Viral Marketing 10

10.1 Overview

Via the social interactions among users, information of various topics, e.g., personal interests, products, commercial services, etc. can extensively propagate throughout the networks, where lots of users can get infected and become activated. Meanwhile, the social information diffusion can bring about great commercial values, and create lots of *viral marketing* [29] opportunities. Lots of commercial companies are utilizing the information diffusion phenomenon in online social networks to promote their products or services. For instance, Apple and Huawei have been promoting their latest cell phones via Facebook and Twitter. They can provide some free cell phone samples, coupons, or even cash to certain users (with lots of followers) in Facebook, and ask them to post some good review comments or advertising photos about the cell phone. Such information will propagate to their friends and followers, who may get activated to purchase the cell phone. Commercial promotions via the online social networks have become more and more important in recent years, which even surpass the traditional print media (like newspaper, magazine, TV, and radio). At the same time, viral marketing has also become one of the most important and secure revenue sources for many online social platforms, like Facebook and Twitter.

To achieve the maximum influence in the online social networks, the commercial companies may need to carry out serious investigations to select the initial user set for information spread. Formally, these information diffusion initiators are called the *seed users* in the existing research works [29]. The problem of selecting the optimal set of *seed users* is called the *influence maximization* problem [29,36] (or the *viral marketing* problem). Furthermore, in commercial promotion campaigns, besides releasing their own advertisements, the competitors may release lots of fake news [52] (i.e., rumors [28, 46]) about the other competing products to cheat the consumers. Identification of these rumor initiators in the online social networks timely can avoid the negative impacts on the marketing activities greatly.

In this chapter, we will study the *seed user* and *rumor initiators* identification problems in *viral marketing*, which are all the crucial problems for designing the optimal marketing strategies for companies in carrying out their promotion campaigns. The problems to be studied in this chapter are mostly based on the information diffusion models introduced in the previous chapter.

In Sect. 10.2, we will first introduce the formulation of the influence maximization problem [29], and introduce several existing seed user selection strategies based on either approximation or heuristics [11, 22, 23, 29]. In Sect. 10.3, we will introduce the *intertwined influence maximization* problem [50] for the seed user selection in promoting multiple products with intertwined relationships.

J. Zhang, P. S. Yu, *Broad Learning Through Fusions*,
https://doi.org/10.1007/978-3-030-12528-8_10

By considering the information diffusion across networks, the *cross-network influence maximization* and seed user selection strategies [47,48] will be introduced in Sect. 10.4. To effectively and efficiently detect the *rumor initiators* [51], a new *rumor initiator* detection algorithm [51] is to be introduced in Sect. 10.5, which is introduced based on the signed network setting but can be applied to other networks as well.

10.2 Traditional Influence Maximization

The *influence maximization* problem first proposed in [29] has been studied for several years, and dozens of algorithms have been introduced to select the optimal marketing strategies for the problem. In the *influence maximization* problem, the *marketing strategy* usually refers to the set of seed users selected by the companies involved in the promotion campaign. In this section, we will introduce the traditional *influence maximization* problem, and provide a brief review of the existing seed user selection algorithms proposed for the problem.

10.2.1 Influence Maximization Problem

The *influence maximization* problem is one of the most fundamental research problems, which studies the word-of-mouth effects on promoting new products and making profitable services. Assuming the information diffusion model has been provided, the *influence maximization* problem aims at identifying the optimal marketing strategy (i.e., the seed users), who can lead to the maximal influence in the social networks.

Definition 10.1 (Influence Maximization) Given a network structure $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the users and $\mathcal{E}$ denotes the relationships among them. The *influence maximization* problem aims at selecting a set of seed users $\mathcal{S} \subset \mathcal{V}$, who can lead to the maximal influence inside the network. Generally, the size of the seed user set is limited by some budget, e.g., $|\mathcal{S}| \leq k$.

In the influence maximization problem, the diffusion model is not the focus, which will be provided as black-box taking the initial seed users as the input and producing the influence number as the output. To quantify the impact achieved by a diffusion model, we can introduce the *influence function* [29] here, which projects the initial seed users to the influence (i.e., the number of infected users).

Definition 10.2 (Influence Function) Let $\mathcal{S} \subset \mathcal{V}$ denote the set of seed users who will initiate the influence propagation. Given a diffusion model M, the *influence function* can be represented as $\sigma_M : \mathcal{S} \rightarrow \mathbb{R}$, which projects the seed user set $\mathcal{S}$ to the expected number of infected users after the diffusion process stops. With an input seed user set $\mathcal{S}$, based on the provided diffusion model M, the number of users who can be infected by these seed users can be represented as $\sigma_M(\mathcal{S})$.

With the *influence function* defined above, the *influence maximization* problem can be formally defined as the following optimization problem:

$$\begin{aligned} &\max_{\mathcal{S} \subset \mathcal{V}} \sigma_M(\mathcal{S}) \\ &s.t. |\mathcal{S}| \leq k. \end{aligned} \tag{10.1}$$

Depending on the specific representation of the influence function $\sigma_M(\cdot)$, the above optimization function can have different varying degrees of difficulty. In most cases, based on the diffusion models, like LT and IC [29], introduced in the previous chapter, solving the objective function may need to enumerate all the potential combination of the seed user set, which renders the problem to be NP-hard.

Generally, by selecting more users to initiate the promotion campaign, the diffusion model will achieve a larger influence and can infect more people. For some diffusion models, the *influence function* usually has the *monotonicity* property. Meanwhile, the total number of users in the social network is limited, and the influence cannot keep increasing as the seed user set size increases. As more users are selected as the seed user, the influence gain obtained by involving these extra seed users will degrade steadily. For some diffusion models, the *influence function* normally follows the *marginal decline*, and has the *submodularity* property. These two properties about the *influence function* are very important for the *influence maximization* problem, which serve as the foundations for many approximation solutions to the problem.

Definition 10.3 (Monotonicity) Given an influence function $\sigma_M(\mathcal{S})$ based on the diffusion model M, the function has the *monotonicity* property iff $\sigma_M(\mathcal{S}) < \sigma_M(\mathcal{S}')$ holds for any $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{V}$.

Definition 10.4 (Submodularity) Given an influence function $\sigma_M(\mathcal{S})$ based on the diffusion model M, the function has the *submodularity* property iff $\sigma_M(\mathcal{S} \cup \{u\}) - \sigma_M(\mathcal{S}) \geq \sigma_M(\mathcal{S}' \cup \{u\}) - \sigma_M(\mathcal{S}')$, for all user $u \in \mathcal{V}$, $u \notin \mathcal{S}$, $u \notin \mathcal{S}'$ and $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{V}$.

Currently, most of the existing algorithms proposed for the influence maximization problem are based on either approximation algorithms or heuristics. In the following subsections, we will introduce some representative algorithms belonging to these two categories.

10.2.2 Approximated Seed User Selection

Based on many of the diffusion models, like LT or IC, the *influence maximization* problem is NP-hard to solve. In the case when the *influence function* is both *monotone* and *submodular*, according to the existing works [29], algorithms applying *greedy strategy* can achieve $(1 - \frac{1}{e})$-approximation of the optimal result. In this section, we will introduce several approximation-based seed user selection algorithms, which include *greedy* [29] and *CELF* [11] algorithms.

10.2.2.1 Greedy Algorithm

Based on the classic diffusion models M, like LT and IC models introduced in Sects. 9.2.1.1 and 9.2.2.1, respectively, the *influence maximization* problem is shown to be NP-hard [29]. Given the objective seed user set size k and the diffusion model, the *influence maximization* problem aims at identifying the optimal seed users of size k who can lead to the maximal influence inside the social network. Let $\mathcal{S}$ denote the set of selected seed users, the influence obtained by these seed users can be represented by the influence function $\sigma_M(\mathcal{S})$. Generally, the value of $\sigma_M(\mathcal{S})$ can be obtained by running the diffusion model M on the selected seed user set $\mathcal{S}$. Based on the LT and IC diffusion models, the influence function has both the *monotonicity* and *submodularity* properties, and the *greedy* algorithm can achieve a constant approximation ratio.

Algorithm 1 Greedy algorithm

```
Require: Input network $G$ with user node set $\mathcal{V}$
         Influence function $\sigma_M(\cdot)$
         Seed use set size $k$
Ensure: Seed user set $\mathcal{S}$
initialize $\mathcal{S} = \emptyset$
while $|\mathcal{S}| \leq k$ do
    $u^* = \arg\max_{u \in \mathcal{V} \setminus \mathcal{S}} \sigma_M(\mathcal{S} \cup \{u\}) - \sigma_M(\mathcal{S})$
    $\mathcal{S} = \mathcal{S} \cup \{u\}$
end while
Return $\mathcal{S}$
```

In the *greedy* algorithm, the k seed users are selected in k rounds, and the user who can lead to the maximum marginal influence gain (of the influence function $\sigma_M(\mathcal{S})$) will be selected in each of the rounds as the seed user. For instance,

- *Round 1*: In round 1, the original seed user set $\mathcal{S}^{(0)}$ is initialized as an empty set, i.e., $\mathcal{S}^{(0)} = \emptyset$, which can achieve 0 influence, $\sigma_M(\mathcal{S}^{(0)}) = 0$. The *greedy* algorithm will enumerate all the users in the social network greedily. User u will be selected if $\sigma_M(\mathcal{S}^{(0)} \cup \{u\}) \geq \sigma_M(\mathcal{S}^{(0)} \cup \{v\}), \forall v \in \mathcal{V}$, $u \neq v$. The time cost of round 1 will be $O\left(|\mathcal{V}|(|\mathcal{V}| + |\mathcal{E}|)\right)$, where $O(|\mathcal{V}| + |\mathcal{E}|)$ denotes the time cost in diffusing the information throughout the network.
- *Round i*: In round i ($i > 1$), the seed user set obtained from the last round can be represented as set $\mathcal{S}^{(i-1)}$, and the *greedy* algorithm will enumerate all the remaining users in the social network and add them to the seed user set. The optimal seed user to be selected in this round can be represented as $u = \arg\max_{u \in \mathcal{V} \setminus \mathcal{S}^{(i-1)}} \sigma_M(\mathcal{S}^{(i-1)} \cup \{u\}) - \sigma_M(\mathcal{S}^{(i-1)})$. The time cost of round i will be $O\left((|\mathcal{V}| - (i-1))(|\mathcal{V}| + |\mathcal{E}|)\right)$.

Such an iteratively process continues until the required k seed users have been selected, and these selected *seed user* sets can be formally represented as $\mathcal{S}$. The pseudo-code of the *greedy* algorithm is available in Algorithm 1. Formally, let $\mathcal{S}^*$ denote the optimal seed user solution to the *influence maximization* problem, and $\mathcal{S}^g$ represent the seed user set selected by the greedy algorithm. According to the existing works [29], the approximation ratio of the performance achieved by the *greedy algorithm* is shown to be

$$\frac{\sigma_M(\mathcal{S}^g)}{\sigma_M(\mathcal{S}^*)} \geq 1 - \frac{1}{e}. \tag{10.2}$$

Actually, the exact computation complexity of $\sigma_M(\mathcal{S})$ is left as an open problem [11], in the context of influence maximization. Later on, Chen et al. [11] demonstrate that the exact computation of $\sigma_M(\mathcal{S})$ is actually #-hard. According to the step-wise analysis of the *greedy* algorithm, the running time of the algorithm at the worst case will be $O\left(|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{E}|)\right)$, which renders the *greedy* algorithm hardly applicable to large-scale social network data sets.

10.2.2.2 CELF

Due to the *submodularity* property of the influence function $\sigma_M(\cdot)$, given two seed user sets $\mathcal{S}^{(i)}$ and $\mathcal{S}^{(i+1)}$ in rounds i and $i+1$, the influence gain introduced by adding user u (where $u \in \mathcal{V}$, $u \notin \mathcal{S}^{(i)}$, $u \notin \mathcal{S}^{(i+1)}$) to $\mathcal{S}^{(i+1)}$ will not surpass the introduced influence gain by adding user u to $\mathcal{S}^{(i)}$. Such a property can be utilized in the seed user selection. For instance, assuming there are two seed user candidates u and v, if the influence gain introduced by u in the current round is greater than the influence gain obtained by v in the previous round, then v will not be selected definitely in the current round. Therefore, based on such an intuition, when choosing the seed users in each round, we don't need to enumerate all the remaining users to identify the one achieving the maximum influence gain. It is also the basic idea of the "cost-effective lazy forward" (*CELF*) algorithm [11] to be introduced here.

In the *CELF* algorithm, a heap data structure is maintained, where the node achieving the maximum influence gain is placed at the root. Formally, in the heap, the tree node is represented as a triple $(u, \Delta(\mathcal{S}, u), r)$, where $u \in \mathcal{V} \setminus \mathcal{S}$ represents the id of the remaining nodes, $\Delta(\mathcal{S}, u) = \sigma_M(\mathcal{S} \cup \{u\}) - \sigma_M(\mathcal{S})$ denotes the influence gain by adding u to the current seed user set $\mathcal{S}$, and r denotes the most recent round updating the triple. In each round, the *CELF* algorithm will pick some node triples form the heap to update, and select the optimal one which can introduce the maximum influence gain.

- *Round 1*: In round 1, *CELF* algorithm constructs the heap data structure involving all the nodes in the network based on the calculated influence introduced by them. For all the use node, we can represent the triples as set $\{(u, \sigma_M(\{u\}), 1)\}_{u \in \mathcal{V}}$, which will be used to construct the heap H.
- *Round i*: In round i ($i > 1$), *CELF* algorithm keeps picking the node triple from the root of the heap, updating the influence gain and round number of the node triple, reinserting the node back to the heap. Such a process continues until the node at the root is the current round number i (i.e., we have just updated it, and it still achieves the maximum influence gain among the remaining nodes), which will be deleted from the heap and added to the seed user set.

The pseudo-code of the *CELF* algorithm is available in Algorithm 2, which is shown to be over 700 times faster than the *greedy* algorithm in identifying the same seed user set [11]. Besides the *greedy* and *CELF* algorithms, many other approximation-based seed user identification algorithms have been proposed, which further optimize the *greedy* algorithm to lower down the time complexity, like *CELF++* [22], *SIMPATH* [23]. If the readers are interested in these algorithms, please refer to these reference papers for more detailed information.

Algorithm 2 CELF algorithm

```
Require: Input network $G$ with user node set $\mathcal{V}$
         Influence function $\sigma_M(\cdot)$
         Seed use set size $k$
Ensure: Seed user set $\mathcal{S}$
initialize $\mathcal{S} = \emptyset$, heap $H = \emptyset$
for each $u \in \mathcal{V}$ do
    calculate the influence gain measure gain=$\sigma_M(\{u\})$
    add tuple $(u$, gain, $1)$ to the heap in decreasing order of influence gain measure
end for
while $|\mathcal{S}| < k$ do
    pick node tuple $(u$, gain, $r)$ from the heap root
    if $r == |\mathcal{S}| + 1$ then
        $\mathcal{S} = \mathcal{S} \cup \{u\}$
        delete node tuple $(u$, gain, $r)$ from the heap $H$
    else
        delete node tuple $(u$, gain, $r)$ from the heap $H$
        update gain=$\sigma_M(\mathcal{S} \cup \{u\}) - \sigma_M(\mathcal{S})$
        update $r = |\mathcal{S}| + 1$
        insert node tuple $(u$, gain, $r)$ back to the heap $H$
    end if
end while
Return $\mathcal{S}$
```

10.2.3 Heuristics-Based Seed User Selection

The algorithms introduced in the previous parts are mostly based on the greedy seed user selection strategy, and are not scalable to large-scale networks. Even though some speed-up techniques have been proposed, e.g., CELF, the time complexity of these algorithms can still be very high. In this part, we will introduce a number of seed user selection algorithms based on heuristics, which can select the promising seed users with a much faster speed.

