\).

To integrate the dynamic influence caused by “time” factors, we use the function \(C : E \rightarrow [0, 1]\) to describe the change of influence caused by “time delay.” In the following, we will use \(c_{uv}\) to denote the \(C\) value on edge \((u, v)\).

Therefore, in ID model, to describe the information diffusion on some mobile social network, we need one four-tuple \(\langle G, P, L, C \rangle\).

In ID model, given the network \(\langle G = (V, E), P, L, C \rangle\) and seed node set \(A\) and a threshold \(0 \leq \theta \leq 1\), the information diffusion process working in discrete time can be explained as follows. Here, we use \(l_0, t_1, \ldots\) to represent the discrete times. Initially, at time \(t_0\), all nodes in \(A\) will become active and inserted into the set \(Z\), and all other nodes will be initialized to be inactive. At time \(t_i\), all active nodes will try to activate their “new” neighbor nodes which are met first time at time \(t_i\). In detail, suppose node \(u\) is active and \(v\) is an inactive new met node of \(u\), that is, \(u\) and \(v\) did not meet before \(t_i\). If \(l_{uv} \geq \theta\), the node \(u\) will try to activate \(v\) in two
steps. First, generate a random value \( x_1 \) between 0 and 1. If \( x_1 \leq l_{uv} \), the node \( v \) will be activated with probability \( c_{uv}^{-1} \). Otherwise, \( v \) will become active with probability \( p_{uv} \cdot c_{uv}^{-1} \). If \( l_{uv} > \theta \), the node \( v \) will become active with probability \( p_{uv} \cdot c_{uv}^{-1} \). It should be noted that each edge \((u, v)\) will be utilized only once, that is, \( u \) has only once chance to activate \( v \). Such procedure iterates until no new nodes can be added into \( Z \). Finally, \( Z \) will be the influenced set of \( A \) under network \( G \). During the whole procedure, it should be noted that the node state can transform from inactive to active, but not vice versa. Moreover, each node can be activated several times but can be activated by each node at most once.

The idea of integrating dynamic influence into information diffusion procedure can be explained as follows. The function \( L \) measures the location similarity between two nodes. If they are similar, \( v \) will become active with probability higher than the original \( p_{uv} \). It can be used to describe the case that one node tends to be influenced by the nodes “nearby.” The function \( C \) measures the decrease in influence caused by time delay. When considering the “time” factor, node \( v \) will try to be activated with probability lower than \( p_{uv} \). In real applications, using this model needs choosing proper values for the parameters used which can be solved by sophisticated learning methods. The problem of choosing values for the parameters has beyond the scope of this article. Furthermore, we have the following observations.

**Observation 1.** Without function \( C \), node \( v \) will be activated by \( u \) with probability \( l_{uv} + (1 - l_{uv})p_{uv} \) which is larger than \( p_{uv} \).

**Observation 2.** Without function \( L \), node \( v \) will be activated by \( u \) with probability \( p_{uv} \cdot c_{uv}^{-1} \) which is smaller than \( p_{uv} \).

Let us consider a real example of ID model shown in Figure 1.

**Example 1.** Assume that we have an ID model \((G, P, L, C)\) for some social network. The network \( G \) is shown in Figure 1(a), which includes five nodes and seven edges. The definition of \( P \) can be represented as follows

\[
P(A, C) = 0.6; \quad P(A, B) = 0.9; \\
P(B, C) = 0.6; \quad P(C, E) = 0.7; \\
P(C, D) = 0.9; \quad P(D, C) = 0.2; \\
P(D, B) = 0.3
\]

The definition of function \( L \) is as follows

\[
L(A, B) = 0.3; \quad L(A, C) = 0.7; \quad L(A, C) = 0.3; \\
L(A, E) = 0.3; \quad L(B, C) = 0.3; \quad L(B, D) = 0.3; \\
L(B, E) = 0.3; \quad L(C, D) = 0.3; \quad L(C, E) = 0.9; \\
L(D, E) = 0.3
\]

The definition of function \( C \) is as follows

\[
C(A, C) = 0.9; \quad C(A, B) = 0.95; \quad C(B, C) = 0.8; \\
C(C, E) = 0.7; \quad C(C, D) = 0.8; \quad C(D, C) = 0.95; \\
C(D, B) = 0.5
\]

Given \( \{A\} \) as the seed node set and the threshold \( \theta = 0.6 \), an example of information diffusion procedure is shown in Figure 2. In the first step \( t_0 \), node \( A \)
will be initialized to be active and no edges will be processed in this step. In the second step $t_2$, two edges connected with $A$ will be processed. For $(A, C)$, since $l_{AC} = L(A, C) = 0.7 > \theta = 0.6$, node $C$ will be tried to be activated in two steps. First, assume that the random value generated is 0.85. Because $0.85 > l_{AC} = 0.7$, $C$ will be activated with probability $p_{AC} \cdot c_{AC}^{-1} = p_{AC} \cdot c_{AC}^{-1} = p_{AC} = 0.6$. For the edge $(A, B)$, because $l_{AB} > \theta$, $B$ will be activated with probability $p_{AB} \cdot c_{AB}^{-1} = p_{AB} = 0.9$. It should be noted that since the diffusing procedure is a probabilistic process, the nodes with high probabilities may still be inactive and the nodes with low probabilities may become active. Let us assume that node $C$ becomes active but node $B$ does not as shown in Figure 2. The following steps shown in Figure 2 can be summarized as follows:

$t_2$: \{(C, D) : p_{CD} \cdot c_{CD} = 0.72\}, \{(C, E) : l_{CE} > \theta, \text{random value} = 0.85 > l_{CE}, c_{CE} = 0.7\}

$t_3$: \{(D, C) : \text{not processed}\}, \{(D, B) : p_{DB} \cdot c_{DB}^2 = 0.075\}

$t_4$: \{(B, C) : \text{not processed}\}

$t_5$: \{no edges left\}

In steps $t_3$ and $t_4$, the edges $(D, C)$ and $(B, C)$ are not processed because $C$ has become active in $t_1$. In $t_5$, all edges have been processed and the procedure of diffusing information is finished. Finally, the nodes $\{A, B, C, D, E\}$ are influenced by the seed set $\{A\}$.

