Maximizing Social Influence for the Awareness Threshold Model

Haiqi Sun[1] Reynold Cheng[1] Xiaokui Xiao[2] Jing Yan[1] Yudian Zheng[1] Yuqiu Qian[1]

[1] The University of Hong Kong
{hqsun,ckcheng,jyan,ydzheng2,yqqian}@cs.hku.hk
[2] Nanyang Technological University xkxiao@ntu.edu.sg

Abstract Given a social network G, the *Influence Maximization* (IM) problem aims to find a *seed set* $S \subseteq G$ of k users. These users are advertised, or *activated*, through marketing campaigns, with the hope that they will continue to influence others in G (e.g., by spreading messages about a new book). The goal of IM is to find the set S that achieves an optimal advertising effect or *expected spread* (e.g., make the largest number of users in G know about the book).
Existing IM solutions make extensive use of *propagation models*, such as Linear Threshold (LT) or the *Independent Cascade* (IC). These models define the *activation probability*, or the chance that a user successfully gets activated by his/her neighbors in G. Although these models are well-studied, they overlook the fact that a user's influence on others decreases with time. This can lead to an over-estimation of activation probabilities, as well as the expected spread.
To address the drawbacks of LT and IC, we develop a new propagation model, called *Awareness Threshold* (or AT), which considers the fact that a user's influence decays with time. We further study the *Scheduled Influence Maximization* (or SIM), to find out the set S of users to activate, as well as *when* they should be activated. The SIM problem considers the time-decaying nature of influence based on the AT model. We show that the problem is NP-hard, and we develop three approximation solutions with accuracy guarantees. Extensive experiments on real social networks show that (1) AT yields a more accurate estimation of activation probability; and (2) Solutions to the SIM gives a better expected spread than IM algorithms on the AT model.

1 Introduction

The *Influence maximization (IM)* problem, first proposed by Kempe et al. [22], has been extensively studied in recent years (e.g., [27,18,37,26,36,9,28,29,6]). The IM plays a fundamental role in *viral marketing*, a business promotion strategy that employs the *word-of-mouth* effect, where the advertisement is based on customers spreading news about something (e.g., a new electronic product) in social networks. The main goal of IM is to find, given an *influence graph* G, a *seed set* S of k users such that the number of users in G influenced by S (or *expected spread*) is maximized.

Figure 1 shows an influence graph, which is derived from the social relationship among users. Each node in the graph represents a social network user. The value of each edge is the *influence probability*, i.e., the chance that a user is affected or *activated* by another one. Suppose that $k = 2$, and a seed set $S = \{A, B\}$ is chosen by an IM solution. A company, which wants to advertise a product, can promote it to A and B (e.g., by giving them discounts), hoping that the expected spread (i.e., the number of people influenced by A and B to buy its product) is maximal.

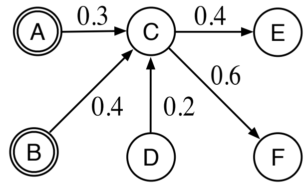


Figure 1. Illustrating the IM problem

To enable IM, *propagation models* have been used to govern the flow of influence in G. For instance, in *Linear Threshold (LT)* [23], a node $v \in G$ is activated, if and only if the sum of the influence probabilities induced by v's active neighbors on v exceed some threshold T. In Figure 1, suppose that $T = 0.5$, and A and B have been previously activated (e.g., through a company's marketing campaign). Node C then receives an activation probability of 0.3+0.4=0.7 from A and B and gets activated. In another well-studied model, *Independent Cascade (IC)*, C is activated by A and B with a probability of $1 - (1 - 0.3) \times (1 - 0.4)$, or 0.58. Once C becomes active, we can then use the propagation model again to activate C's neighbors (i.e., E and F). Based on these models, efficient IM algorithms have been designed to find the best set S of seed nodes that yield the highest expected spread (or the number of nodes activated) [27,19,12,37,36].

The main problem of *IC* and *LT* is that they overlook the fact that the influence of a user received from other nodes *decreases* with time. For example, in Figure 1, the influence probabilities on C due to A is not always 0.3 as claimed by IC and LT; rather, it gradually drops as time goes by. Once A has influenced C, A's effect on C becomes less profound, possibly because C's memory or awareness of what A told him/her fades. In our experiments performed on social networks (*Twitter* and *Digg*), the average activation probability of a node, which is a function of influence probabilities, drops quickly with time. However, IC and LT do not consider this factor, and subsequently overestimate the expected spread significantly. This calls for a better propagation model.

Awareness Threshold. To deal with the issues of decaying influence probabilities, we study the *awareness threshold* (or *AT* in short). This propagation model is inspired by *brand awareness*, suggesting that a person u's impression about the brand of a product is affected by two factors: (1) *build*, which means u's awareness about a product is increased through exposure to more marketing activities [8,13]; and (2) *decay*, which says that u's awareness drops exponentially because he/she receives no further advertisement about the product [4,7,16,13]. The AT model integrates brand awareness into the influence propagation process. Compared to IC and LT, AT attains a more accurate estimation of spread in our experiments.