10.2.3.1 Centrality Heuristics

In our daily life, the important users (e.g., the famous *celebrities*) can usually have much more influence in disseminating information. In the real-world online social networks, the posts from famous people (e.g., celebrities, politician, and movie stars) can always influence more people, and people tend to follow them. Viewed in such a perspective, when selecting the seed users, selecting the nodes with large *centrality* [5] measures will be a good choice. As introduced in Sect. 3.3.2, the node centrality score can be defined based on different kinds of metrics, like *node degree* [1] and *PageRank score* [8].

For the user nodes with larger degrees, there will exist more neighbors that these nodes can spread their influence to, as introduced in the LT and IC model. Lots of commercial brands tend to invite them to help share some advertising posts and photos to promote products, as they can infect more people in the network. Viewed in such a perspective, choosing the nodes with large degrees is a good way for selecting the seed users.

However, when calculating the weight among users in LT and IC models, if we apply the Jaccard's coefficient [26] as the weight measure of social links among users, the weight of links incident to the large-degree nodes will be small since their degrees will penalize the diffusion weight greatly. In other words, selecting the nodes merely based on the node degree may have some problems. Therefore, some other works propose to apply PageRank score to select the seed users, where users with larger PageRank scores [8] tend to be selected in advance. Another method proposed for the networks with small diffusion weights is the *degree discount* [11] heuristics to be introduced in the following part.

Besides the diffusion models have introduced in Chap. 9, there also exist many other diffusion models, like the *path*-based models. In the *shortest path* (SP) model proposed in [30], the nodes are activated through the shortest path form the initial seed user set. Based on these diffusion models, some other types of heuristics have been applied, like *distance-based centrality*. Nodes can be sorted according to the average distance from them to all the other nodes in the network, where those with smaller average distances will be picked as the *seed user nodes*.

10.2.3.2 Degree Discount Heuristics

Both the *node degree* and *PageRank score*-based heuristics work very well in the experimental simulations, and they can achieve much larger influence than the other heuristics. However, the influence obtained by them is still much smaller than the *greedy* algorithm. Furthermore, for the nodes with large degrees, the influence they can send out to their neighbors will be relatively small and can hardly activate their neighbors. To resolve such a problem, some works propose to further improve the pure degree-based heuristics, and introduce the *degree discount* method [11].

Let v be a neighbor of user node u in the network. If u has been selected as a seed user, when considering adding node v as a new seed user based on his/her degree, we should not count seed user u as his neighbor towards the degree. Since u has been added to the seed user set already, node v' degree should be discounted by 1 for u, and similarly for the other neighbors who have been selected as the seed nodes. Such a heuristic is applicable to all the diffusion model introduced before.

Algorithm 3 Degree discount heuristics

```
Require: Input network $G$ with user node set $\mathcal{V}$
         Seed use set size $k$
         Diffusion weight $w$
Ensure: Seed user set $\mathcal{S}$
initialize $\mathcal{S} = \emptyset$
for each node $u \in \mathcal{V}$ do
    compute degree $D(u) = |\Gamma(u)|$
    initialize discounted degree $DD(u) = D(u)$
    initialize $T(u) = 0$
end for
while $|\mathcal{S}| < k$ do
    select $u^* = \arg\max_{u \in \mathcal{V} \setminus \mathcal{S}} DD(u)$
    $\mathcal{S} = \mathcal{S} \cup \{u\}$
    for neighbor $v \in \Gamma(u) \setminus \mathcal{S}$ do
        $T(v) = T(v) + 1$
        $DD(v) = 1 + (D(v) - 2T(v) - (D(v) - T(v))T(v)p)p.$
    end for
end while
Return $\mathcal{S}$
```

Specifically, for the IC model with a relatively small diffusion weight $w \rightarrow 0$, a more accurate *degree discount* heuristic has been proposed in [11]. In the IC mode, user u will activate his/her neighbor v with a probability w. If user u has been selected into the seed user set and u can activate v, then we don't need to further add v into the seed user set. In the case that w is small, the two-hop diffusion can be ignored, and the *degree discount* is applied to a local subgraph.

Let v be a user who has not been selected as the seed user yet, we can represent the his/her neighbor as set $\Gamma(v)$. The number of v's neighbors who have been selected as the seed user can be represented as $T(v)$, while the original degree of node v is $D(v) = |\Gamma(v)|$. The expected number of additional nodes in $\Gamma(v)$ to be infected by adding v into the seed user set can be approximately represented as v's *discounted degree*

$$DD(v) = 1 + (D(v) - 2T(v) - (D(v) - T(v))T(v)p) \cdot p. \tag{10.3}$$

where p denotes the activation probability between users. For all the nodes in the network, the *degree discount* method will pick the seed users with larger *discounted degree* iteratively. The pseudo-code of the *degree discount* method is available in Algorithm 3.

10.3 Intertwined Influence Maximization

Besides the traditional *influence maximization* problems about one single product studied based on the online social networks, in the real scenarios, the promotions of multiple products can co-exist in the social networks at the same time. The relationships among the products to be promoted in the network can be very complicated. In this section, we want to maximize the influence of one specific product that we target on in online social networks, where many other products are being promoted simultaneously. The relationships among these product can be obtained in advance via effective market research, which can be *independent*, *competitive*, or *complementary* as introduced in Sect. 9.3. Formally, we define this problem as the interTwined Influence Maximization (TIM) problem [50].

More specifically, depending on the promotional order of other products and the target product, the TIM problem can have two different variants (we don't care about the case that other products are promoted after the target product):

- C-TIM *problem*: In some cases, the other products have been promoted ahead of the target products, where their selected seed users are known and product information has already been propagated within the network. In such a case, the variant of TIM is defined as the Conditional interTwined Influence Maximization (C-TIM) problem.
- J-TIM *problem*: However, in some other cases, the promotion activities of multiple products occur simultaneously, where the *marketing strategies* of all these products are confidential to each other. Such a variant of TIM is defined as the Joint interTwined Influence Maximization (J-TIM) problem.

To solve the above two sub-problems, in this section, we will introduce a unified *greedy* framework interTwined Influence EstimatoR (TIER) proposed in [50]. The TIER method also has two variants: (1) C-TIER (Conditional TIER) for the C-TIM problem, and (2) J-TIER (Joint TIER) for the J-TIM problem. TIER is based on the diffusion model TLT [50] introduced in Sect. 9.3, which quantifies the *impacts* among products with the *intertwined threshold updating strategy* and can handle the intertwined diffusion of these products at the same time. To solve the C-TIM problem, C-TIER will select seed users greedily and is proved to achieve a $(1 - \frac{1}{e})$-approximation to the optimal result. For the J-TIM problem, we show that the theoretical influence of upper and lower bounds calculation is *NP-hard*. Alternatively, we formulate the J-TIM problem as a game among different products and propose to infer the potential *marketing strategies* of other products. The *step-wise greedy* method J-TIER can achieve promising results by selecting seed users wisely according to the inferred marketing strategies of other products.

10.3.1 Conditional TIM

Formally, we can represent the online *social network* as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of users and $\mathcal{E}$ contains the interactions among users in $\mathcal{V}$. The set of n different products to be promoted in network G can be represented as $\mathcal{P} = \{p^1, p^2, \ldots, p^n\}$. For a given product $p^j \in \mathcal{P}$, users who are influenced to buy p^j are defined to be "*active*" to p^j, while the remaining users who have not bought p^j are defined to be "*inactive*" to p^j. User u_i's status towards all the products in $\mathcal{P}$ can be represented as "*user status vector*" $\mathbf{s}_i = (s_i^1, s_i^2, \ldots, s_i^n)$, where s_i^j is u_i's status to product p^j. Users can be activated by multiple products at the same time (even competing products), i.e., multiple entries in *status vector* $\mathbf{s}_i$ can be "*active*" concurrently.

In traditional single-product viral marketing problems, the selected *seed users* will propagate the influence of the target product in the network and the number of users getting activated can be obtained with the *influence function* $\sigma : \mathcal{S} \rightarrow \mathbb{R}$, which maps the selected seed users to the number of influenced users. Traditional one single-product viral marketing problem aims at selecting the optimal seed users $\bar{\mathcal{S}}$ for the target product, who can achieve the maximum influence:

$$\bar{\mathcal{S}} = \arg_{\mathcal{S}} \max \sigma(\mathcal{S}). \tag{10.4}$$

However, in the TIM problem, promotions of multiple products in $\mathcal{P}$ co-exist simultaneously. The influence function of the target product $p^j \in \mathcal{P}$ depends on not only the seed user set $\mathcal{S}^j$ selected for itself but also the seed users of other products in $\mathcal{P} \setminus \{p^j\}$. In the case that the other products are promoted ahead of the target product, we formally define the TIM problem as the *conditional* *intertwined influence function* C-TIER problem and the corresponding influence function is called the *conditional intertwined influence maximization function*.

Definition 10.5 (Conditional Intertwined Influence Function) Formally, let the notation $\mathcal{S}^{-j} = (\mathcal{S}^1, \ldots, \mathcal{S}^{j-1}, \mathcal{S}^{j+1}, \ldots, \mathcal{S}^n)$ be the known seed user sets selected for all products in $\mathcal{P} \setminus \{p^j\}$, the *influence function* of the target product p^j given the known *seed user sets* $\mathcal{S}^{-j}$ is defined as the *conditional intertwined influence function*: $\sigma(\mathcal{S}^j|\mathcal{S}^{-j})$.

C-TIM Problem: C-TIM problem aims at selecting the optimal *marketing strategy* $\bar{\mathcal{S}}^j$ to maximize the *conditional intertwined influence function* of p^j in the network, i.e.,

$$\bar{\mathcal{S}^j} = \arg_{\mathcal{S}^j} \max \sigma(\mathcal{S}^j|\mathcal{S}^{-j}). \tag{10.5}$$

10.3.1.1 Conditional TIM Problem Analysis

In the C-TIM problem, the promotion activities of other products have been done before we start to promote our target product. Subject to the TLT diffusion model, users' thresholds to the target product can be updated with the *threshold updating strategy* after the promotions of other products. Based on the updated network, the C-TIM can be mapped to the *tradition single-product viral marketing*, which has been proved to be *NP-hard* already.

Theorem 10.1 *The* C-TIM *problem is NP-hard based on the* TLT *diffusion model.*

The proof of Theorem 10.1 is omitted and will be left as an exercise for the readers. Meanwhile, based on the TLT diffusion model, the *conditional influence function* of the target product $\sigma(\mathcal{S}^j|\mathcal{S}^{-j})$ is observed to be both *monotone* and *submodular*.

Theorem 10.2 *For the* TLT *diffusion model, the conditional influence function is monotone and submodular.*

Proof We will prove the theorem from two perspectives:

(1) *monotone*: Given the existing seed user sets $\mathcal{S}^{-j}$ for existing products $\mathcal{P} - \{p^j\}$ in the market, let $\mathcal{T}$ be a seed user set of product p^j. Users in the network who are not involved in $\mathcal{T}$ can be represented as $\mathcal{V} - \mathcal{T}$. For the given seed user set $\mathcal{T}$ and the fixed seed users set $\mathcal{S}^{-j}$ of other products, adding a new seed user, e.g., $u \in \mathcal{V} - \mathcal{T}$, to the seed user set $\mathcal{T}$ will not decrease the number of influenced users, i.e., $\sigma(\mathcal{T} \cup \{u\}|\mathcal{S}^{-j}) \geq \sigma(\mathcal{T}|\mathcal{S}^{-j})$.

(2) *submodular*: After the diffusion process of the existing products in $\mathcal{P} - \{p^j\}$ users the thresholds towards product p^j will be updated. Based on the updated network, for two given seed user sets $\mathcal{R}$ and $\mathcal{T}$, where $\mathcal{R} \subseteq \mathcal{T} \subseteq \mathcal{V}$, it is easy to show that $\sigma(\mathcal{R} \cup \{v\}|\mathcal{S}^{-j}) - \sigma(\mathcal{R}|\mathcal{S}^{-j}) \geq \sigma(\mathcal{T} \cup \{v\}|\mathcal{S}^{-j}) - \sigma(\mathcal{T}|\mathcal{S}^{-j})$ with the "*live-edge path*" [29].

10.3.1.2 The C-TIER Algorithm

According to the above analysis, a greedy algorithm C-TIER is proposed to solve the problem C-TIM in this section, whose pseudo-code is available in Algorithm 4. In C-TIER, we select the user u who can lead to the maximum increase of the conditional influence function $\sigma(\mathcal{S}^j \cup \{u\}|\mathcal{S}^{-j})$ at each step as the new seed user. This process repeats until either no potential seed user is available or all the k^j required seed users have been selected. The time complexity of C-TIER is $O(k^j|\mathcal{V}|(|\mathcal{V}| + |\mathcal{E}|))$. Since

Algorithm 4 The C-TIER algorithm

```
Require: input social network $G = (\mathcal{V}, \mathcal{P}, \mathcal{E})$
         target product: $p^j$
         known seed user sets of $\mathcal{P} - \{p^j\}$: $\mathcal{S}^{-j}$
         conditional influence function of $p^j$: $I(\mathcal{S}^j | \mathcal{S}^{-j})$
         seed user set size of $p^j$: $k^j$
Ensure: selected seed user set $\mathcal{S}^j$ of size $k^j$
initialize seed user set $\mathcal{S}^j = \emptyset$
propagate influence of products $\mathcal{P} - \{p^j\}$ with $\mathcal{S}^{-j}$ and update users' thresholds with intertwined threshold updating strategy
while $\mathcal{V} \setminus \mathcal{S}^j \neq \emptyset \wedge |\mathcal{S}^j| \neq k^j$ do
    pick a user $u \in \mathcal{V} - \mathcal{S}^j$ according to equation $\arg\max_{u \in \mathcal{V}} I(\mathcal{S}^j \cup \{u\} | \mathcal{S}^{-j}) - I(\mathcal{S}^j | \mathcal{S}^{-j})$
    $\mathcal{S}^j = \mathcal{S}^j \cup \{u\}$
end while
return $\mathcal{S}^j$.
```

the *conditional influence function* is *monotone* and *submodular* based on the TLT diffusion model, then the *step-wise greedy* algorithms C-TIER, which select the users who can lead to the maximum increase of influence, can achieve a $(1 - \frac{1}{e})$-approximation of the optimal result for the target product.