**Influence maximization problem on ID model**

Based on the observations of ID model above, it is easy to find that we can partition the result set $Z$ to several disjoint subsets $\{Z_0, Z_1, \ldots\}$ according to the time when nodes in $Z$ become active. Also, if we consider the procedure of information diffusion on ID model as a breadth-first traversal on directed graph $G$, $\{Z_i\}$ is obtained by partition $Z$ using the depth of each node.

The object of general influence maximization problem is to maximize the node set influenced by the seed node set. Obviously, in ID model, information diffusion is a probabilistic process, in which node can become active during the procedure is uncertain. Therefore, we need to understand the procedure based on possible world semantics.

Let $\Omega$ be the set of all different possible worlds of given ID model. In fact, each possible world $X \in \Omega$ can be determined uniquely by giving assignments to all probabilistic variables in the information diffusion. For special $G = (V, E)$, if we do not consider the seed node set, the number of possible worlds is $2^{|E|}$. For more, for given seed set $A$, let $G_A$ be the induced graph of $G$ on node set $A$, $E_A$ be the set of edges in $G_A$, and the number of all possible worlds should be $2^{|E| - |E_A|}$. In the following, we will introduce how to compute the probability of the diffusion process and define the influence maximization problem based on possible world semantics.

**Example 2.** As shown in Figure 1, three possible worlds of given graph $G$ in the ID model are listed. Different possible worlds have different edge selections of $G$. For example, in $G_1$, the edges $(A, C)$, $(C, D)$, $(D, C)$, and $(B, D)$ are not selected by the information diffusion procedure.

It should be noted that in ID model, two different processes may reach the same possible world. Therefore, the probabilities of each single process and possible world are different. Formally, given an information diffusion process $M$, let $\mathcal{S}$ be the set of edges processed during $M$, then the probability $\Pr(G_M)$ can be computed by $\prod_{e \in G_M \setminus \mathcal{S}} \Pr(e) \cdot \prod_{e \in \mathcal{S} \setminus G_M} (1 - \Pr(e))$. Since the edges considered by our information diffusing model are independent from each other, the main idea of $\Pr(G_M)$ is to compute the probability of the whole graph by combining the probabilities of all edges processed during information diffusion. In the following
example, we will explain how to compute the probabilities intuitively.

**Example 3.** Consider the information diffusion process $M$ described in Example 1, let the final possible world be $G_M$. The probability $\Pr(M)$ can be computed by combining the probabilities of all possible choices. That is

$$\Pr(M) = (1 - 0.7) \cdot 0.6 \cdot (1 - 0.9) \cdot 0.72 \cdot 0.9 \cdot 0.7 \cdot 0.075$$

Let $\Pr(s, t)$ represent the edge $(s, t)$ exist in the final possible world. Then, we have

$$\Pr(A, C) = (1 - 0.7) \cdot 0.6 + 0.7 \cdot 1.0 = 0.88$$
$$\Pr(A, B) = 0.9$$
$$\Pr(C, D) = 0.72$$
$$\Pr(C, E) = (1 - 0.9) \cdot 0.49 + 0.9 \cdot 0.7 = 0.679$$
$$\Pr(D, B) = 0.075$$

Therefore, by combining those probabilities, we have

$$\Pr(G_M) = \Pr(A, C) \cdot (1 - \Pr(A, B)) \cdot \Pr(C, D) \cdot \Pr(C, E) \cdot \Pr(D, B)$$

Obviously, not all edges in $G$ should be considered since there are some edges not visited during the information diffusion procedure because of the topology structures.

**Definition 1.** Influence function. Given an ID model $\langle G, P, C, L \rangle$, seed node set $A$, and threshold $\theta$, let $\{G_1, \ldots, G_m\}$ be the set of possible worlds. The influence function $\delta$ can measure the expected value of influence of $A$ on $G$. For special $A$, $\delta(G, A, \theta)$ is defined to be $\sum \Pr(G_i) \cdot |V_{G_i}|$, and it is also denoted by $\delta(A)$ for simplicity.

It can be found that for each possible world $G_i$ its node size is just the node set which can be influenced by $A$.

**Definition 2.** Influence maximization on ID model. Given an ID model $\langle G, P, C, L \rangle$, threshold $\theta$, and an integer $k > 0$, the question is to find a subset $A$ satisfying $|A| = k$ and the size $\delta(A)$ is maximized.

In the following parts, we will use IMD (influence maximization under dynamic influence) to represent the influence maximization problem on ID model.

### Approximation algorithm for IMD problem

In this section, first, the computational complexity of IMD problem is studied which indicates that it is intractable and should be solved by approximation or randomized ways. To design approximation algorithms for IMD problem, the ID model proposed above is simplified to reduce possible worlds of the model and integrate dynamic influence. Finally, an approximation algorithm is proposed and formal analysis shows that the approximation ratio can be efficiently bounded.

#### Hardness of solving IMD problem

Since the influence maximization problem on classic models is usually NP-hard and approximation and heuristic algorithms are often needed, we first consider the computational complexities of IMD problem.

**Theorem 1.** IMD problem is NP-hard.

**Proof.** The theorem can be proved by observing that classical influence maximization problem on the IC model in the work by Kempe et al. is a special case of IMD problem. The details can be analyzed as follows.

For IMD problem, we can prove that they are NP-hard by making a direct reduction from the classical influence maximization problem on IC model in the work by Kempe et al. Given a classical influence maximization instance $I$, by setting the parameter $c_{uv} = 1$, $l_{uv} = 0$ for every $(u, v)$ and $\theta = 0$, it is easy to obtain the corresponding instance $I'$ of IMD problem. Furthermore, it can be easily verified that there are bijective maps between the solutions of $I$ and $I'$. Therefore, IMD problem is NP-hard.

#### Simplifying the ID model

By Example 3, it is easy to find that the computation of possible world probability is tricky and it is hard to process in solving IMD problem. In this part, we propose a method to simplify ID model by integrating the $L$ function with ID model.

Given an instance $I = \langle G, P, L, C \rangle$ of ID model, we can build another instance $I' = \langle G, P', L', C \rangle$ as follows:

- Let the threshold be $\theta$.
- For each edge $(u, v) \in E_G$, if $l_{uv} \geq \theta$, let $p_{uv} = l_{uv} + (1 - l_{uv}) \cdot p_{uv}$.
- For each edge $(u, v) \in E_G$, if $l_{uv} > \theta$, let $p_{uv} = p_{uv}$.
- For each edge $(u, v) \in E_G$, let $l'_{uv} = 0$. 


After such transformation, the new instance will only contain zero-value \( L \) function, in fact, we can delete the \( L \) function in \( I' \).

**Lemma 1.** For each possible world \( G_i \) which is produced by information diffusion process over \( I \) or \( I' \), we have \( \Pr_i(G_i) = \Pr_i(G_1) \).

**Proof.** Considering one possible world \( G_i \), we will show that the probabilities of \( G_i \) are same. Because \( G_i \) is a deterministic graph, by making a breadth-first traversal, it is easy to determine the edges which will not be visited during the information diffusion. For the left edges, we consider the following cases. For special edge \((u, v)\) in \( I \), if \( l_{uv} > \theta \), \( \Pr(u, v) = l_{uv} \cdot c_{uv}^{-1} \). Otherwise, \( \Pr(u, v) = p_{uv} \cdot c_{uv}^{-1} \). The computation of \( \Pr(u, v) \) is based on the definition of information diffusion procedure.

While in \( I' \), since \( L' \) value is 0, for any edge \((u, v)\), the value \( \Pr(u, v) = p_{uv} \cdot c_{uv}^{-1} \). For any edge \((u, v)\) which satisfies \( l_{uv} > \theta \) in \( I \), we have \( \Pr(u, v) = p_{uv} \cdot c_{uv}^{-1} \) according the definition of transformation above. For edge \((u, v)\) satisfying \( l_{uv} \geq \theta \) in \( I \), we have \( \Pr(u, v) = p_{uv} \cdot c_{uv}^{-1} = l_{uv} \cdot c_{uv}^{-1} + (1 - l_{uv}) \cdot p_{uv} \cdot c_{uv}^{-1} \).

Obviously, all edges in the possible world have same probabilities to be chosen. Therefore, each possible world in \( I \) and \( I' \) has same probabilities. Finally, the distributions of possible worlds of \( I \) and \( I' \) are same.

As shown above, the ID instance after transformation has zero-value \( L \) function; therefore, we can ignore the \( L \) function in the following discussions and only use \( \langle G, P, C \rangle \) to represent the instance of ID model. For more, given any instance \( I = \langle G, P, L, C \rangle \) and threshold \( \theta \), the transformation can be finished in polynomial time cost.

**Efficient approximation algorithm**

According to Theorem 1, it is almost impossible to solve IMD problem in polynomial time; therefore, in the following parts, our aim is to find efficient approximation algorithms with performance guarantee. As shown in the work by Kempe et al.,\(^2\) monotone and submodular properties allow us to develop greedy algorithms to achieve \( (1 - 1/e) \) approximation ratio. Here, given function \( \delta() : 2^V \rightarrow R \), \( \delta \) is called to be monotone iff \( \delta(S_1) \leq \delta(S_2) \) for any \( S_1 \subseteq S_2 \), it is called to be submodular iff \( \delta(S_1 \cup x) - \delta(S_1) \geq \delta(S_2 \cup x) - \delta(S_2) \) for any \( S_1 \subseteq S_2 \). Therefore, in the following parts, we will try to utilize such strategies to design efficient approximation algorithms for the influence maximization problems in this article.

We proposed an algorithm based on greedy idea which can produce approximation algorithms with ratio \( 1 - 1/e \) as shown by Fisher et al.\(^5\) The algorithm is shown in Figure 3. The Algorithm **GreedyIMD** takes \( I = \langle G = (V, E), P, C \rangle \) and integer \( k > 0 \) as the input parameters. First, set \( S \) for storing the optimal node seeds and is initialized to be empty (lines 1–2). Then, by considering the node one by one, at each time, the algorithm only chooses a node \( v \) with maximized \( \Delta_v \) and insert it into \( S \) (lines 3–11). \( \Delta_v \) is the influence gain obtained by adding \( v \) to \( S \), that is, \( \delta_{IMD}(S \cup v) - \delta_{IMD}(S) \). Here, the value of function \( \delta() \) is computed by invoking the procedure **GetInfluence** (line 7).

In the **GetInfluence** procedure, the inputs are composed of seed node set \( S \) and the instance \( I \) of ID model, and the goal is to return \( \delta_{IMD}(S) \). As shown in the work by Chen et al.,\(^1\) the problem of computing \( \delta() \) under classic IC model has already been \( tP \) – hard. Therefore, in the **GetInfluence** procedure, we use the sampling method to estimate the value of influence for given seed set. First, the variable for storing the final result influence is initialized to be zero (line 1). Then, the sampling method will be ran for \( n \) times (lines 1–17) (the value of \( n \) can be determine according to the results in the work by Kempe et al.,\(^2\)) and the averaged value of all result influences will be returned (line 18). During each sampling iteration, all temporary variables used will be initialized first (lines 3–6). Then, the seed nodes will be inserted into a queue \( Q \) (line 7). \( Q \) is helpful to do a breadth-first traversal on \( G \). For each node, the probability of becoming active is calculated by \( p_{uv} \cdot c_{uv}^{-1} \) (line 12).

Finally, based on the observation that the main procedure of Algorithm 3 is to iterate among all nodes and the **GetInfluence** procedure only enumerates every edge of \( G \), it is easy to verify that Algorithm **GreedyIMD** can be finished in polynomial time. In **GetInfluence**, the time cost of codes between lines 3 and 17 can be bounded by \( O(|E|^2) \); therefore, the time cost of **GetInfluence** can be bounded by \( O(n \cdot |E|^2) \). Combining **GetInfluence** with **GreedyIMD**, the total time cost of **GreedyIMD** can be bounded by \( O(k \cdot n \cdot |V| \cdot |E|^2) \).

**Analysis of Algorithm GreedyIMD**

In this part, we will show that Algorithm **GreedyIMD** has performance guarantee on the approximation ratio. The main idea is to show the influence function \( \delta_{IMD} \) satisfies the properties of monotone and submodular.