10.3.2 Joint TIM

C-TIM studies a common case in real-world viral marketing, where different companies have different schedules to promote their products and some can be conducted ahead of the target product. Meanwhile, in this part, we will study a more challenging case: J-TIM, where other products are being promoted at the same time as our target product and the marketing strategies of different products are totally confidential.

Definition 10.6 (Joint Intertwined Influence Function) When the seed user sets of products $\mathcal{P} \setminus \{p^j\}$ are unknown, i.e., $\mathcal{S}^{-j}$ is not given, the *influence function* of product p^j together with other products in $\mathcal{P} \setminus \{p^j\}$ is defined as the *joint intertwined influence function*: $\sigma(\mathcal{S}^j; \mathcal{S}^{-j})$.

J-TIM Problem: J-TIM problem aims at choosing the optimal *marketing strategy* $\bar{\mathcal{S}}^j$ to maximize the *joint intertwined influence function* of p^j in the network, i.e.,

$$\bar{\mathcal{S}}^j = \arg_{\mathcal{S}^j} \max \sigma(\mathcal{S}^j; \mathcal{S}^{-j}), \tag{10.6}$$

where set $\mathcal{S}^{-j}$ can take any possible value.

10.3.2.1 Joint TIM Problem Analysis

When the *marketing strategies* of other products are unknown, the *influence function* of the target product and other products co-exist in the network is defined as the *joint influence function*: $\sigma(\mathcal{S}^j; \mathcal{S}^{-j})$. Meanwhile, by setting $\mathcal{S}^1 = \cdots = \mathcal{S}^{j-1} = \mathcal{S}^{j+1} = \cdots = \mathcal{S}^n = \emptyset$, the J-TIM problem can be mapped to the traditional *single-product influence maximization* problem in polynomial time, which is an NP-hard problem.

Theorem 10.3 *The* J-TIM *problem is NP-hard based on the* TLT *diffusion model.*

Proof We construct an instance of the J-TIM problem by setting $\mathcal{S}^1 = \cdots = \mathcal{S}^{j-1} = \mathcal{S}^{j+1} = \cdots = \mathcal{S}^n = \emptyset$, which will map the J-TIM problem to the traditional *single-product influence maximization* problem in polynomial time. Meanwhile, as proved in [29], the traditional *single-product influence maximization* problem is *NP-hard*. As a result, the J-TIM is also a *NP-hard* problem.

Meanwhile, if all the products in $\mathcal{P} \setminus \{p^j\}$ are *independent* to p^j, the *joint influence function* $\sigma(\mathcal{S}^j; \mathcal{S}^{-j})$ will be both *monotone* and *submodular*.

Theorem 10.4 *Based on the* TLT *diffusion model, the joint influence function is monotone and submodular if all the other products are independent to* p^j*.*

Proof If all the other products are *independent* to product p^j, then the promotion process of all the other products has no effects on the promotion of p^j. According to the *threshold updating strategy* in the TLT diffusion model, users' thresholds to the target product p^j will not be affected by all the remaining products, i.e., the TIM problem identical to the *traditional single-product viral marketing* problem. Furthermore, the *joint influence function* of p^j and all the other products will be reduced to the *influence function* of product p^j in the traditional *single-product influence maximization* setting:

$$\sigma(\mathcal{S}^j; \mathcal{S}^{-j}) \rightarrow \sigma(\mathcal{S}^j). \tag{10.7}$$

Meanwhile, as proved in [29], the *influence function* of p^j in the *single-product influence maximization* setting is both *monotone* and *submodular*. As a result, based on the TLT diffusion model, the *joint influence function* is *monotone* and *submodular* if all the other products are *independent* to p^j.

However, when there exist products in $\mathcal{P} \setminus \{p^j\}$ to be either *competing* or *complementary* to p^j, the *joint influence function* $\sigma(\mathcal{S}^j; \mathcal{S}^{-j})$ will be neither *monotone* nor *submodular*.

Theorem 10.5 *Based on the* TLT *diffusion model, the joint influence function is not monotone if there exist products which are either competing or complementary to the target product* p^j*.*

Proof We propose to prove the above Theorem 10.5 with counterexamples shown in Fig. 10.1a, where we can find one product p^i to be either *competing* or *complementary* to p^j.

Case (1): *when competing products exist*: as shown in the upper two plots in Fig. 10.1a, we have four users in the network $\{A, B, C, D\}$ and we want to select seed users for products p^i and p^j. The influence from A to B and C are 0.3 and 0.5, whose original thresholds to the target product p^j are 0.25 and 0.45, respectively. In the example, the *seed users* selected for two *competing* products p^j and p^i are (1) $\{A\}$ and $\{D\}$, respectively, in competing case 1 at the upper left corner; and (2) $\{A, B\}$ and $\{C\}$ in competing case 2 at the upper right corner. In competing case 1, p^j can influence three users $\{A, B, C\}$ as the influence from A to B and C can both exceed their thresholds, i.e., $\sigma(\mathcal{S}^j = \{A\}; \mathcal{S}^i = \{D\}) = 3$. However, in competing case 2, p^j can only influence two users even though the seed use set has been expanded by adding B as a *seed user*, i.e., $\sigma(\mathcal{S}^j = \{A, B\}; \mathcal{S}^i = \{C\}) = 2$. The reason is that the competing product p^i selects C as the seed user which increase C's threshold towards p^j from 0.45 to 0.55.
So, we can find a counterexample where $\{A\} \subset \{A, B\}$ but $\sigma(\mathcal{S}^j = \{A\}; \mathcal{S}^i = \{D\}) > \sigma(\mathcal{S}^j = \{A, B\}; \mathcal{S}^i = \{C\})$, when there exists *competing* product p^i in the network.

Case (2): *when complementary products exist*: similar counterexamples are shown in the lower two plots of Fig. 10.1a, which are identical to the upper two plots except that the influence from A to C for product p^j is changed to 0.4 and p^i is *complementary* to p^j instead. In complementary case 1,

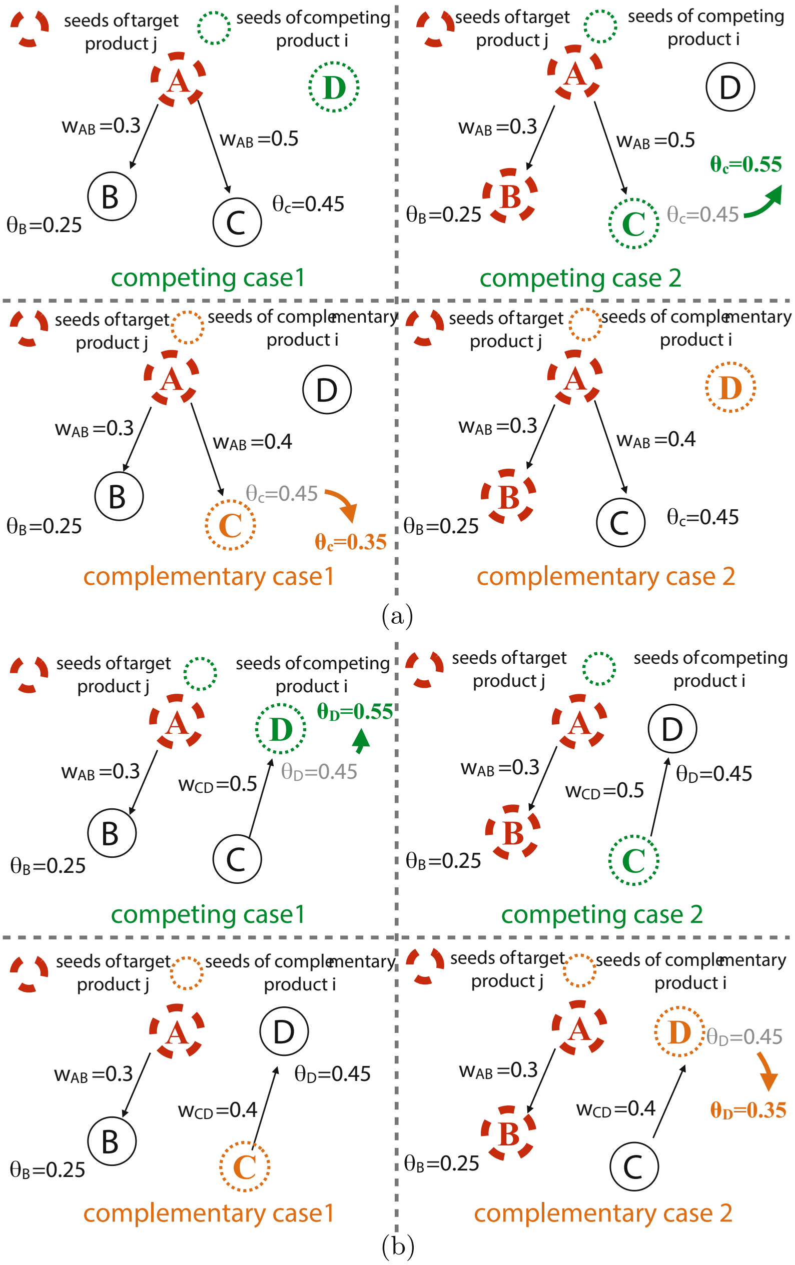


Fig. 10.1 Counterexamples of monotone and submodular properties.
(**a**) Counterexamples of *monotone* property.
(**b**) Counterexamples of *submodular* property

p^i selects C as the seed user, which can decrease C's threshold towards p^j and p^j can achieve a influence of three by choosing A as the seed user. However, in complementary case 2, p^i selects D as the seed user and p^j can only influence two users even though the seed user set has been expanded by adding B to the set.

So, we can find a counterexample where $\{A\} \subset \{A, B\}$ but $\sigma(\mathcal{S}^j = \{A\}; \mathcal{S}^i = \{C\}) > \sigma(\mathcal{S}^j = \{A, B\}; \mathcal{S}^i = \{D\})$ when there exists *complementary* product p^i in the network.

As a result, based on the TLT diffusion model, the *joint influence function* is not *monotone* if these exist products which are either *competing* or *complementary* to p^j.

Theorem 10.6 *For the* TLT *diffusion model, the joint influence function is not submodular if these exist products which are either competing or complementary to the target product p^j.*

Proof We propose to prove Theorem 10.6 with counterexamples shown in Fig. 10.1b, where we can find one product p^i to be either *competing* or *complementary* to p^j.

Case (1): *when competing products exist*: Let $\mathcal{T} = \{A\} \subset \mathcal{S} = \{A, B\}$ and $u = C$. In the competing case 1, $\mathcal{T}$ is the seed user set selected by product p^j and $\{D\}$ is selected as the seed user by product p^i, which increase D's threshold to p^j from 0.45 to 0.55. As a result, p^j can only influence two users ($\{A, B\}$) when using $\mathcal{T}$ as the seed user set and influence three users ($\{A, B, C\}$) when using $\mathcal{T} \cup \{u\}$ as the seed user set. However, in the competing case 2, where p^i selects C as the seed user, p^j can activate two users ($\{A, B\}$) when using $\mathcal{S}$ as the seed user set but can activate four users ($\{A, B, C, D\}$) when using $\mathcal{S} \cup \{u\}$ as the seed user set.

So, we can find a counterexample where $\mathcal{T} = \{A\} \subset \mathcal{S} = \{A, B\}$ and $u = C$, but $\sigma(\mathcal{S}^j = \mathcal{T} \cup \{u\}; \mathcal{S}^i = \{D\}) - \sigma(\mathcal{S}^j = \mathcal{T}; \mathcal{S}^i = \{D\}) < \sigma(\mathcal{S}^j = \mathcal{S} \cup \{u\}; \mathcal{S}^i = \{C\}) - \sigma(\mathcal{S}^j = \mathcal{S}; \mathcal{S}^i = \{C\})$.

Case (2): *when complementary products exist*: similar counterexample is shown in the lower two plots of Fig. 10.1b, where p^i is *complementary* to p^i. We can also find a counterexample where $\mathcal{T} = \{A\} \subset \mathcal{S} = \{A, B\}$ and $u = C$, and $\sigma(\mathcal{S}^j = \mathcal{T} \cup \{u\}, \mathcal{S}^i = \{C\}) - \sigma(\mathcal{S}^j = \mathcal{T}, \mathcal{S}^i = \{D\}) < \sigma(\mathcal{S}^j = \mathcal{S} \cup \{u\}, \mathcal{S}^i = \{C\}) - \sigma(\mathcal{S}^j = \mathcal{S}, \mathcal{S}^i = \{D\})$.

As a result, for the TLT diffusion model, the *joint influence function* is not *submodular* if these exist products which are either *competing* or *complementary* to p^j.

When all the other products are *independent* to p^j, the *joint influence function* of p^j will be *monotone* and *submodular*, which is solvable with the *traditional greedy algorithm* proposed [29] and can achieve a $(1 - \frac{1}{e})$-approximation of the optimal results. However, when there exists at least one product which is either *competing* or *complementary* to p^j, the *joint influence function* will be no longer *monotone* or *submodular*. In such a case, the J-TIM will be very hard to solve and no promising optimality bounds of the results are available.

By borrowing ideas from the game theory studies [6, 40], for product p^j, the lower bound and upper bound of influence the J-TIM problem can be achieved by selecting seed users of size k can be represented as

$$\max_{\mathcal{S}^j} \min_{\mathcal{S}^{-j}} \sigma(\mathcal{S}^j; \mathcal{S}^{-j}), \quad \max_{\mathcal{S}^j} \max_{\mathcal{S}^{-j}} \sigma(\mathcal{S}^j; \mathcal{S}^{-j}) \tag{10.8}$$

respectively, which denotes the maximum influence p^j can achieve in the worst (and the best) cases where all the remaining products work together to make p^j's influence as low (and high) as possible. The *seed user set* selected by p^j when achieving the lower bound and upper bound of influence can be represented as

$$\hat{\mathcal{S}}^j_{low} = \arg\max_{\mathcal{S}^j} \min_{\mathcal{S}^{-j}} \sigma(\mathcal{S}^j; \mathcal{S}^{-j}), \; \hat{\mathcal{S}}^j_{up} = \arg\max_{\mathcal{S}^j} \max_{\mathcal{S}^{-j}} \sigma(\mathcal{S}^j; \mathcal{S}^{-j}). \tag{10.9}$$

However, the lower and upper bounds of the optimal results of the J-TIM problem are hard to calculate mathematically.

Theorem 10.7 *Computing the Max-Min for three or more player games is NP-hard.*

Proof As proposed in [6], the problem of finding any (approximate) Nash equilibrium for a three-player game is computationally intractable and it is NP-hard to approximate the min-max payoff value for each of the player [6, 9, 10, 16].

10.3.2.2 The J-Tier Algorithm

In addition, in the real world, the other products will not co-operate together in designing their marketing strategies to create the worst or the best situations for the target product p^j, i.e., choosing the *marketing strategies* $\mathcal{S}^{-j}$ such that the *joint influence function* $\sigma(\mathcal{S}^j; \mathcal{S}^{-j})$ is minimized or maximized. To address the J-Tim problem, in this part, we propose the J-Tier algorithm to simulate the intertwined round-wise greedy seed user selection process of all the products.