First, we introduce another kind of view of the information diffusion on ID model. According to the results in the work by Kempe et al.,\(^2\) an equivalent view of information diffusion process on IC model is as follows: each edge \((u, v)\) of \( G \) is identified to be **live** independently with probability \( p_{uv} \) and **blocked** otherwise. Therefore, we can use different assignments of **live** and **blocked** states of the edges to represent different results of information diffusion on IC models. Moreover, Kempe et al.,\(^2\) have shown that they have same
distributions. Here, we can also use the similar view of information diffusion process on ID model. For each edge \((u, v)\) of \(G\), \((u, v)\) is identified to be live with probability \(p_{uv} \cdot c_{uv}^{-1}\) and blocked otherwise. It should be noted that the probabilities are not independent any more but depend on the depth \(i\) which is affected by whether other edges have become live. This difference makes the formal analysis even harder, and we will show how to solve this problem in the following.

Second, we introduce a new representation for the influence function \(\delta_{IMD}(\cdot)\). According to the definition of \(\delta_{IMD}\), we have

\[
\delta_{IMD}(A) = \mathbb{E}(F(G, A)) = \sum_{v \in F} \mathbb{E}(f(G, A, v)) = \sum_{v \in F} \sum_{x \in X_G} g(G, x, A, v) \cdot \Pr_v(x, G, A)
\]

where \(F(G, A)\) represents the size of influence node set of \(A\) on network \(G\) with fixed choices of \(P\) and \(C\), and \(f(G, A, v)\) is 1 or 0 which represents whether node \(v\) is in the influence node set of \(A\) on \(G\).

In equation (1), given network \(G\), we use \(X_G\) to represent the set of all different live-blocked assignments on \(E_G\) and \(x\) to represent some special assignment of
follows. First, according to the assignments of \( g \) given seed set \( A \) are different. We use \( \mathbf{Pr}_i(x, G, A) \) to represent those probabilities. The function \( g(G, x, A, v) \) is the particular value of \( f(G, A, v) \) on special \( x \). The value of \( g \) can be determined as follows. First, according to \( x \), a subgraph \( G' \) of \( G \) can be obtained by only taking live edges into \( E_G \). Then, given seed set \( A \), if \( v \) can be influenced by \( A \) on \( G' \), the value \( g \) will be 1 and 0 otherwise. Finally, equation (1) can be obtained directly from the definition of \( \delta_{IMD} \).

In the following, we will show the monotone and submodular property of \( \delta_{IMD} \).

**Lemma 2.** The function \( g \) in equation (1) is monotone.

**Proof.** We can prove the function \( g \) is monotone by analyzing the connectivity between node \( v \) and the seed set \( A \). Obviously, we need to show, given \( S_1 \subseteq S_2 \subseteq V \), there is \( g(G, x, S_1, v) \leq g(G, x, S_2, v) \). Since the value of function \( g \) can be only 1 or 0, we need to show that \( g(G, x, S_1, v) = 1 \) and \( g(G, x, S_2, v) = 0 \) cannot be satisfied at the same time. Assume that \( g(G, x, S_1, v) = 1 \) and \( g(G, x, S_2, v) = 0 \). Because \( g(G, x, S_1, v) = 1 \), let \( G' \) be the subgraph of \( G \) obtained by only choosing live edges identified by \( x \) in \( E_G \), and there must be some node \( u \in S_1 \) such that \( u \) and \( v \) are connected in \( G' \). Since \( S_1 \subseteq S_2 \), we have \( u \in S_2 \). Then, \( v \) is connected by some node in \( S_2 \) that is, \( g(G, x, S_2, v) = 1 \) which is a conflict. Therefore, \( g(G, x, S_1, v) \leq g(G, x, S_2, v) \).

Naturally, it is hoped that \( \mathbf{Pr}_i(x, G, A) \) is monotone. If so, we can obtain the result that \( \delta_{IMD}(\cdot) \) is monotone directly based on equation (1). Unfortunately, we have the following Lemma.

**Lemma 3.** The function \( \mathbf{Pr}_i(x, G, A) \) is not monotone.

**Proof.** It can be understood by the example in Figure 4. Assume that \( c_{uv} = 0.5 \) for every edge \( (u, v) \in G \), \( S_1 = \{v_a\} \), and \( S_2 = \{v_a, v_b\} \). For \( S_1 \), the edges \((v_a, v_b)\) and \((v_c, v_b)\) will be visited in the first iteration, and the edge \((v_a, v_c)\) will be visited in the second iteration. Therefore, \( \mathbf{Pr}_i(x, G, S_1) = 1 \cdot 0.9 \cdot (1 - 0.1 \cdot 0.5) = 0.855 \). For \( S_2 \), the edge \((v_a, v_b)\) will not be processed and the other two edges will be visited in the first iteration. Therefore, \( \mathbf{Pr}_i(x, G, S_2) = 0.9 \cdot (1 - 0.1) = 0.81 \). Finally, we have

\[
\mathbf{Pr}_i(x, G, S_1) > \mathbf{Pr}_i(x, G, S_2)
\]

For another example, let us modify the original graph \( G \) into \( H \) as shown in Figure 4. Similarly, we have \( \mathbf{Pr}_i(x, H, S_1) = 0.9 \cdot 0.9 = 0.81 \) and \( \mathbf{Pr}_i(x, H, S_2) = 0.9 \). Therefore

\[
\mathbf{Pr}_i(x, H, S_1) < \mathbf{Pr}_i(x, H, S_2)
\]

Finally, we can obtain the result that \( \mathbf{Pr}_i(x, G, \cdot) \) is not monotone.

Since \( \mathbf{Pr}_i(x, G, \cdot) \) is not monotone, it is hard to prove the theorem directly based on equation (1) and we need an alternative view of \( \delta_{IMD}(\cdot) \).

**Theorem 2.** The influence function \( \delta_{IMD}(\cdot) \) is monotone.

**Proof.** To prove that the influence function \( \delta_{IMD} \) is monotone, we let us consider the given network \( G = (V, E, P, C) \) and two fixed seed node sets \( S_1 \) and \( S_2 \) satisfying \( S_1 \subseteq S_2 \). We want to prove that \( \delta_{IMD}(S_1) \leq \delta_{IMD}(S_2) \).

Observing that those combinations of \( v \) and \( x \) such that \( g(G, x, A, v) = 0 \) have no contributions to the final result of \( \delta_{IMD}(A) \), we can reform the representation of \( \delta_{IMD}(\cdot) \) as follows

\[
\delta_{IMD}(A) = \sum_{v \in E} Q_v(G, A, \bar{p})
\]

Here, we use \( Q_v(G, A, \bar{p}) \) to represent the probability that node \( v \) can be influenced by \( A \) on the network obtained from \( G \) by selecting live edges with probabilities defined in the information diffusion procedure. The parameter \( \bar{p} \) is a subset of \( P = \{ p_{uv} | (u, v) \in E \} \). Actually, in equation (1), the parameter \( \bar{p} \) is implicitly contained in \( G \). In equation (4), the aim of separating \( \bar{p} \) from \( G \) is to show which variables in \( P \) are essential to the values of \( Q_v \). Therefore, for variables in \( P \), we assume that the function \( Q_v \) only involves \( \bar{p} \). For \( Q_v \), we have following observations:

![Figure 4](image-url)
The set $\bar{p}$ only includes variables of $\{p_{mn}\}$ for $w \in A, (m, n)$ appears on the path between $w$ and $v$. We will prove this by showing that the value change of $p_{mn}$ out of $\bar{p}$ does not affect the value of $Q$. Comparing equation (1) with (4), it is easy to find that $Q_2$ is the sum of several $Pr_G s$, each of which represents the probability of some assignment of $X_G$. Assume that edge $(s, t)$ does not appear on any path between $v$ and nodes in $A$. Given some $x \in X_G$ such that $Pr_G(x, G, A)$ is included in $Q_2$, first, let us assume that $(s, t)$ is labeled as “blocked” in $x$. Obviously, the expression $Pr_G(x', G, A)$ for assignment $x'$ obtained by changing the state of $(s, t)$ to be “live” of $x$ must also appear in $Q$, because adding the edge $(s, t)$ to $G_x$ will not disconnect the paths between $A$ and $v$ and $v$ will still be influenced by $A$ in $G_x$. However, let the edge $(s, t)$ be labeled as “live” in $x$ and $x'$ be obtained by changing the state of $(s, t)$ to be “blocked” in $x$. Because $(s, t)$ does not appear on any path between $A$ and $v$, the deletion of $(s, t)$ will not destroy the connectivity between $v$ and $A$. That is, $\bar{v}$ will still be influenced by $A$ in $G_x$. $Pr_G(x', G, A)$ will appear in $Q_2$. Since both $Pr_G(x, G, A)$ and $Pr_G(x', G, A)$ appear in $Q_2$, and the only difference between $x$ and $x'$ is the state of edge $(s, t)$, the sum of $Pr_G(x, G, A)$ and $Pr_G(x', G, A)$ will eliminate the variable $p_{st}$. Thus, $p_{st}$ will not appear in the expression of $Q$. No matter what value $p_{st}$ is assigned to, the value of $Q$ will not change. Finally, $Q_2$ can be denoted by an expression without any variables out of $\bar{p}$.

$Q_2$ is monotone with respect to each variable $p_{mn}$ in $\bar{p}$. Obviously, for any assignment $x \in X_G$ and its corresponding graph $G_x$, if the value of $p_{mn}$ in $\bar{p}$ increases, the edge $(m, n)$ is more likely to appear in the graph $G_x$. If there is no path through $(m, n)$ connecting $A$ and $v$, the change in $p_{mn}$ will not affect the probability that $v$ is influenced in graph $G_x$. Otherwise, if there is indeed one path passing $(m, n)$ and connecting $A$ and $v$, the probability that $v$ will change in $G_x$ will become larger. Therefore, the increase in $p_{mn}$ will not reduce the value of $Q_2$, and $Q_2$ is indeed monotone with respect to variables in $\bar{p}$.

$Q_2$ is monotone with respect to $A$. Given $S \subseteq V$, let $v \in V \setminus S$ and $S' = S \cup v$. We will show that $Q_2(G, S, \bar{p}) \leq Q_2(G, S', \bar{p})$. We divide all edges involved in the procedure of information diffusion into several parts according to the iteration steps they need to be influenced. For example, observing the information diffusion procedure shown in Figure 2, all edges can be labeled with a number representing the steps they utilized. For example, in Figure 2, node $A$ is labeled 0, node $B$ is labeled 3, and node $C$ is labeled 2. According to the definition of $\delta_{IMD}$, the success probability of each edge $(u, v)$ is computed based on the number labeled shown above. The smaller the number, the lesser the influence reduced and the more likely that the edge is successfully used. After inserting some node $v$ into $A$, during the information diffusion procedure, the numbers labeled on some edges will decrease and the probabilities corresponding to those edges will increase. Since we have shown that $Q_2$ is monotone with respect to variables in $\bar{p}$, it is easy to verify that the value of $Q_2$ will increase as the insertion of node $y$. Finally, we have $Q_2$ is monotone with respect to $A$.

Based on the above observations and equation (4), $\delta_{IMD}(A)$ is monotone.

**Theorem 3.** The influence function $\delta_{IMD}(\cdot)$ is submodular.

**Proof.** To prove that the influence function $\delta_{IMD}$ is submodular, let us consider a given network $G = (V, E, P, C)$, two fixed seed node sets $S_1$ and $S_2$ satisfying $S_1 \subseteq S_2$, and one node $u \in V \setminus S_2$. We need to show that $\delta_{IMD}(S_1 \cup u) - \delta_{IMD}(S_1) \geq \delta_{IMD}(S_2 \cup u) - \delta_{IMD}(S_2)$.

Using the similar idea of the proof of Theorem 2, let us consider equation (4), we will explain the proof based on the function $Q_2$ and show the submodular properties for each $Q_2$. For a fixed node $v$, to finish the proof, let us consider the relationship between $u$ and $v$. To be simplicity, we will use $Q_2(S)$ to represent $Q_2(G, S, \bar{p})$.