In J-Tier, all products are assumed to be *selfish* and want to maximize their own influence when selecting seed users based on the "*current*" situation created by all the products. J-Tier will infer the next potential *marketing strategies* of other products round by round and select the *optimal* seed users for each product based on the inference.

In algorithm J-Tier, we let all products in $\mathcal{P}$ choose their optimal *seed users* randomly at each round. For example, let $(\mathcal{S})^{\tau-1}$ be the seed users selected by products in $\mathcal{P}$ at round $\tau - 1$. At round τ, a random product p^i can select one seed user. To achieve the largest influence, product p^i will infer the next potential seed users to be selected by other products based on the assumption that they are all selfish. For example, based on p^i's inference, the next seed user to be selected by p^j can be represented as $\bar{u}^j$, i.e.,

$$\arg \max_{u \in \mathcal{V} - (\mathcal{S}^j)^{\tau-1}} [I((\mathcal{S}^j)^{\tau-1} \cup \{u\}; (\mathcal{S}^{-j})^{\tau-1}) - I((\mathcal{S}^j)^{\tau-1}; (\mathcal{S}^{-j})^{\tau-1})]. \tag{10.10}$$

Similarly, p^i can further infer the potential seed users to be selected next by products in $\mathcal{P} \setminus \{p^i, p^j\}$, and these selected seed users can be represented as a set$\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{i-1}, \bar{u}_{i+1}, \ldots, \bar{u}_{j-1}, \bar{u}_{j+1}, \ldots, \bar{u}_n\}$, respectively. Based on such inference, p^i knows who are the next seed users to be selected by other products and will make use of the "prior knowledge" to select its own seed user $\hat{u}^i$ in round τ:

$$\hat{u}^i = \arg \max_{u \in \mathcal{V} - (\mathcal{S}^i)^{\tau-1}} [I((\mathcal{S}^i)^{\tau-1} \cup \{u\}; \bar{\mathcal{S}}^{-i}) - I((\mathcal{S}^i)^{\tau-1}; \bar{\mathcal{S}}^{-i})]. \tag{10.11}$$

where $\bar{\mathcal{S}}^{-i}$ is the "inferred" seed user sets of other products inferred by p^i based on current situation by "adding" these inferred potential seed users to their seed user sets.

The selected $(\hat{u}^i)^\tau$ will be added to the seed user set of product p^i, i.e.,

$$(\mathcal{S}^i)^\tau = (\mathcal{S}^i)^{\tau-1} \cup \{(\hat{u}^i)^\tau\}. \tag{10.12}$$

And the "*current*" seed user sets of all the products, i.e., $\mathcal{S}$, are updated as follows:

$$\mathcal{S} = ((\mathcal{S}^1)^\tau, (\mathcal{S}^2)^{\tau-1}, \ldots, (\mathcal{S}^n)^{\tau-1}). \tag{10.13}$$

The selected $(\hat{u}^i)^\tau$ will propagate his influence in the network and all the users just activated to product p^i will update their thresholds to other products in $\mathcal{P} \setminus \{p^i\}$.

Next, we let another random product (which has not selected seed users yet) to infer the next seed users to be selected by other products and choose its seed user based on the inferred situation. In each round, each product will have a chance to select one seed user and the user selection order of

Algorithm 5 The J-Tier algorithm

```
Require: input social network $G = (\mathcal{V}, \mathcal{P}, \mathcal{E})$
         target product: $p^j$
         set of other products: $\mathcal{P} - \{p^j\}$
         joint influence function of $p^j$: $I(\mathcal{S}^j; \mathcal{S}^{-j})$
         seed user set size of products in $\mathcal{P}$:$k^1, k^2, \ldots, k^j, \ldots, k^n$
Ensure: selected seed user sets $\{\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^n\}$ of products in $\mathcal{P}$ respectively
initialize seed user set $\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^n = \emptyset$
while $(\mathcal{V} \setminus \mathcal{S}^1 \neq \emptyset \vee \cdots \vee \mathcal{V} \setminus \mathcal{S}^n \neq \emptyset) \wedge (|\mathcal{S}^1| \neq k^1 \vee \cdots \vee |\mathcal{S}^n| \neq k^n)$ do
    for random $i \in \{1, 2, \ldots, n\}$ ($p^i$ has not selected seeds in the round yet) do
        if $\mathcal{V} \setminus \mathcal{S}^i \neq \emptyset \wedge |\mathcal{S}^i| \neq k^i$ then
            $p^i$ infers the seed user sets $\bar{\mathcal{S}}^{-i}$ of other products
            $p^i$ selects its seed user $u^i \in \mathcal{V} - \mathcal{S}^i$, who can maximize $I(\mathcal{S}^i \cup \{u^i\}; \bar{\mathcal{S}}^{-i}) - I(\mathcal{S}^i; \bar{\mathcal{S}}^{-i})$
            $\mathcal{S}^i = \mathcal{S}^i \cup \{u^i\}$
            propagate influence of $u$ in $G$ and update influenced users' thresholds to products in $\mathcal{P}$ with the intertwined threshold updating strategy.
        end if
    end for
end while
return $\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^n$.
```

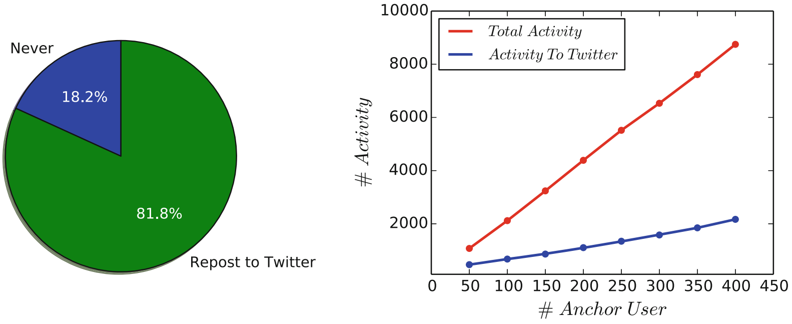
different products in each round is totally random. Such a process will stop when all the products either have selected the required number of *seed users* or no users are available to be chosen. With the J-Tier model, we simulate an alternative seed user selection procedure of multiple products in viral marketing and the pseudo-code J-Tier method is given in Algorithm 5. The time complexity of the J-Tier algorithm is $O((\sum_i k_i \cdot n)|\mathcal{V}|(|\mathcal{V}| + |\mathcal{E}|)$, where $k_i = |\mathcal{S}^i|$ is the number of seed users to be selected for product p^i.

10.4 Cross-Network Influence Maximization

Traditional viral marketing problem aims at selecting the set of seed users to maximize the awareness of ideas or products merely based on the *social connections* among users in *one single social network* [12, 22, 27]. However, in the real world, social networks usually contain heterogeneous information [45,49], e.g., various types of nodes and complex links, via which users are extensively connected and have multiple channels to influence each other [24]. Meanwhile, as studied in [31,49], users nowadays are also involved in multiple social networks simultaneously to enjoy more social network services. Via these shared users, information can propagate not only within but also across social networks [39].

Example 10.1 To support such a claim, we investigate a partially aligned network data set (i.e., Twitter and Foursquare) and the results are given in Fig. 10.2. In Fig. 10.2a, we randomly sample a subset of anchor users from Foursquare and observe that 409 out of 500 (i.e., 81.8%) sampled users have reposted their activities (e.g., tips, location check-ins, etc.) to Twitter. Meanwhile, the activities reposted by these 409 anchor users only account for a small proportion of their total activities in Foursquare, as shown in Fig. 10.2b. In other words, these anchor users will repose the information to other networks selectively.

In this section, we study the influence maximization problem across multiple partially aligned heterogeneous social networks simultaneously. This is formally defined as the *aligned heterogeneous network influence maximization (ANIM)* problem [48]. The *ANIM* problem studied in this section



(a) (b)

Fig. 10.2 Cross-network information propagation analysis. (**a**) Anchor users' reposting. (**b**) Cross-network reposted activities

is very important and has extensive concrete applications in real-world social networks, e.g., *cross-community* [3], even *cross-platform* [39], *product promotion* [42], and *opinion diffusion* [13]. Based on different inter-network information diffusion models introduced in Sect. 9.4, two different seed user selection algorithms will be introduced in this section for the inter-network information diffusion scenario, including the *greedy seed user selection* algorithm [47] and the *dynamic programming-based seed user selection* algorithm [48].

10.4.1 Greedy Seed User Selection Across Networks

In this part, a new information diffusion model named Multi-aligned Multi-relational network (M&M) [47] will be introduced to address the cross-network *seed user selection* challenges. M&M first extracts multi-aligned multi-relational networks with the heterogeneous information across the input *online social networks* based on a set of inter- and intra-network social meta paths [45, 49]. M&M extends the traditional linear threshold (LT) model to depict the information propagation within and across these multi-aligned multi-relational networks. Based on the extended diffusion model, the influence function which maps seed user set to the number of activated users is proved to be both *monotone* and *submodular* [47]. Thus the greedy algorithm used in M&M, which selects seed users greedily at each step, is proved to achieve a $(1 - \frac{1}{e})$-approximation of the optimal result. The M&M diffusion model to be introduced in this part is very similar to the Muse diffusion model introduced in Sect. 9.5, which is also defined based on the meta path concept.

Formally, given two partially aligned networks $G^{(1)}$ and $G^{(2)}$ together with the undirected anchor link set $\mathcal{A}$ between $G^{(1)}$ and $G^{(2)}$, the user sets of $G^{(1)}$ and $G^{(2)}$ can be represented as $\mathcal{U}^{(1)}$ and $\mathcal{U}^{(2)}$, respectively. Let $\sigma(\cdot) : \mathcal{S} \to \mathbb{R}, \mathcal{S} \subset \mathcal{U}^{(1)} \cup \mathcal{U}^{(2)}$ be the *influence function*, which maps the seed user set $\mathcal{S}$ to the number of users influenced by users in $\mathcal{S}$. The *ANIM* problem aims at selecting the optimal set $\mathcal{S}^*$ with size k to maximize the propagation of information across the networks, i.e.,

$$\mathcal{S}^* = \arg \max_{\mathcal{S} \subseteq \mathcal{U}^{(1)} \cup \mathcal{U}^{(2)}} \sigma(\mathcal{S}). \tag{10.14}$$

10.4.1.1 Multi-Aligned Multi-Relational Networks Extraction

We utilize the meta paths [45,49] defined based on the *network schema* to extract multi-aligned multi-relational networks with the heterogeneous information in aligned networks. In both Foursquare and Twitter, users can follow other users and check-in at locations, forming two intra-network influence channels among users. Meanwhile, (1) in Foursquare, users can create/like lists containing a set of locations; (2) while in Twitter, users can retweet other users' tweets, both of which will form an intra-network influence channel among users in Foursquare and Twitter, respectively. The set of intra-network social meta paths considered here as well as their physical meanings are listed as follows:

Intra-Network Social Meta Paths in Foursquare

(1) *follow*: User $\xrightarrow{follow^{-1}}$ User,

(2) *co-location check-ins*: User $\xrightarrow{check-in}$ Location $\xrightarrow{check-in^{-1}}$ User,

(3) *co-location via shared lists*: User $\xrightarrow{create/like}$ List $\xrightarrow{contain}$ Location $\xrightarrow{contain^{-1}}$ List $\xrightarrow{create/like^{-1}}$ User.

Intra-Network Social Meta Paths in Twitter

(1) *follow*: User $\xrightarrow{follow^{-1}}$ User,

(2) *co-location check-ins*: User $\xrightarrow{check-in}$ Location $\xrightarrow{check-in^{-1}}$ User,

(3) *contact via tweet*: User $\xrightarrow{write}$ Tweet $\xrightarrow{retweet}$ Tweet $\xrightarrow{write^{-1}}$ User.

Users can diffuse information across networks via the anchor links formed by anchor users. This can be abstracted as

inter-network social meta path (1) User $\xleftrightarrow{Anchor}$ User.

By taking the inter-network meta paths into account, the studied problem becomes even more complex due to the fact that anchor users in both networks can also be connected via intra- and inter-network meta paths. As a result, the number of social meta path instances grows mightily. Each meta path defines an influence propagation channel among linked users. If linked users u, v are connected by only intra-network meta path, we say u has *intra-network relation* to v, otherwise there is an *inter-network relation* between them. Based on these relations, we can construct multi-aligned multi-relational networks for the aligned heterogeneous networks. The formal definition of multi-aligned multi-relational networks is given as follows.

Definition 10.7 (Multi-Aligned Multi-Relational Networks (MMNs)) For two given heterogeneous networks $G^{(1)}$ and $G^{(2)}$, we can define the multi-aligned multi-relational network constructed based on the above intra- and inter-network social meta paths as $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{R})$, where $\mathcal{U} = \mathcal{U}^{(1)} \cup \mathcal{U}^{(2)}$ denotes the set of user nodes in the MMNs $\mathcal{G}$. Set $\mathcal{E}$ contains the links among nodes in $\mathcal{U}$ and element $e \in \mathcal{E}$ can be represented as $e = (u, v, r)$ denoting that there exists at least one link (u, v) of link type $r \in \mathcal{R} = \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)} \cup \{Anchor\}$, where $\mathcal{R}^{(1)}$, $\mathcal{R}^{(2)}$, and $Anchor$ denote the *intra-network* and *inter-network social meta paths* defined above.

10.4.1.2 M&M Diffusion Model

In this subsection, we will extend the traditional *linear threshold* (LT) model to handle the information diffusion across the multi-aligned multi-relational networks. In the traditional *linear threshold* (LT) model, for one single homogeneous network $G = (\mathcal{V}, \mathcal{E})$, user $u_i \in \mathcal{V}$ can influence his neighbor $u_k \in \Gamma_{in}(u_i) \subseteq \mathcal{V}$ according to weight $w_{i,k} \geq 0$ ($w_{i,k} = 0$ if u_i is *inactive*), where $\Gamma_{in}(u_i)$ represents the users following u_i (i.e., set of users that u_i can influence) and $\sum_{u_k \in \Gamma_{in}(u_i)} w_{i,k} \leq 1$. Each user, e.g., u_i, is associated with a *static threshold* θ_i, which represents the minimal required influence for u_i to become *active*.

Meanwhile, based on the MMNs $\mathcal{G} = (\mathcal{U}, \mathcal{E}, \mathcal{R})$, the weight of each pair of users with different diffusion relations is estimated by PathSim [45]. Formally, the intra-network (inter-network) diffusion weight between user u and v with relation $i(j)$ is defined as

$$\phi^i_{(u,v)} = \frac{2|\mathcal{P}^i_{(u,v)}|}{|\mathcal{P}^i_{(u,)}| + |\mathcal{P}^i_{(,v)}|}, \ \psi^j_{(u,v)} = \frac{2|\mathcal{Q}^j_{(u,v)}|}{|\mathcal{Q}^j_{(u,)}| + |\mathcal{Q}^j_{(,v)}|}, \tag{10.15}$$

where $\mathcal{P}^i_{(u,v)}(\mathcal{Q}^j_{(u,v)})$ denotes the set of intra-network (inter-network) diffusion meta path instances starting from u and ending at v with relation $i(j)$. $|\cdot|$ denotes the size of the set. Thus, $\mathcal{P}^i_{(u,)}(\mathcal{Q}^j_{(u,)})$ and $\mathcal{P}^i_{(,v)}(\mathcal{Q}^j_{(,v)})$ means the number of meta path instances with users u, v as the starting and ending users, respectively.