- The first case is that $u$ cannot be directed to $v$. That is, there are no paths from $u$ to $v$ in $G$. Obviously, the addition of $u$ will not add the paths between sets seed and node $v$, and $u$ will not increase the $Q_2$ values defined over $S_1$ and $S_2$. Another way of changing the value of $Q_2$ is that $u$ affects the topology structures of $S_1$ and $S_2$. Suppose there is one path $p$ between some node $w \in S_1$ and $v$ and the addition of $u$ changes the iteration levels of some edge $(y, z)$ of $p$, the value of $Q_2$ will change because of the change in influence probability for $(y, z)$. However, in that case, there would be also one path between $u$ and $v$ through $y$, that is, conflict with the assumption. Therefore, when $u$ cannot reach $v$ in $G$, we have $Q_2(S_1 \cup u) - Q_2(S_1) \geq Q_2(S_2 \cup u) - Q_2(S_2)$.

- The second case is that there are paths between $u$ and $v$. It should be noted that the topology structure constructed by the information diffusion procedure is in fact a tree and the edges between trees are eliminated by the mechanism that every
node can be activated at most once. Therefore, we can divide all kinds of paths into three types: the path started from \( S_1 \), the path started from \( S_2 \), and the path started from \( u \). We use \( P_1 \), \( P_2 \), and \( P_u \) to denote them, respectively. The value of \( Q_e \) changes because there are edges moving from one set to another set during the addition of \( u \). Let us consider some special edge \( e = (y,z) \). We ignore the trivial cases that \( e \) does not move between sets since the value \( Q_e \), will not change. We have the following two observations. (1) If \( e \) moves from \( P_1 \) to \( P_u \) after inserting \( u \) to \( S_1 \), it will also move from \( P_2 \) to \( P_u \) because \( S_1 \subseteq S_2 \). (2) If \( e \) moves from \( P_1 \) to \( P_2 \) or from \( P_2 \) to \( P_u \), \( u \) can reach \( v \) through the edge \( e \). Therefore, it can be found that for node \( v \), we have \( Q_v(S_1 \cup u) - Q_v(S_1) \geq Q_v(S_2 \cup u) - Q_v(S_2) \).

By combining the above two results and equation (4), we have \( \delta_{IMD}(S_1 \cup u) - \delta_{IMD}(S_1) \geq \delta_{IMD}(S_2 \cup u) - \delta_{IMD}(S_2) \).

**Theorem 4.** Algorithm GreedyIMD can solve the IMD problem with \((1 - 1/e)\) approximation ratio.

**Proof.** According to the result in the work by Kempe et al., since we have Theorems 2 and 3, it is easy to verify that Algorithm GreedyIMD can solve the IMD problem with \((1 - 1/e)\) approximation ratio.

**Improving Algorithm GreedyIMD**

In Algorithm GreedyIMD, during simulating the procedure of information diffusion, there are still many redundant operations to be performed, and we can improve the algorithm by merging and eliminating those operations. The main idea of this part can be explained by the following example. Let us consider an extreme case, suppose in \( G \), that is the original graph in ID model, there is a subset \( V' \subseteq V \) satisfying that \( V' \) forms a connected component and the nodes in \( V' \) have no edges with the nodes outside. It is easy to check that if the seed set \( A \) satisfies \( A \subseteq V' \), we need not process the edges out of \( V' \) in the algorithm for solving influence maximization problem. Obviously, we can improve the performance of GreedyIMD by removing such edges. In fact, the optimization idea comes from the observation that sparse subgraphs exist generally in real applications. For those cases, if some node is specified to be the beginning of the diffusion, some part of the whole graph will never be visited because of the sparse part in the graph. Therefore, for given node \( a \), find and eliminate the part which will never be visited with \( a \) as the beginning node is useful for improving the efficiency of the algorithm.

Based on the idea above, propose improved version of GetInfluenceImproved to solve the influence maximization under ID model. The improved algorithm is shown in Figure 5. First, given the input \( I \) and \( S \), to simplify the computation and enlarge the chances to remove redundant edges, in GetInfluenceImproved, the nodes are considered one by one and their expected influences are estimated (line 2). For each special node, a graph \( G' \) with reversed edges of \( G \) is built (line 3). Then, by making a traversal over \( G' \) from the node \( v \), we can find which edges are related with \( v \) during the information diffusion procedures (lines 4–5). It should be noted that if the node \( v \) has no relations with the given seed set \( S \), we can find that and ignore \( v \) in this step. Also, by a traversal from \( v \), not all nodes in \( S \) need to be reached. Therefore, the unrelated seed nodes are also filtered in this step. By extracting edges from the obtained nodes set, we can get a graph \( G_D \) which is much smaller than \( G \) and the simulation operations will only be performed on \( G_D \) (line 6). For each graph \( G_D \), the simulation process is run multiple times to get a precise estimation (lines 8–29). In each simulation, first the edges \( E_1 \) in \( G_D \) are extracted and added to an queue structure (line 10–14). Then, the following operations are similar to GetInfluence. The average value of influence size obtained by multiple simulations will be added to the final results (lines 29–30). Finally, the result will be returned (line 31). It can be found that the optimization is implemented by preprocessing the original graph and labeling nodes that are not useful. The time cost of GetInfluenceImproved can be bounded by \( O(|V| \cdot n \cdot |E|^3) \). Compared with GetInfluence, the worse-case time cost is increased because there are cases that no subgraphs can be eliminated by the optimization steps.

**Experimental evaluation**

Based on real data sets, we evaluate the performance of Algorithm GreedyIMD and GetInfluenceImproved and compare them with some current influence maximization algorithms. All codes are implemented in C+++, and all experiments are run on a personal computer with Intel Quad CPU 2.33 GHz and 8 GB main memory. All experiments about running time are run five times and the average values are reported.