Based on the traditional LT model, influence propagates in discrete steps in the network. In step t, all *active* users remain *active* and *inactive* user can be *activated* if the received influence exceeds his threshold. Only activated users at step t can influence their neighbors at step $t+1$ and the activation probability for user v in one network (e.g., $G^{(1)}$) with intra-network relation i and inter-network relation j can be represented as $g_v^{(1),i}(t+1)$ and $h_v^{(1),j}(t+1)$, respectively:

$$g_v^{(1),i}(t+1) = \frac{\sum_{u \in \Gamma_{in}(v,i)} \phi^i_{(u,v)} \mathbb{I}(u,t)}{\sum_{u \in \Gamma_{in}(v,i)} \phi^i_{(u,v)}}, \tag{10.16}$$

$$h_v^{(1),j}(t+1) = \frac{\sum_{u \in \Gamma_{in}(v,j)} \phi^j_{(u,v)} \mathbb{I}(u,t)}{\sum_{u \in \Gamma_{in}(v,j)} \phi^j_{(u,v)}} \tag{10.17}$$

where $\Gamma_{in}(v,i)$, $\Gamma_{in}(v,j)$ are the neighbor sets of user v in relations i and j, respectively and $\mathbb{I}(u,t)$ denotes if user u is activated at timestamp t. Note that anchor user $v^{(1)}$ is activated does not mean that his/her corresponding account in network $G^{(2)}$, i.e., $v^{(2)}$, will be activated at the same time, but $v^{(2)}$ will get influence from $v^{(1)}$ via the anchor link.

By aggregating all kinds of intra-network and inter-network relations, we can obtain the integrated activation probability of $v^{(1)}$ [24]. Here logistic function is used as the aggregation function.

$$p_v^{(1)}(t+1) = \frac{e^{\sum_{(i)} \rho_i^{(1)} g_v^{(1),i}(t+1) + \sum_{(j)} \omega_j^{(1)} h_v^{(1),j}(t+1)}}{1 + e^{\sum_{(i)} \rho_i^{(1)} g_v^{(1),i}(t+1) + \sum_{(j)} \omega_j^{(1)} h_v^{(1),j}(t+1)}}, \tag{10.18}$$

where $\rho_i^{(1)}$ and $\omega_j^{(1)}$ denote the weights of each relation in the diffusion process, whose values satisfy $\sum_{(i)} \rho_i^{(1)} + \sum_{(j)} \omega_j^{(1)} = 1, \ \rho_i^{(1)} \geq 0, \ \omega_j^{(1)} \geq 0$. Similarly, we can get the activation probability of a user $v^{(2)}$ in $G^{(2)}$.

10.4.1.3 Problem Solution and Algorithm Analysis

Kempe et al. [29] proved that traditional influence maximization problem is an NP-hard for LT model, where the objective function of influence $\sigma(\mathcal{S})$ is *monotone* and *submodular*. Based on these properties, the greedy approximation algorithms can achieve an approximation ratio of $1 - 1/e$. With the above background knowledge, we will show that the influence maximization problem under the M&M model is also NP-hard and prove the influence spread function $\sigma(\mathcal{S})$ is both *monotone* and *submodular*.

Theorem 10.8 *Influence Maximization Problem across Partially Aligned Heterogeneous Social Networks is NP-hard.*

Theorem 10.9 *For the M&M model, the influence function $\sigma(\mathcal{S})$ is monotone and submodular.*

The proofs of Theorems 10.8 and 10.9 will be left as an exercise for the readers. Since the influence function is both *monotone* and *submodular*, as well as *non-negative*, based on the M&M model, step-wise greedy algorithm introduced in Sect. 10.2.2.1 can be applied to select the seed users who can lead to the maximum marginal influence increase in each step from both networks $G^{(1)}$ and $G^{(2)}$. According to the analysis provided before, such a *step-wise greedy seed user selection* approach can achieve a $(1 - \frac{1}{e})$-approximation of the optimal result.

10.4.2 Dynamic Programming-Based Seed User Selection

In the real world, selecting users as the seed user may introduce certain costs but the cost can be different for users in different networks. Normally, the mature online social networks with a large number of active users may cost more than other smaller-sized online social networks in commercial promotion. In this part, we will still focus on the *influence maximization* problem across multiple aligned social networks. Here, we will introduce another *influence maximization* method to activate users in a specific target network only. We propose to select seed users from both the target network and other aligned source networks subject to certain budget constraint, and these selected users will propagate information to activate users in the target network via both intra- and inter-network information diffusion routes.

Formally, let $G^{(t)}$ and $G^{(s)}$ denote the target and source network, respectively, whose involved user sets can be represented as $\mathcal{U}^{(t)}$ and $\mathcal{U}^{(s)}$, respectively. We can represent the influence function defined based on a certain information diffusion model as $\sigma(\cdot)$, which projects the selected seed user set to the number of infected users in the target network. According to the random walk based information diffusion model introduced in Sect. 9.4.2, to calculate the final number of activated users in $G^{(t)}$, we define a $(|\mathcal{U}^{(s)}|+|\mathcal{U}^{(t)}|)$-dimensional constant vector $\mathbf{b} = [0, 0, \ldots, 0, 1, 1, \ldots, 1]$, where the number of 0 is $|\mathcal{U}^{(s)}|$ and the number of 1 is $|\mathcal{U}^{(t)}|$. Thus the influence function of the IPATH model can be denoted as

$$\sigma(\mathcal{Z}) = \mathbf{b} \cdot h(\pi^*) = \mathbf{b} \cdot h\left(a[\mathbf{I} - (1-a)\mathbf{W}]^{-1} \cdot g(\mathcal{Z})\right). \tag{10.19}$$

From the above function, we can achieve the number of users who can be activated by the seed user set, while the specific user status can be obtained from the status vector π.

The objective function of the *influence maximization* problem studied in this part can be represented as

$$\begin{aligned} \max_{\mathcal{S}} \quad & \sigma(\mathcal{S}) \\ s.t. \quad & \mathcal{S} \subset \mathcal{U}^{(t)} \cup \mathcal{U}^{(s)} \\ & \sum_{u \in \mathcal{S}} c_u \le b, \end{aligned} \tag{10.20}$$

where c_u denotes the introduced cost in adding u in the seed node set $\mathcal{S}$ and b represents the pre-specified budget.

To solve the above problem, we will provide a theoretic analysis about it first and then introduce a new viral marketing method "**I**nfluence **M**aximization algorithm based on **D**ynamic **P**rogramming" (IMDP) proposed in [48]. In IMDP, the information diffusion process is described by the random walk-based diffusion model IPATH introduced in Sect. 9.4.2. Furthermore, IMDP employs dynamic programming to address the problem, and can identify a fully polynomial approximation of the optimal seed user set.

10.4.2.1 Problem Analysis

This section will analyze the problem based on the IPATH model. We first prove that the studied problem is NP-hard. To simplify the notations, let $\mathbf{D} = a[\mathbf{I} - (1-a)\mathbf{W}]^{-1} \in \mathbb{R}^{(|\mathcal{U}^{(s)}|+|\mathcal{U}^{(t)}|)^2}$ (where $a[\mathbf{I} - (1-a)\mathbf{W}]^{-1} \in \mathbb{R}^{(|\mathcal{U}^{(s)}|+|\mathcal{U}^{(t)}|)^2}$ is a term used in the IPATH model as described in Sect. 9.4.2) and $\pi^{(0)} \in \{0,1\}^{(|\mathcal{U}^{(s)}|+|\mathcal{U}^{(t)}|)^2}$ denote the vector indicating the initially selected seed users from both of these networks G^t and $G^{(s)}$. For the entries in vector $\pi^{(0)}$ filled with value 0, the corresponding users are selected as the seed users.

Theorem 10.10 *The problem denoted by Eq.* (10.20) *is NP-hard.*

Proof 0–1 Knapsack Problem, which is NP-hard, can be reduced to the problem in polynomial time. *0–1 Knapsack Problem* is a combination optimization problem: Given a set of items, each with mass w_i and value v_i, the aim of the problem is to determine the number of copies x_i of each kind of item to include in a collection, where x_i is restricted to zero or one, so that the total weight is less than a given limit and their total value is as large as possible, i.e.,

$$\begin{aligned} &\max_{x} && \sum_{i=1}^{n} v_i x_i \\ &\text{s.t.} && \sum_{i=1}^{n} w_i x_i \le W \text{ and } x_i \in \{0,1\} \end{aligned} \tag{10.21}$$

The above objective function is actually equivalent to Eq. (10.20). The constraint of budget in (10.20) is equivalent to the weight limit. In the IPATH model, entry $D(j,u)$ is the information that user j can get when u is the only seed. Thus $\mathbf{b} \cdot h(D(:,u))$ is the number of users activated by u, which is mapped to the value v_i of item i. In the notation, function $h(\cdot)$ denotes the floor function introduced in the IPATH diffusion model. Hence the *0–1 Knapsack Problem* can be reduced to the studied problem in polynomial time, and the problem is also *NP-hard*.

According to Theorem 10.10, there is no polynomial algorithm which can give an optimal solution to the problem, if P≠NP. While the greedy seed user selection algorithm for traditional influence maximization problem is proved to achieve an approximated optimal solution with a factor $(1-1/e)$ [29]. This approximation is achieved when the influence function $\sigma(\mathcal{S})$ is monotone and submodular.

A function $f(\cdot)$ is *monotone* iff $f(\mathcal{A}) \le f(\mathcal{B})$ when $\mathcal{A} \subseteq \mathcal{B}$. We observe that the influence function $\sigma(\mathcal{S})$ is monotone, which means that adding a new seed user, the number of activated users will not decrease. It is also rational intuitively because all values in the weight matrix $\mathbf{W}$ are non-negative, involving more seed users will increase the activation probability of other users. Formally, we can prove such a claim as follows.

Theorem 10.11 *Based on the IPATH diffusion model, the influence function $\sigma(\mathcal{S})$ is monotone.*

Proof Given the current seed user set $\mathcal{S} \subset (\mathcal{U}^{(t)} \cap \mathcal{U}^{(s)})$, by incorporating user $u \in \mathcal{U}^{(t)} \cap \mathcal{U}^{(s)} \setminus \mathcal{S}$ to $\mathcal{S}$, we can represent the influence gain as

$$\begin{aligned}
&\sigma(\mathcal{S} \cup \{u\}) - \sigma(\mathcal{S}) \\
&= \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi^{(0)} + \mathbf{D} \cdot \mathbf{u}\right) - \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi^{(0)}\right) \\
&= \mathbf{b} \cdot h\left(\pi^* + \mathbf{u}^*\right) - \mathbf{b} \cdot h\left(\pi^*\right) \\
&= \mathbf{b} \cdot \left(\lfloor \pi^* + \mathbf{u}^* + \mathbf{c} \rfloor - \lfloor \pi^* + \mathbf{c} \rfloor\right)
\end{aligned} \tag{10.22}$$

The binary vector $\mathbf{u}$ denotes the new seed user u, where only $\mathbf{u}[u] = 1$, and other values are 0. According to Eq. (9.33), $\pi^* = \mathbf{D} \cdot \pi^{(0)}$ represents the information amount of each user can get at convergence with initial state $\pi^{(0)}$. Similarly, $\mathbf{u}^* = \mathbf{D} \cdot \mathbf{u}$ denotes the information amount of each user can get at the convergence state merely with the new seed user u.

Since $\lfloor x + y \rfloor \geq \lfloor x \rfloor + \lfloor y \rfloor$, we can have

$$\begin{aligned}
&\sigma(\mathcal{S} \cup \{u\}) - \sigma(\mathcal{S}) \\
\geq &\mathbf{b} \cdot \left(\lfloor \pi^* + \mathbf{c} \rfloor + \lfloor \mathbf{u}^* \rfloor - \lfloor \pi^* + \mathbf{c} \rfloor\right) = \mathbf{b} \cdot \lfloor \mathbf{u}^* \rfloor
\end{aligned} \tag{10.23}$$

As all elements in both $\mathbf{b}$ and $\mathbf{u}^*$ are non-negative, so $\sigma(\mathcal{S} + \{u\}) - \sigma(\mathcal{S}) \geq 0$, i.e., the influence function $\sigma(\mathcal{S})$ is monotone.

Meanwhile, a function $f(\cdot)$ is submodular, iff $f(\mathcal{A} \cup \{a\}) - f(\mathcal{A}) \geq f(\mathcal{B} \cup \{a\}) - f(\mathcal{B})$ for $\forall \mathcal{A} \subseteq \mathcal{B}$. It implies that for a specific seed user, his marginal contribution will be larger when being added into a smaller seed user set. Meanwhile, we observe that the influence function $\sigma(\mathcal{S})$ does not have such a property, and will prove it with a counterexample as follows.

Theorem 10.12 *Based on the IPATH diffusion model, the influence function $\sigma(\mathcal{S})$ is not submodular.*

Proof To prove the influence function is not submodular, we need to find a pair of seed sets $\mathcal{S}_1$, $\mathcal{S}_2$ and user u ($u \notin \mathcal{S}_1, u \notin \mathcal{S}_2$), where $\mathcal{S}_1 \subseteq \mathcal{S}_2$ and $\sigma(\mathcal{S}_1 \cup \{u\}) - \sigma(\mathcal{S}_1) < \sigma(\mathcal{S}_2 \cup \{u\}) - \sigma(\mathcal{S}_2)$.

$$\begin{aligned}
&\sigma(\mathcal{S}_1 \cup \{u\}) - \sigma(\mathcal{S}_1) - \sigma(\mathcal{S}_2 \cup \{u\}) + \sigma(\mathcal{S}_2) \\
&= \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi_1^{(0)} + \mathbf{D} \cdot \mathbf{u}\right) - \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi_1^{(0)}\right) \\
&\quad - \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi_2^{(0)} + \mathbf{D} \cdot \mathbf{u}\right) + \mathbf{b} \cdot h\left(\mathbf{D} \cdot \pi_2^{(0)}\right) \\
&= \mathbf{b} \cdot h\left(\pi_1^* + \mathbf{u}^*\right) - \mathbf{b} \cdot h\left(\pi_1^*\right) - \mathbf{b} \cdot h\left(\pi_2^* + \mathbf{u}^*\right) + \mathbf{b} \cdot h\left(\pi_2^*\right) \\
&= \mathbf{b} \cdot \left(\lfloor \pi_1^* + \mathbf{u}^* + \mathbf{c} \rfloor - \lfloor \pi_1^* + \mathbf{c} \rfloor - \lfloor \pi_2^* + \mathbf{u}^* + \mathbf{c} \rfloor + \lfloor \pi_2^* + \mathbf{c} \rfloor\right)
\end{aligned} \tag{10.24}$$

We suggest the following counterexample, involving four users $\{u_1, u_2, u_3, u_4\}$. Let $\theta = \frac{3}{4}$, $\mathbf{b} = [0, 0, 1, 1]$, $\mathcal{S}_1 = \emptyset$, $\mathcal{S}_2 = \{u_1\}$ and $u = u_2$, i.e., $\pi_1^{(0)} = [0, 0, 0, 0]^\top$, $\pi_2^{(0)} = [1, 0, 0, 0]^\top$ and $\mathbf{u} = [0, 1, 0, 0]^\top$. In addition, let the weighted diffusion matrix among these four users be

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix} \tag{10.25}$$

Thus the convergence state $\pi_1^* = [0, 0, 0, 0]^\top$, $\pi_2^* = [0, 0, \frac{1}{2}, \frac{1}{2}]^\top$ and $\mathbf{u} = [0, 0, \frac{1}{2}, \frac{1}{2}]^\top$. When we substitute them into (10.24), we can have

$$\begin{aligned}
&\sigma(\mathcal{S}_1 \cup \{u\}) - \sigma(\mathcal{S}_1) - \sigma(\mathcal{S}_2 \cup \{u\}) + \sigma(\mathcal{S}_2) \\
&\quad = \mathbf{b} \cdot \left([0, 0, 0, 0]^\top - [0, 0, 0, 0]^\top - [0, 0, 1, 1]^\top + [0, 0, 0, 0]^\top\right) \\
&\quad = [0, 0, 1, 1] \cdot [0, 0, -1, -1]^\top = -2
\end{aligned} \tag{10.26}$$

Therefore, in such a counterexample, we have $\sigma(\mathcal{S}_1 \cup \{u\}) - \sigma(\mathcal{S}_1) - [\sigma(\mathcal{S}_2 \cup \{u\}) - \sigma(\mathcal{S}_2)] < 0$. It shows the influence function $\sigma(\mathcal{S})$ of the IPATH is not submodular.

Since the influence function is monotone but not submodular, no theoretic performance guarantee exists for the traditional step-wise greedy seed user selection algorithm [29] any more.

10.4.2.2 The IMDP Optimization Algorithm

In the proof of *NP-hardness*, we mentioned user u's contribution is the number of users activated by u, i.e., $p_u = \mathbf{b} \cdot h(\mathbf{D}(:, u))$. We propose to adjust the contribution with a factor $\delta = \frac{\epsilon \cdot p_{\max}}{n}$, where $n = |V_s| + |V_t|$ is the number of all users, $p_{\max} = \max_{u \in (V_s \cup V_t)} p_u$ denotes the largest contribution from all the users. We define $\bar{p}_u = \lfloor \frac{p_u}{\delta} \rfloor$, for $u = 1, 2, \ldots, n$. As $p_{\max}$ is the largest contribution, we cannot get the profit larger than $np_{\max}$, which denotes the contribution upper bound in the dynamic programming. Let $f(i, \rho)$, $(1 \le i \le n, 1 \le \rho \le np_{\max})$ be the smallest cost sum, so that a solution with scaled contribution sum equal to ρ can be obtained by users $j = 1, 2, \ldots, i$. Thus, $f(i, p)$ can be represented as

$$f(i, \rho) = \min \left\{ \sum_{j=1}^{i} c_j : \sum_{j=1}^{i} \bar{p}_j x_j = \rho, x_j \in \{0, 1\}, j = 1, 2, \ldots, i \right\} \tag{10.27}$$

All values of $f(i, \rho)$ can be calculated through the following recurrence:

$$f(i, \rho) = \begin{cases} \min\{f(i-1, \rho), f(i-1, \rho - \bar{p}_i) + c_i\} & \text{if } \quad \bar{p}_i < \rho \\ f(i-1, \rho) & \text{otherwise.} \end{cases} \tag{10.28}$$

When there are several choices (i.e., users) introducing the same amount of contribution, the method will pick one of them randomly as the seed user. We initialize the base case $f(1, \rho)$ as follows:

$$f(1, \rho) = \begin{cases} c_i & \text{if } \rho = \bar{p}_i, \\ \infty & \text{otherwise.} \end{cases} \tag{10.29}$$

Thus the IMDP algorithm first gets the intra-network weight matrix $\mathbf{W}^s$ and $\mathbf{W}^t$, and constructs the inter-network weight matrix $\mathbf{W}^{s \to t}$ and $\mathbf{W}^{s \to t}$. Then the final weight matrix is built with four components as Fig. 9.5 shows in Sect. 9.4.2. At last IMDP uses dynamic programming to identify the optimal seed users across the networks.

10.4.2.3 Analysis of the IMDP Algorithm

In this part, we will analyze the performance of IMDP from the theoretical perspective. We will prove that by scaling with respect to the desired ϵ, and we will be able to get a solution that is at least $(1-\epsilon) \cdot OPT$ (the optimal solution) in polynomial time with respect to both n and $1/\epsilon$.

Lemma 10.1 *The set $\mathcal{S}$ output by the* IMDP *satisfies*

$$\sigma(\mathcal{S}) \geq (1-\epsilon) \cdot OPT. \tag{10.30}$$

Proof Let O be the optimal seed set activating the maximum users. For any user $\bar{p}_u = \lfloor \frac{p_u}{\delta} \rfloor$, thus $0 \leq p_u - \delta \bar{p}_u \leq \delta$. Therefore the profit of the optimal set O can decrease is at most $n\delta$:

$$\sigma(O) - \delta \cdot \bar{\sigma}(O) \leq n\delta \tag{10.31}$$

where $\bar{\sigma}(\cdot)$ denotes the influence function scaled by the factor δ.

$$\begin{aligned} \sigma(\mathcal{S}) &\geq \delta \cdot \bar{\sigma}(O) && (10.32)\\ &\geq \sigma(O) - n\delta = OPT - \epsilon \cdot p_{\max} && (10.33)\\ &\geq (1-\epsilon) \cdot OPT && (10.34) \end{aligned}$$

Inspired by [34], the seed set $\mathcal{S}$ selected from the dynamic programming is optimal for the scaled instance and therefore must be at least as good as choosing the set O with the smaller profits.

The approximation algorithm IMDP is said to be a *polynomial time approximation scheme*, if for each fixed $\epsilon > 0$, its running time is bounded by a polynomial in the size n. And the *fully polynomial time approximation scheme* is an approximation scheme for which the algorithm is bounded polynomially in both the size n and $1/\epsilon$. We prove that the IMDP method is a fully polynomial approximation scheme for the ANIM problem, with the following Theorem 10.13.

Theorem 10.13 *The* IMDP *method is a fully polynomial approximation scheme for the ANIM problem.*

Proof Since $\delta = (\epsilon \times p_{\max})/n$, the running time of IMDP is $O(n^2 \lfloor \frac{p_{\max}}{\delta} \rfloor) = O(n^2 \lfloor \frac{n}{\epsilon} \rfloor)$, which is polynomial in both n and $1/\epsilon$. As shown in Lemma 10.1, the IMDP framework can achieve a $(1-\epsilon)$-approximation of the optimal result.

10.5 Rumor Initiator Detection

This section is a follow-up problem based on the MFC diffusion model introduced in Sect. 9.3.2 based on signed networks. Rumor initiation and incorrect information dissemination are both common in social networks [44]. Incorrect rumors sometimes can bring about devastating effects, and an important goal in improving the credibility of the social channel is to identify rumor initiators [35, 41, 43, 44] in signed social networks. This section studies the detection of rumor initiators in infected signed social networks, given the state of the network at a specific moment in time.

To identify the rumor initiators, we study the problem based on the MFC diffusion model introduced in Sect. 9.3.2. Although the exact identification of the rumor initiators is NP-hard for general graphs, but it can be resolved in polynomial time for binary tree structured networks, and it provides the insights for high quality solutions in the general case. We leverage these insights to introduce the RID framework [51] to identify the optimal *rumor initiators*, including their number, identities, and initial states. The readers are suggested to read this section together with Sect. 9.3.2 introduced in the previous chapter. Here, we will not introduce the definitions of the *weighted signed social network* and *weighted signed diffusion network* concepts again, the readers may refer to Sect. 9.3.2 for more information.

The social psychology literature defines a *rumor* as a story or a statement in general circulation without confirmation or certainty of facts [2]. The originators of rumors are formally defined as *rumor initiators*, which can be individuals, groups, or institutes. In this section, we refer to *rumor initiators* as the users who initially spread the rumor to other users in online social networks. Within the *diffusion networks*, *rumors* can spread from the *initiators* to other users via diffusion links, which will lead to *infected signed diffusion networks*. Since all networks studied in this section are all weighted and signed by default, we will refer to them as *diffusion networks* for simplicity.

Definition 10.8 (Infected Diffusion Network) The *infected diffusion network* $G_I = (\mathcal{V}_I, \mathcal{E}_I, s_I, w_I)$ is a subgraph of the complete diffusion network G_D, where $\mathcal{V}_I \subseteq \mathcal{V}_D$ is the set of infected users, $\mathcal{E}_I \subseteq \mathcal{E}_D$ is the set of potential diffusion links among these infected users. s_I, w_I are the *sign* and *weight* mappings, whose domains are all those diffusion links in $\mathcal{E}_I$.

Definition 10.9 (Activation Link) Among all the links $\mathcal{E}_I$ in the infected diffusion networks, link (u, v) is called an activation link iff u activates v in the screenshot of the infected diffusion network.

Based on the MFC model introduced before, each node in the infected diffusion network screenshot can be activated by exactly one node via the *activation link* and the *rumor initiators* have no incoming activation links. As a result, all the nodes in $\mathcal{V}_I$ together with the activation links among them can actually form a set of cascade trees, where nodes at higher levels are activated by nodes in the lower levels and *rumor initiators* are the roots (at level 1).

In this section, our main goal is to work backwards from the available state of the network given at any moment in time, and we will use the developed diffusion model to track down the rumor initiators. Let $\mathcal{I} \subseteq \mathcal{V}_I \subseteq \mathcal{V}$ be the potential set of *rumor initiators*, whose initial states towards the rumor can be represented as $\mathcal{S} = \{+1, -1\}^{|\mathcal{I}|}$, where $+1$ indicates a belief in the fact at hand, and -1 denotes belief in the opposite fact. We use binary modes of information propagation because of its relative simplicity and intuitive appeal in modeling a variety of situations. The ISOMIT problem aims at inferring the optimal *rumor initiator* set $\mathcal{I}^*$ as well as their initial states $\mathcal{S}^*$, which can maximize the likelihood that it will lead to the current state of the *infected signed network* G_I:

$$\mathcal{I}^*, \mathcal{S}^* = \arg\max_{\mathcal{I}, \mathcal{S}} \mathbf{P}(G_I | \mathcal{I}, \mathcal{S}), \tag{10.35}$$

Here, $\mathbf{P}(G_I | \mathcal{I}, \mathcal{S})$ represents the likelihood of obtaining the infected network G_I based on the influence propagated from $\mathcal{I}$ with states $\mathcal{S}$.

Formally, we will call the above problem as the "Infected Signed netwOrk ruMor Initiator deTection" (ISOMIT) problem. In summary, the input of the ISOMIT problem is the infected signed network G_I, while the objective output is the inferred rumor initiators $\mathcal{I}$ together with their initial states $\mathcal{S}$ which can maximize the likelihood $\mathbf{P}(G_I | \mathcal{I}, \mathcal{S})$.

10.5.1 The ISOMIT Problem

Given the *rumor initiators* $\mathcal{I}$ together with their initial states $\mathcal{S}$, influence can propagate from them to other users in the network via different paths. For any user u in the infected network, the influence propagation paths from *initiators* to u can be represented as the set $\{\mathcal{P}(u_i, u)\}_{u_i \in \mathcal{I}}$, where $\mathcal{P}(u_i, u)$ represents the set of paths from initiator u_i to user u specifically. Each path (e.g., $p \in \mathcal{P}(u_i, u)$) is a sequence of directed diffusion links from u_i to u. We use the notation $(x, y) \in p$ to denote the fact that the diffusion link (x, y) lies on path p. Depending on the sign of link (u, v) as well as the states of u and v, link (u, v) can be either *sign consistent* or *sign inconsistent*.

Definition 10.10 (Sign Inconsistent Diffusion Link) Diffusion link (u, v) is defined to be *sign inconsistent* if $s(u) \cdot s(u, v) \neq s(v)$.

The probability that $u \in \mathcal{V}$ is infected with state $s(u)$ because of influence from the initiators $\mathcal{I}$ with state $\mathcal{S}$ can be computed as

$$\begin{aligned} &\mathbf{P}\left(u, s(u)|\mathcal{I}, \mathcal{S}\right) \\ &= 1 - \prod_{i \in \mathcal{I}} \prod_{p \in \mathcal{P}(i,u)} \left(1 - \prod_{(x,y) \in p} g\left(s(x), s_I(x, y), s(y), w_I(x, y)\right)\right), \end{aligned} \tag{10.36}$$

where the function

$$\begin{aligned} &g\left(s(x), s_I(x, y), s(y), w_I(x, y)\right) \\ &= \begin{cases} \min\{1, \alpha \cdot w_I(x, y)\}, & \text{if } s(x) \cdot s_I(x, y) = s(y), s_I(x, y) = +1, \\ w_I(x, y), & \text{if } s(x) \cdot s_I(x, y) = s(y), s_I(x, y) = -1, \\ 0, & \text{if } s(x) \cdot s_I(x, y) \neq s(y). \end{cases} \end{aligned} \tag{10.37}$$

Consider a link (x, y) lying on the path from rumor initiators in $\mathcal{I}$ to u, such that states of x and y are consistent (i.e., $s(x) \cdot s_I(x, y) = s(y)$). In such a case, the probability of link (x, y) being an activation link would be $\min\{1, \alpha \cdot w_I(x, y)\}$ if (x, y) is a positive link (due to the boosting of positive links in MFC model), and it would be $w_I(x, y)$, otherwise. However, in case of inconsistency (i.e., $s(x) \cdot s_I(x, y) \neq s(y)$), link (x, y) will be either not an activation link or was an activation link originally but y's state is flipped by some other nodes. In other words, y would not be activated by x in the screenshot of the infected diffusion network, and the $g(\cdot)$ is assigned with value one in the *sign inconsistent* case.

One can model the probability of the current state of the *infected signed network* G_I, conditional on the *rumor initiators* $\mathcal{I}$ with initial states $\mathcal{S}$ as follows:

$$\mathbf{P}(G_I|\mathcal{I}, \mathcal{S}) = \prod_{u \in \mathcal{V}_I} \mathbf{P}\left(u, s(u)|\mathcal{I}, \mathcal{S}\right). \tag{10.38}$$

10.5.2 NP-Hardness of Exact ISOMIT Problem

Based on the aforementioned remarks, we will show that obtaining the whole *infected networks* exactly based on $\mathcal{I}$ and $\mathcal{S}$ achieving 100% inference probability with minimum number of rumor initiators is an NP-hard problem.