**Experiment setup**

We ran our experiments on four real data sets, whose summary information is shown in Table 1. The digital bibliography & library project (DBLP) data set is a large network of research collaboration maintained by Michael Ley. In the network of DBLP, the nodes represent the authors of academic papers and there exist one edge between two nodes if and only if the two
corresponding authors have collaborations. For DBLP, we use the coauthor relationships to compute the influence probability between two authors. Twitter is the network composed of twitters and the tweets posted by them, which is the most popular micro-blogging system in the world. In this network, the nodes represent the twitters and the edges represents the “following” relations between them. For Twitter, we use the repost actions to compute the influence probability. Epinions is a network built by who-trust-whom relations. In this network, nodes represent the users and the edges between them represent the trust relation. We use the

**Table 1.** Statistics of real data sets.

| Data set     | Node size | Edge size  | Average degree |
|--------------|-----------|------------|----------------|
| DBLP\(^{30}\) | 980,562   | 7,324,579  | 15.92          |
| Twitter\(^{31}\) | 112,044   | 468,238    | 8.36           |
| Epinions\(^{32}\) | 131,828   | 841,372    | 12.8           |
| Wikivote\(^{33}\) | 7115      | 103,689    | 26.6           |
metadata about trust relation to compute the influence probability. Wikivote is the data about voting administrator on Wiki. In this network, the nodes represent the wiki users and the edges represent the voting relation. We use the metadata about voting actions to compute the influence probability.

**Experimental results and analysis**

We compare the algorithms proposed in this article on qualities of seed sets and running time costs based on the four real data sets.

We use different parameters of $L$ and $C$ to run the experiments. Here, given a constant $c$, the value of function $C$ on each edge is generated randomly with following Poisson distribution. Similarly, $L$ values are generated randomly for each pair of nodes. In the following parts, we use $MID-A$ to represent the GreedyIMD algorithm running with $L$ and $C$ values generated by ($l = 0.2$, $c = 0.2$). Also, we use $MID-B$, $MID-C$, and $MID-D$ to represent GreedyIMD algorithm with parameters ($l = 0.2$, $c = 0.8$), ($l = 0.8$, $c = 0.2$), and ($l = 0.8$, $c = 0.8$), respectively. When comparing the running times, to be clear, we use IMD-A and IMD-D to represent the GreedyIMD algorithm with GetInfluence procedure, while we use IMD-AX and IMD-DX to represent the GreedyIMD algorithm with GetInfluenceImproved procedure.

**Effects of seed sets**

The effect of given seed set can be evaluated by the influenced nodes size. On four data sets, we compare four algorithms with different parameters on seed nodes sets with different sizes. The results are shown in Figure 6. It can be observed that as the size of seed nodes set increases, the size of influenced nodes increases almost with linear speed. The result is expected since in a large enough network, all nodes tend to perform uniformly during the information diffusion. Also, it can be found that when increasing the value of $L$ and $C$ functions, the size of influenced nodes set gets larger. Actually, when $L$ becomes larger, essentially it increases the probability of influence, thus the size of influenced set will get larger. When $C$ becomes larger, it reduces the decreasing speed of time delay parameters and the size of influenced set will also gets larger. This is because that the two parameters are used to control the influence abilities of the set of seed nodes, which will be much clearer in the following experimental results.

**Effects of function L**

The effect of function $L$ is evaluated by the influenced nodes size. On four data sets, fixing the size of the seed nodes to be 20, we compare four algorithms with different $L$ function values. The results are shown in Figure 7. It can be observed that as the value of $L$
increases, the size of influenced nodes increases. When the $L$ value increases, the increasing speed of influenced nodes set becomes slow. In fact, the effect of $L$ values is to increase the original influence probability in a proper level; therefore, when the $L$ value becomes much larger, the increasing speed of influence set size will get slow. Also, it can be found that for fixed $L$ value, when we scale $C$ from 0.2 to 0.8, the size of influenced nodes set gets larger.

Effects of function $C$

The effect of function $C$ is evaluated by the influenced nodes size. On four data sets, fixing the size of the seed nodes to be 20, we compare four algorithms with different $C$ function values. The results are shown in Figure 8. It can be observed that the results are similar to the results of $L$ function. It should be noted that when the value of $C$ is relatively small, the increasing trend of size of influenced nodes set is more sharp than the results in $L$ function. It depends on the working mechanism of $L$ and $C$. $L$ is used in linear way in the diffusing information procedure, while $C$ is used in exponential way.

Running time. For two data sets, we ran the original greedy algorithm and improved greedy algorithm proposed in this article for different sizes of seeds set. The running time results are shown in Figure 9. It can be found that as the size of seeds set increases, the running time cost also increases; when seed node size becomes larger, the increasing speed of running time cost becomes slow. Also, we can find that the improved algorithms are three times in average faster than the original algorithms. This is because of the reduction in computation cost and optimizing strategies.

Conclusion

In this article, based on the observations of information diffusion process on mobile social networks, the ID model for diffusing information under dynamic influence is proposed. By theoretical analysis, we determine the complexities of solving influence maximization on the new model and design efficient algorithms with approximation performance guarantee. By experiments over real data set, the performances of ID model and the algorithms proposed are verified. One possible further question is how to design more efficient algorithms for dynamic influence in social networks. Another question comes from the methods of modeling dynamic influence in this article. Obviously, our methods cannot cover all possibilities of dynamic influences, and we need to investigate more typical representations for dynamic influences and study how to design algorithms for the related influence maximization problem.
Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Basic Research Program (973 Program) of China via grant 2014CB340503, the National Natural Science Foundation of China (NSFC) via grant 61472107 and 61632011, the National Social Science Foundation of China via Grant 14C062, the Research Project for Humanities and Social Science of the Ministry of Education of China via grant 13YJC860013 and the Heilongjiang Social Science Foundation via grant 12D062.

References

1. Chen W, Wang C and Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (KDD '10), pp.1029–1038. New York: ACM, http://doi.acm.org/10.1145/1835804.1835934
2. Kempe D, Kleinberg J and Tardos E. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03), pp.137–146. New York: ACM, http://doi.acm.org/10.1145/956750.956766
3. Kempe D, Kleinberg J and Tardos E. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03), pp.137–146. New York: ACM, http://doi.acm.org/10.1145/956750.956766
3. Chen W, Wang Y and Yang S. Efficient influence maximization in social networks. In: *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD ’09)*, pp.199–208. New York: ACM, http://doi.acm.org/10.1145/1557019.1557047

4. Liu L, Tang J, Han J, et al. Mining topic-level influence in heterogeneous networks. In: *Proceedings of the 19th ACM international conference on information and knowledge management (CIKM ’10)*, pp.199–208. New York: ACM, http://doi.acm.org/10.1145/1871437.1871467