Lemma 10.2 *Based on the* MFC *diffusion model, the* Isomit *problem of achieving probability* $\mathbf{P}(G_I|\mathcal{I}, \mathcal{S}) = 1$ *with the minimum number of initiators is NP-hard.*

Proof We will prove the lemma by showing that the set-cover problem (which is known to be NP-hard) can be reduced to the Isomit problem in polynomial time. Formally, given a set of elements $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ and a set of m subsets of $\mathcal{E}$, $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m\}$, where $\mathcal{L}_i \subseteq \mathcal{E}, i \in \{1, 2, \ldots, m\}$. The set-cover problem aims at finding as few subsets as possible from $\mathcal{L}$, so that the union of the selected subsets is equal to $\mathcal{E}$, i.e., $\bigcup \mathcal{L}_i = \mathcal{E}$ [21].

For an arbitrary instance of the set-cover problem, we define an instance of the infected signed graph to be a directed graph, denoted by G_I. The graph G_I contains $n + m + 1$ nodes: (1) for each element $e_i \in \mathcal{E}$, we construct a corresponding node n_i; (2) for each set $\mathcal{L}_j \in \mathcal{L}$, we construct node n_{j+n}; and (3) a dummy node d (i.e., the $(n + m + 1)_{th}$ node) is added to the infected network. The links in G_I include: (1) for all the elements in each set, e.g., $e_i \in \mathcal{L}_j$), we add a directed link connecting their corresponding nodes in the graph from n_i to n_{j+n}; (2) all the corresponding nodes of elements in $\mathcal{E}$ are connected to d via a directed link; and (3) d connects to the corresponding nodes of sets in $\mathcal{L}$ by directed links as well. The signs of all these links are all assigned $+1$, whose weights are: (1) $w(n_i, n_{j+n}) = 1$, for $\forall e_i \in \mathcal{E}, \forall e_i \in \mathcal{L}_j, \mathcal{L}_j \in \mathcal{L}$; (2) $w(n_i, d) = \frac{1}{n}$, for $\forall e_i \in \mathcal{E}$; (3) $w(n_{j+n}, d) = 1$, for $\forall \mathcal{L}_j \in \mathcal{L}$.

Now, we want to activate all the nodes in G_I with state $+\mathbf{1}$ (i.e., all trust the rumor) with as few rumor initiators as possible. Based on G_I, the solution to the Isomit problem will be equivalent to the set-cover problem based on elements $\mathcal{E}$ and subsets $\mathcal{L}$.

10.5.3 A Special Case: k-ISOMIT-BT Problem

In the previous section, the Isomit problem of achieving probability 100% with the minimum number of initiators is proven to be NP-hard. In this part, we will study a special case of the Isomit problem, where the number of *rumor initiators* is known to be k and the network is a binary tree, i.e., the k-ISOMIT-BT (k Isomit on Binary Tree) problem. We will show that the k-ISOMIT-BT problem can be addressed efficiently in polynomial time. This will also provide the insight needed to solve the general case to be introduced in the next section.

Let $T_I = (\mathcal{V}_I, \mathcal{E}_I, s_I, w_I)$ be an infected signed binary tree. If the user node $u \in \mathcal{V}_I$ is regarded as the root in the tree, its left and right children can be represented as $left(u)$ and $right(u)$, respectively. At the beginning, the *rumor initiator set* and the *state set* is empty, i.e., $\mathcal{I} = \emptyset$ and $\mathcal{S} = \emptyset$. The cost of the optimal solution (i.e., the inferred initiators $\mathcal{I}$ and states $\mathcal{S}$) can be recursively computed with the following dynamic programming equation:

$$\mathbf{OPT}(u, \mathcal{I}, \mathcal{S}, k) = \max \Bigg\{$$

$$\min_{m=0}^{k} \Big\{ \mathbf{OPT}\big(left(u), \mathcal{I}, \mathcal{S}, m\big) + \mathbf{OPT}\big(right(u), \mathcal{I}, \mathcal{S}, k - m\big) + \mathbf{P}\big(u, s(u)|\mathcal{I}, \mathcal{S}\big) \Big\}; \tag{10.39}$$

$$\mathbf{P}\big(u, s(u) = +1|\mathcal{I} \cup \{u\}, \mathcal{S} \cup \{s(u) = +1\}\big) + \min_{m=0}^{k-1} \Big\{ \mathbf{OPT}\big(left(u), \mathcal{I} \cup \{u\},$$

$$\mathcal{S} \cup \{s(u) = +1\}, m\big) + \mathbf{OPT}\big(right(u), \mathcal{I} \cup \{u\}, \mathcal{S} \cup \{s(u) = +1\}, k - 1 - m\big) \Big\}; \tag{10.40}$$

$$\mathbf{P}\big(u, s(u) = -1 | \mathcal{I} \cup \{u\}, \mathcal{S} \cup \{s(u) = -1\}\big) + \min_{m=0}^{k-1} \Big\{ \mathbf{OPT}\big(left(u), \mathcal{I} \cup \{u\},$$
$$\mathcal{S} \cup \{s(u) = -1\}, m\big) + \mathbf{OPT}\big(right(u), \mathcal{I} \cup \{u\}, \mathcal{S} \cup \{s(u) = -1\}, k - 1 - m\big)\Big\}\Big\}. \quad (10.41)$$

From root u, the optimal *rumor initiator* detection can generally follow one of three cases:

- u is not the *initiator*: The root u is not added to the *rumor initiator* set, and we make recursive calls with its left and right children nodes to identify the k *rumor initiators*.
- u is the *initiator* with state $s(u) = +1$: The root u and its state are added into the *rumor initiator* set and the *state* set, respectively (i.e., $\mathcal{I} \cup \{u\}$, and $\mathcal{S} \cup \{s(u) = +1\}$). Furthermore, we make recursive calls with its left and right children nodes to identify the remaining $k - 1$ *rumor initiators* based on the updated *rumor initiator* and their *state*.
- u is the *initiator* with state $s(u) = -1$: The root u and its state are added into the *rumor initiator* set and *state* set, respectively (i.e., $\mathcal{I} \cup \{u\}$, and $\mathcal{S} \cup \{s(u) = -1\}$). Furthermore, we make recursive calls with its left and right children nodes to identify the remaining $k - 1$ *rumor initiators* based on the updated *rumor initiator* and their *state*.

The formal definition of $\mathbf{P}(u, s(u)|\mathcal{I}, \mathcal{S})\}$ is available in Sect. 10.5.1. Meanwhile, the special case $\mathbf{P}(u, s(u)|\{u\}, \{s(u)\})$, for a *single node* u, is computed as follows:

$$\mathbf{P}(u, s(u)|\{u\}, \{s(u)\}) = \begin{cases} 1, & \text{if } s_I(u) = s(u); \\ 0, & \text{if } s_I(u) \neq s(u), \end{cases} \quad (10.42)$$

where $s_I(u)$ is the real state of u in the infected network.

The aforementioned dynamic programming objective function can be addressed in polynomial time, and we will not introduce the details involved in solving it here due to the limited space.

10.5.4 RID Method for General Networks

For the ISOMIT problems in social networks of general structure and an unknown number of rumor initiators, the method introduced in the previous section cannot be directly applied. In this section, we will introduce the RID framework to address the ISOMIT problem. We propose to first detect the infected connected components from the whole network. For each detected connected component, we propose to further prune the non-existing activation links among users to extract the "*infected cascade trees*" in the signed networks. From each infected cascade tree, we introduce the objective function to detect the optimal rumor initiators (the number, identities as well as their states).

10.5.4.1 Infected Connected Components Detection

The infected diffusion network can contain multiple infected connected components, where users in each component can be connected to each other via potential diffusion links among them. In this part, we will introduce the method to detect the infected connected components from the network.

Definition 10.11 (Infected Connected Components) An *infected connected component* is a subgraph of the *infected network* and, by ignoring the directions of diffusion links, any two vertices in the component are connected to each other.

The *signed connected components* in the pruned networks can be detected with algorithms, like breadth-first search (BFS) [15] and depth-first search (DFS) [15], in linear time. For instance, based on the BFS algorithm, we will loop through all the infected vertices in the pruned infected signed network and once we reach an unvisited vertex, e.g., u, we will call BFS function to find the entire connected component containing u. The time cost of BFS-based connected component detection algorithm will be $O(n+m)$, where n and m are the numbers of user nodes and diffusion links in the infected diffusion network.

10.5.4.2 Signed Infected Cascade Forest Extraction

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ be the set of l connected components detected in the pruned infected signed network. As introduced earlier, the real information diffusion process in the infected connected component based on MFC can form a set of infected cascade trees. We show how to extract such trees later in this section.

Definition 10.12 (Infected Cascade Tree) The *signed infected cascade tree* summarizes the state of the information propagation and user activation process in the network. Let $T = (\mathcal{V}_T, \mathcal{E}_T, s, w)$ be a *signed infected cascade tree*. The node set $\mathcal{V}_T \subseteq \mathcal{V}_D$ consists of all the infected users in the tree and the directed activation link $(u, v) \in \mathcal{E}_T \subseteq \mathcal{E}_D$ if and only if u succeeds in activating v.

The signed infected cascade trees can be inferred from the infected network, and we propose to extract the trees capturing the most information (i.e., the most likely trees) for each connected component. Let $C_i = (\mathcal{V}_{C_i}, \mathcal{E}_{C_i}, s, w)$ be a detected connected component consisting of multiple infected cascade trees, and let $T = (\mathcal{V}_T, \mathcal{E}_T, s, w)$ be one of the trees extracted from C_i, where $\mathcal{V}_T \subseteq \mathcal{V}_{C_i}$ and $\mathcal{V}_T \subseteq \mathcal{V}_{C_i}$. The likelihood of tree T is $\mathcal{L}(T) = \prod_{(u,v)\in\mathcal{E}_T} w(u, v)$. Furthermore, the *optimal* infected cascade tree T^* in C_i can be defined as:

$$T^* = \arg\max_{T \in \mathcal{T}} \mathcal{L}(T), \tag{10.43}$$

where $\mathcal{T}$ denotes the set of all potential trees that can be detected from component C_i. The maximum likelihood *infected cascade trees* can be extracted using the Chu-Liu/Edmonds' algorithm [14, 20] from the directed connected components. The pseudo-code of the *infected cascade trees* extraction method is available in Algorithm 8, which will call the functions in Algorithms 6 and 7 to get the maximum weight spanning graphs and resolve the circles in the graph.

10.5.4.3 Rumor Initiator Inference

Based on the methods introduced in the previous sections, we are able to detect a set of diffusion trees from the network, the roots of which without incoming edges represent the rumor initiators. Meanwhile, besides the roots, multiple *rumor initiators* can co-exist in one *infected cascade tree*. In other words, the number of extracted diffusion trees is a lower bound on the number of rumor

Algorithm 6 Maximum weight spanning graph (MWSG)

```
Require: Graph $G = (\mathcal{V}, \mathcal{E}, s, w)$
Ensure: Maximum weight spanning graph $G' = (\mathcal{N}, \mathcal{L}, w)$
initialize node set $\mathcal{N} = \emptyset$, link set $\mathcal{L} = \emptyset$
for $u \in \mathcal{V} \setminus \mathcal{N}$ do
    $\mathcal{N} = \mathcal{N} \cup \{u\}$
    find edge $e = \arg\max_{e \in \mathcal{E}} w(e)$
    $\mathcal{L} = \mathcal{L} \cup \{e\}$
end for
```

Algorithm 7 Contract circles (CC)

```
Require: Graph containing circles $G = (\mathcal{N}, \mathcal{L}, w)$
Ensure: Contracted graph without circles $G' = (\mathcal{N}', \mathcal{L}', w')$
$\mathcal{L}' = \emptyset$ and new link weight mapping $w'$
for each circle $O = (\mathcal{N}_O, \mathcal{L}_O)$ in $(\mathcal{N}, \mathcal{L})$ do
    contract all nodes in $O$ into a pseudo-node $u_o$
    for each link $(u_x, u_y) \in \mathcal{L}$ do
        if $u_x \notin \mathcal{N}_O$ and $u_y \in \mathcal{N}_O$ then
            $\mathcal{L}' = \mathcal{L}' \cup \{(u_x, u_o)\}$
            $w'(u_x, u_o) = w(u_x, u_y) - w(\pi(u_y), u_y)$, where $(\pi(u_y), u_y) \in \mathcal{L}$ is the link with the maximum weight linked to $u_y$
        else
            if $u_x \in \mathcal{N}_O$ and $u_y \notin \mathcal{N}_O$ then
                $\mathcal{L}' = \mathcal{L}' \cup \{(u_o, u_y)\}$
                $w'(u_o, u_y) = w(u_x, u_y)$
            else
                $\mathcal{L}' = \mathcal{L}' \cup \{(u_x, u_y)\}$
                $w'(u_x, u_y) = w(u_x, u_y)$
            end if
        end if
    end for
end for
```

Algorithm 8 Infected cascade trees extraction

```
Require: infected connected component set $\mathcal{C}$
Ensure: infected cascade tree set $\mathcal{T}$
initialize tree set $\mathcal{T} = \emptyset$
for component $C_i = (\mathcal{V}_{C_i}, \mathcal{E}_{C_i}, s_{C_i}, w_{C_i}) \in \mathcal{C}$ do
    $(\mathcal{N}, \mathcal{L}, w) = \text{MWSG}(C_i)$
    if $(\mathcal{N}, \mathcal{L}, w)$ contains circles $\mathcal{O}$ then
        $(\mathcal{N}', \mathcal{L}', w') = \text{CC}(\mathcal{N}, \mathcal{L}, w_{C_i})$
        $(\mathcal{N}', \mathcal{L}', w') = \text{MWSG}((\mathcal{N}', \mathcal{L}', w'))$
    end if
    for circle $O \in \mathcal{O}$ do
        for link $(u_x, u_o) \in \mathcal{L}'$ do
            get the corresponding link $(u_x, u_y)$, where $u_y$ is in the circle
            remove link $(\pi(u_y), u_y)$ from $\mathcal{L}$ to break the circle $O$
        end for
    end for
    $\mathcal{T} = \mathcal{T} \cup \{(\mathcal{N}, \mathcal{L})\}$
end for
```

initiators. The detected cascade tree can actually be partitioned into several isolated sub-trees instead. The roots of these sub-trees provide additional candidates for being rumor initiators. Such a partitioning process can be achieved with the algorithm introduced in Sect. 10.5.3 effectively. However, the extracted *infected cascade trees* from the infected signed network may not necessarily be binary trees, and this can be very complex to deal with [35]. Next, we propose to transform each cascade tree into a binary tree first and then identify the optimal rumor initiators.

To transform a general tree into a binary tree without distorting information about the relative influence relationships, we propose to add extra dummy nodes to the trees, which have no effect on information diffusion, and they cannot be selected as *rumor initiators*.

Example 10.2 For example, in Fig. 10.3, the tree in the left figure is not a binary tree, where the root node has three children nodes. To transform it into a binary tree, between the root and its children, $\lceil \log_2 3 \rceil$ extra nodes are added to the tree as the root's new children and the root's children nodes are

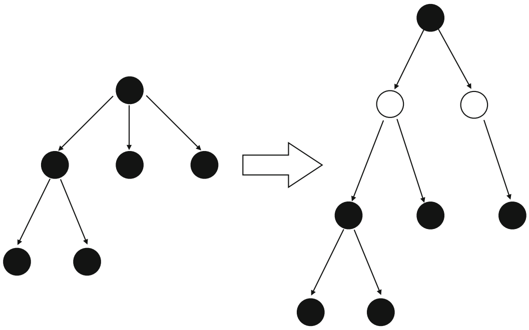


Fig. 10.3 Example of the binary tree transformation

assigned as the new nodes' children. These newly added nodes will not participate in the information diffusion and they cannot be selected as *rumor initiators*.

Meanwhile, to avoid the case of having too many *rumor initiators* (e.g., every user in the component is a *rumor initiators*), we will add a penalty term to constrain the number of detected rumor initiators. For each tree $T \in \mathcal{T}$ rooted at u, we can represent the optimal rumor initiators $\mathcal{I}^*$ of size k^* with initial state $\mathcal{S}^*$ as follows:

$$k^*, \mathcal{I}^*, \mathcal{S}^* = \arg \min_{k,\mathcal{I},\mathcal{S}} -\mathbf{OPT}(u, \mathcal{I}, \mathcal{S}, k) + (k-1) \cdot \beta, \tag{10.44}$$

Here, parameter β denotes the penalty of each introduced rumor initiator and term $(k-1)$ represents the extra initiators detected besides the original root of tree T. Function $\mathbf{OPT}(u, \mathcal{I}, \mathcal{S}, k)$ can be computed with the dynamic programming based method introduced in the previous section. By enumerating k from 1 to the number of nodes in T (i.e., $|\mathcal{V}_T|$), we are able to obtain the optimal solution of the above objective function. However, such a process can be very time consuming. To balance between the time cost and quality of the result, we propose to increase k from 1 to $|\mathcal{V}_T|$ and stop once the increase in k cannot lead to increase in the objective function.

10.6 Summary

In this chapter, we talked about the viral marketing problem and provided an introduction to several viral marketing algorithms for seed user selection based on various learning settings. The viral marketing problem is formulated as an optimization problem, which aims at selecting the optimal seed user set, who can maximize the impact in the information diffusion process.

Based on the classic information diffusion models, like LT and IC, we introduced two viral marketing algorithms, greedy and CELF, to pick the seed users. The greedy algorithms will select the seed user with multiple rounds, where the users who can introduce the maximum influence gain will be added into the seed user set in each round. To resolve the high time cost problem, CELF uses a heap data structure to keep record of the users' influence, and the heap structure will be updated dynamically in the seed user selection process. To further lower down the time complexity, we also introduced two algorithms based on heuristics, where one is based on the node centrality and the other one is based on degree discount, respectively.

In the multi-product information diffusion setting, the products may have intertwined relationships with each other, including independent, competing, and complimentary, respectively. Depending on the promotion orders of the other products and the target product, we categorized the viral marketing problem in such a learning setting into the C-TIM and J-TIM, respectively. To resolve the problem, the C-TIER and J-TIER algorithms were introduced in this chapter.

To study the viral marketing problem across multiple aligned social networks, we introduced the M&M diffusion model. Based on a set of meta paths, M&M defines the multi-relational network with the meta path concept, based on which the viral marketing problem is addressed with a greedy algorithm. Meanwhile, based on the IPATH diffusion model, we introduced a dynamic programming-based seed user selection algorithm named IMDP.

At the end of this chapter, we talked about the rumor initiator detection problem. Given an infected network, the rumor initiator detection problem aims at identifying the rumor initiators IDs, numbers, and initial status concurrently. By assuming that the rumor diffusion process is based on the MFC model, to address the problem, we introduced a multi-phase rumor initiator detection algorithm, which partitions the infected network into several components and further identifies the rumor initiators from them with a dynamic program algorithm.

10.7 Bibliography Notes

Viral marketing (i.e., influence maximization) problem in customer networks first proposed by Domingos et al. [19] has been a hot research topic. Richardson et al. [42] study the viral marketing based on knowledge-sharing sites and propose a new model which needs less computational cost than the model proposed in [19]. Kempe et al. propose to study the influence maximization problem through a social network [29] and propose two different diffusion models: independent cascade (IC) model and linear threshold (LT) model, which have been widely used in later influence maximization papers. Zhan et al. propose to extend the traditional single-network viral marketing problem to multiple aligned networks in [47].

Meanwhile, the promotions of multiple products can exist in social networks simultaneously, which can be independent, competing, or complementary. Datta et al. [17] study the viral marketing for multiple independent products at the same time and aim at selecting seed users for each products to maximize the overall influence. Bharathi et al. [4] propose to study the competitive influence maximization in social networks, where multiple competing products are to be promoted. He et al. [25] propose to study the influence blocking maximization problem in social networks with the competitive linear threshold model. Chen et al. [13] study the influence maximization in social networks when negative opinions can emerge and propagate. Multiple threshold models for competitive influence in social networks are proposed in [7], whose submodularity and monotonicity are studied in details. Meanwhile, Narayanam et al. [38] study the viral marketing for product cross-sell through social networks to maximize the revenue, where products can have promotion cost, benefits, and promotion budgets. Lu et al. [37] study the influence propagation and maximization problem in the setting from competition to complementarity.

Among these works on information diffusion and viral marketing problems, rumor propagation in online social networks is of practical importance. Kwon et al. identify characteristics of rumors by examining temporal, structural, and linguistic aspects of rumors [33]. Rumors can spread very fast in online social networks, and Doerr et al. propose to study the structural and algorithmic properties of networks which accelerate such a propagation in [18]. To maximize the influence or rumors, the diffusion of competing rumors in social networks is studied in [32].

Influence source identification in unsigned networks has been studied in the existing works. Lappas et al. [35] propose the problem of finding effectors in social networks. In [35], the k-effectors problem is formally defined and the time complexity of the problem for different types of graphs is analyzed in detail. Shah et al. study similar problems in [44] to infer the sources of a rumor in a network, where a SIR-based rumor diffusion model is introduced. They propose to detect the rumor sources by identifying users with high "*rumor centrality*," which is also used in their computer virus sources discovery [43]. Prakash et al. propose to study the culprits in epidemics in [41]. The underlying structure of cascades in social networks is studied in [53].

10.8 Exercises

1. (Easy) Please try to complete the proof of the Theorem 10.8 we introduced in Sect. 10.4.1.2.
2. (Easy) Please try to complete the proof of the Theorem 10.9 we introduced in Sect. 10.4.1.2.
3. (Medium) Please try to prove Theorem 10.1 we introduced in Sect. 10.3.1.1.
4. (Medium) Please try to implement the *greedy* algorithm we introduced in Sect. 10.2.2.1 based on the LT model with a preferred programming language, and test its performance with some simulations on a toy network data set.
5. (Medium) Please try to implement the *greedy* algorithm we introduced in Sect. 10.2.2.1 based on the IC model with a preferred programming language, and test its performance with some simulations on a toy network data set.
6. (Medium) Please refer to [29], and try to prove that for the LT and IC diffusion models, their influence function is *monotone*.
7. (Medium) Please refer to [29], and try to prove that for the LT and IC diffusion models, their influence function is *submodular*.
8. (Hard) Based on the LT model, if adding each user into the seed user set will bring about a certain cost, please try to consider to extend and improve the *step-wise greedy influence maximization* algorithm for such a scenario.
9. (Hard) Please try to implement the *CELF* algorithm with a preferred programming language, and compare its efficiency with the *greedy* algorithm.
10. (Hard) Please try to implement the dynamic programing algorithm introduced in Sect. 10.5.3.

References

1. L. Adamic, R. Lukose, A. Puniyani, B. Huberman, Search in power-law networks. CoRR, cs.NI/0103016 (2001)
2. G. Allport, L. Postman, *The Psychology of Rumor* (Henry Holt, New York, 1947)
3. V. Belak, S. Lam, C. Hayes, Towards maximising cross-community information diffusion, in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (2012)
4. S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in *International Workshop on Web and Internet Economics* (2007)
5. S. Borgatti, M. Everett, A graph-theoretic perspective on centrality. Soc. Netw. **28**(4), 466–484 (2006)
6. C. Borgs, J. Chayes, N. Immorlica, A. Kalai, V. Mirrokni, C. Papadimitriou, The myth of the folk theorem. Games Econ. Behav. **70**, 34–43 (2010)
7. A. Borodin, Y. Filmus, J. Oren, Threshold models for competitive influence in social networks, in *International Workshop on Internet and Network Economics* (2010)
8. S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in *Proceedings of the Seventh International Conference on World Wide Web 7 (WWW7)* (1998)
9. X. Chen, X. Deng, S. Teng, Computing Nash equilibria: Approximation and smoothed complexity, in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* (2006)
10. X. Chen, S. Teng, P. Valiant, The approximation complexity of win-lose games, in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '07)* (2007)
11. 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 (KDD '09)* (2009)
12. 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 (KDD '10)* (2010)

13. W. Chen, A. Collins, R. Cummings et al., Influence maximization in social networks when negative opinions may emerge and propagate – Microsoft research, in *Conference: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011* (2011)
14. Y. Chu, T. Liu, On the shortest arborescence of a directed graph. Sci. Sin. **14**, 1396–1400 (1965)
15. T. Cormen, C. Stein, R. Rivest, C. Leiserson, *Introduction to Algorithms*, 2nd edn. (McGraw-Hill Higher Education, New York, 2001)
16. C. Daskalakis, P. Goldberg, C. Papadimitriou, The complexity of computing a Nash equilibrium, in *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC '06)* (2006)
17. S. Datta, A. Majumder, N. Shrivastava, Viral marketing for multiple products, in *2010 IEEE International Conference on Data Mining* (2010)
18. B. Doerr, M. Fouz, T. Friedrich, Why rumors spread so quickly in social networks. Commun. ACM **55**, 70–75 (2012)
19. P. Domingos, M. Richardson, Mining the network value of customers, in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)* (2001)
20. J. Edmonds, Optimum branchings. J. Res. Natl. Bur. Stand. **71**, 233–240 (1967)
21. U. Feige. A threshold of ln n for approximating set cover. J. ACM **45**, 634–652 (1998)
22. A. Goyal, W. Lu, L. Lakshmanan, Celf++: optimizing the greedy algorithm for influence maximization in social networks, in *Proceedings of the 20th International Conference Companion on World Wide Web (WWW '11)* (2011)
23. A. Goyal, W. Lu, L. Lakshmanan, Simpath: an efficient algorithm for influence maximization under the linear threshold model, in *2011 IEEE 11th International Conference on Data Mining* (2011)
24. H. Gui, Y. Sun, J. Han, G. Brova, Modeling topic diffusion in multi-relational bibliographic information networks, in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM '14)* (2014)
25. X. He, G. Song, W. Chen, Q. Jiang, Influence blocking maximization in social networks under the competitive linear threshold model, in *Conference: Proceedings of SIAM International Conference on Data Mining, SDM 2012* (2012)
26. P. Jaccard, Étude comparative de la distribution florale dans une portion des alpes et des jura. Bull. Soc. Vaud. Sci. Nat. **37**, 547–579 (1901)
27. Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, in *Twenty-Fifth AAAI Conference on Artificial Intelligence* (2011)
28. F. Jin, E. Dougherty, P. Saraf, Y. Cao, N. Ramakrishnan, Epidemiological modeling of news and rumors on twitter, in *Proceedings of the 7th Workshop on Social Network Mining and Analysis (SNAKDD '13)* (2013)
29. D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2003)
30. M. Kimura, K. Saito, Approximate solutions for the influence maximization problem in a social network, in *Knowledge-Based Intelligent Information and Engineering Systems*, ed. by B. Gabrys, R. Howlett, L. Jain (Springer, Berlin, 2006)
31. X. Kong, J. Zhang, P. Yu, Inferring anchor links across multiple heterogeneous social networks, in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13)* (2013)
32. J. Kostka, Y. Oswald, R. Wattenhofer, Word of mouth: rumor dissemination in social networks, in *International Colloquium on Structural Information and Communication Complexity* (2008)
33. S. Kwon, M. Cha, K. Jung, W. Chen, Y. Wang, Prominent features of rumor propagation in online social media, in *IEEE 13th International Conference on Data Mining* (2013)
34. K. Lai, The knapsack problem and fully polynomial time approximation schemes (fptas). Technical report (2006)
35. T. Lappas, E. Terzi, D. Gunopulos, H. Mannila, Finding effectors in social networks, in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)* (2010)
36. J. Leskovec, L. Adamic, B. Huberman, The dynamics of viral marketing. ACM Trans. Web **1**(1), 5 (2007)
37. W. Lu, W. Chen, L. Lakshmanan, From competition to complementarity: comparative influence diffusion and maximization, in *Proceedings of VLDB Endowment* (2015)
38. R. Narayanam, A. Nanavati, Viral marketing for product cross-sell through social networks, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2012)* (2012)
39. D. Nguyen, H. Zhang, S. Das, M. Thai, T. Dinh, Least cost influence in multiplex social networks: model representation and analysis, in *2013 IEEE 13th International Conference on Data Mining* (2013)
40. N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani, *Algorithmic Game Theory* (Cambridge University Press, New York, 2007)
41. B. Prakash, J. Vreeken, C. Faloutsos, Spotting culprits in epidemics: how many and which ones? in *2012 IEEE 12th International Conference on Data Mining* (2012)
42. M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002)

43. D. Shah, T. Zaman, Detecting sources of computer viruses in networks: theory and experiment, in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '10*, New York (2010), pp. 203–214. http://doi.acm.org/10.1145/1811039.1811063
44. D. Shah, T. Zaman, Rumors in a network: who's the culprit? IEEE Trans. Inf. Theory **57**(8), 5163–5181 (2011)
45. Y. Sun, J. Han, X. Yan, P. Yu, T. Wu, Pathsim: meta path-based top-k similarity search in heterogeneous information networks, in *Proceedings of VLDB Endowment* (2011)
46. F. Yang, Y. Liu, X. Yu, M. Yang, Automatic detection of rumor on Sina Weibo, in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS '12)* (2012)
47. Q. Zhan, J. Zhang, S. Wang, P. Yu, J. Xie, Influence maximization across partially aligned heterogeneous social networks, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2015)
48. Q. Zhan, J. Zhang, P. Yu, J. Xie, Viral marketing through aligned networks. Technical report (2018)
49. J. Zhang, P. Yu, Z. Zhou, Meta-path based multi-network collective link prediction, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)* (2014)
50. J. Zhang, S. Wang, Q. Zhan, P. Yu, Intertwined viral marketing in social networks, in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2016)
51. J. Zhang, C. Aggarwal, P. Yu, Rumor initiator detection in infected signed networks, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (2017)
52. J. Zhang, L. Cui, Y. Fu, F. Gouza, Fake news detection with deep diffusive network model. CoRR, abs/1805.08751 (2018)
53. B. Zong, Y. Wu, A. Singh, X. Yan, Inferring the underlying structure of information cascades, in *12th IEEE International Conference on Data Mining (ICDM 2012)* (2012)