5. Tang J, Sun J, Wang C, et al. Social influence analysis in large-scale networks. In: *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining (KDD ’09)*, pp.807–816. New York: ACM, http://doi.acm.org/10.1145/1557019.1557108

6. Zhou T, Cao J, Liu B, et al. Location-based influence maximization in social networks. In: *Proceedings of the 24th ACM international conference on information and knowledge management (CIKM ’15)*, pp.1211–1220. New York: ACM, http://doi.acm.org/10.1145/2806416.2806462

7. Goyal A, Bonchi F and Lakshmanan LV. Learning influence probabilities in social networks. In: *Proceedings of the third ACM international conference on web search and data mining (WSDM ’10)*, pp.241–250. New York: ACM, http://doi.acm.org/10.1145/1718487.1718518

8. Leskovec J, Adamic LA and Huberman BA. The dynamics of viral marketing. *ACM Trans Web* 2007, 1(1), http://doi.acm.org/10.1145/1232722.1232727

9. Domingos P and Richardson M. Mining the network value of customers. In: *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD ’01)*, pp.57–66. New York: ACM, http://doi.acm.org/10.1145/502512.502525

10. Richardson M and Domingos P. Mining knowledge-sharing sites for viral marketing. In: *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining (KDD ’02)*, pp.61–70. New York: ACM, http://doi.acm.org/10.1145/775047.775057

11. Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. In: *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining (KDD ’07)*, pp.420–429. New York: ACM, http://doi.acm.org/10.1145/1281192.1281239

12. Goyal A, Lu W and Lakshmanan LV. CELF++: optimizing the greedy algorithm for influence maximization in social networks. In: *Proceedings of the 20th international conference companion on world wide web (WWW ’11)*, pp.47–48. New York: ACM, http://doi.acm.org/10.1145/1963192.1963217

13. Chen YC, Peng WC and Lee SY. Efficient algorithms for influence maximization in social networks. *Knowl Inf Syst* 2012, 33(3): 577–601, http://dx.doi.org/10.1007/s10115-012-0540-7

14. Goyal A, Lu W and Lakshmanan LV. SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In: *Proceedings of the 2011 IEEE 11th international conference on data mining (ICDM ’11)*, pp.211–220. Washington, DC: IEEE Computer Society, http://dx.doi.org/10.1109/ICDM.2011.132

15. Jiang Q, Song G, Cong G, et al. Simulated annealing based influence maximization in social networks. In: *Proceedings of the twenty-fifth AAAI conference on artificial intelligence (AAAI’11)*, pp.127–132. Palo Alto, CA: AAAI Press, http://dl.acm.org/citation.cfm?id=2900423.2900443

16. Kimura M and Saito K. Approximate solutions for the influence maximization problem in a social network. In: *Proceedings of the 10th international conference on knowledge-based and intelligent information and engineering systems (KES 2006)*, Bournemouth, 9–11 October 2006, pp.937–944. Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/11893004_120

17. Han M, Yan M, Cai Z, et al. An exploration of broader influence maximization in timeliness networks with opportunistic selection. *J New Comput Appl* 2016; 63: 39–49, http://dx.doi.org/10.1016/j.jnca.2016.01.004

18. Shi T, Cheng S, Cai Z, et al. Retrieving the maximal time-bounded positive influence set from social networks. *Personal Ubiquitous Comput* 2016; 20(5): 717–730, http://dx.doi.org/10.1007/s00779-016-0943-7

19. Kim J, Kim SK and Yu H. Scalable and parallelizable processing of influence maximization for large-scale social networks? In: *Proceedings of the 2013 IEEE 29th international conference on data engineering (ICDE)*, Brisbane, QLD, Australia, 8–12 April 2013, pp.266–277. New York: IEEE.

20. Cai JLZ, Yan M and Li Y. Using crowdsourced data in location-based social networks to explore influence maximization. In: *Proceedings of the 35th annual IEEE international conference on computer communications (IEEE INFOCOM 2016)*, San Francisco, CA, 10–15 April 2016, pp.1–9. New York: IEEE.

21. Han M, Yan M, Cai Z, et al. Influence maximization by probing partial communities in dynamic online social networks. *T Emerg Telecommun Technol*. Epub ahead of print 28 June 2016. DOI: 10.1002/ett.3054.

22. Li G, Chen S, Feng J, et al. Efficient location-aware influence maximization. In: *Proceedings of the 2014 ACM SIGMOD international conference on management of data (SIGMOD ’14)*, pp.87–98. New York: ACM, http://doi.acm.org/10.1145/2588555.2588561

23. Tang S, Yuan J, Mao X, et al. Relationship classification in large scale online social networks. *J Netw Comput Appl* 2016; 63: 39–49, http://dx.doi.org/10.1007/s00779-016-0943-7

24. Chen S, Fan J, Li G, et al. Online topic-aware influence maximization. *Proc VLDB Endow* 2015; 8(6): 666–677, http://dx.doi.org/10.14778/2735703.2735706

25. Cai Z, He Z, Guan X, et al. Collective data-sanitization for preventing sensitive information inference attacks in
social networks. *IEEE T Depend Secure*. Epub ahead of print 26 September 2016. DOI: 10.1109/TDSC.2016.2613521.

26. Han W, Zhu X, Zhu Z, et al. A comparative analysis on weibo and twitter. *Tsinghua Sci Technol* 2016; 21(1): 1–16.

27. Bi R, Li Y and Zheng X. An optimal content caching framework for utility maximization. *Tsinghua Sci Technol* 2016; 21(4): 374–384.

28. Weng J, Lim EP, Jiang J, et al. TwitterRank: finding topic-sensitive influential twitterers. In: *Proceedings of the third ACM international conference on web search and data mining (WSDM ’10)*, pp.261–270. New York: ACM, http://doi.acm.org/10.1145/1718487.1718520

29. Fisher ML, Nemhauser GL and Wolsey LA. *An analysis of approximations for maximizing submodular set functions—II*. Berlin, Heidelberg: Springer, pp.73–87, http://dx.doi.org/10.1007/BFb0121195

30. DBLP. http://dblp.uni-trier.de

31. Twitter. https://twitter.com

32. Epinions. https://snap.stanford.edu/data/soc-epinions1.html

33. Wikivote. https://snap.stanford.edu/data/wiki-vote.html