Influence Maximization under AT. We further study the IM problem under the AT model. We show that this problem is NP-hard. We thus develop an efficient solution with accuracy guarantees on the expected spread. We observe that the expected spread can be further improved through a *scheduled* activation of seed nodes. To explain, in the IM problem, the set S of k nodes is activated at the same time. However, it may be wiser by activating these nodes at different time instants, taking into account their brand awareness. Hence, in addition to deciding the seed nodes, we also want to determine the exact time that they are activated. We term this variant of the IM problem *scheduled influence maximization* (or *SIM*). Because SIM is NP-hard, we design a fast approximate solution, called the *Two-Phase Search* (TPS). Under the TPS framework, we study three scheduling policies, namely the *Breadth First Schedule*, the *Depth First*

Table 1. Frequently used notations

Notation	*Description*
$w_{u,v}$	the propagation probability defined on edge $e_{u,v}$
$N_v^{(t)}$	active in-degree neighbor set of user v at step t
θ_v	activation threshold of user v
$A_v^{(t)}$	awareness level of user v at step t
S/U	seed set / scheduled seed set
k	number of seeds
$t_{u,v}$	the time at which u attempts to activate v
$t^{(v)}$	the activation time of user v
$t^{(u,v)}$	time delay defined on the edge $e_{u,v}$
$\sigma_m(S)/\sigma_m(U)$	the expected spread of S/U under model m

Schedule, and the *Bucket Schedule*. Our results on real datasets show that SIM solutions yield higher spread than their IM algorithms.

The rest of the paper is as follows. In Section 2, we discuss the research background. Section 3 discusses the limitations of existing propagation models. In Section 4, we describe the AT model. We then present our IM and SIM solutions for AT in Sections 5 and 6 respectively. We discuss the experiment results in Section 7, and related works in Section 8. Section 9 concludes.

2 Preliminaries

2.1 Problem Definition

Given a directed graph $G = \{V, E, W\}$, V is the user set, and E is the set of edges (e.g., the user influence relation). W is the set of probabilisties associated on E, where for each $e_{u,v} \in E$, the weight $w_{u,v} \in [0, 1]$ defined on the edge captures the probability that u will influence v. Under a fixed budget k and a propagation model *m* (e.g. LT), the *IM* problem looks for a seed set S consisting of no more than k users, whose *expected spread* is the largest. Its definition is $S^* = \text{argmax}_{S \subseteq V \& |S| \leq k}\{\sigma_m(S)\}$, where influence function $\sigma_m(S)$ outputs the number of active users under model m, when each node in S is activated. To address the *IM* problem, it is important to model how the information is propagated, as will be shown next.

2.2 Propagation Models

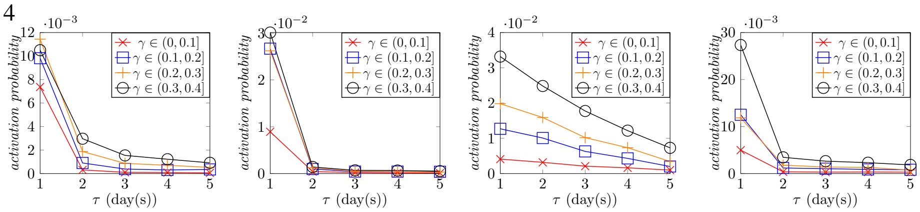
There are two major classic propagation models, namely *LT* and *IC* model [23]. In both models, information (or *influence*) spreads from the seed set to other nodes as discrete time steps unfold.

***LT* model.** Initially, all seed users are initialized as active and others stay inactive. At each discrete time step, each inactive user will be activated if the weighted sum contributed by her active neighbors is larger than a pre-defined threshold. Once a user is activated, the user will stay active. The influence propagation terminates when no more activations are possible.

***IC* model.** Here the propagation process is also triggered by a set of active seed users. At each time step, the newly-activated user (i.e., activated at the previous step), influences each of her inactive neighbors with a pre-defined probability. If the activation is successful, then the inactive neighbor(s) will be activated and remain active in further steps. The influence propagation terminates when no more users can be activated.

2.3 Activation Probability

Under a specific propagation model, at time step t, given an inactive user v and her active neighbor set $N_v^{(t)}$, activation probability of v is defined as the likelihood that v can be activated by $N_v^{(t)}$. Specifically, in *LT* and *IC*, the activation probability values of v are denoted by $\sum_{u \in N_v^{(t)}} w_{u,v}$ and $1 - \prod_{u \in N_v^{(t)}}(1 - w_{u,v})$ respectively.



(a) Digg ($\gamma = \sum w_{u,v}$) (b) Digg ($\gamma = 1 - \prod(1 - w_{u,v})$) (c) Twitter ($\gamma = \sum w_{u,v}$) (d) Twitter ($\gamma = 1 - \prod(1 - w_{u,v})$)

Figure 2. Activation Probability

3 Limitations of IC and LT

As discussed in Section 2.3, *IC* and *LT* both adopt the following assumption: *At time t, for a given inactive user v and her active neighbor set $N_v^{(t)}$, the activation probability of v estimated by* LT *(*IC*) model is computed based on the influence weights associated with his/her incoming edges..*

However, as will be shown in the following case studies on real datasets, the assumption is inconsistent with the reality. Suppose that each active neighbor $u \in N_v^{(t)}$ always attempts to activate v. Then at step t, we define the average timespan, denoted by τ, as $\tau = \frac{1}{|N_v^{(t)}|} \sum_{u \in N_v^{(t)}} (t - t_{u,v})$. This is the average time that v is reactivated by u.

Next we conduct experiments to show the relationship between activation probability and the average timespan τ.

Datasets. Two real-world datasets, *Digg* and *Twitter* [1], are used in this experiment. *Digg* and *Twitter* are both social network applications, where *Digg* serves as a platform for sharing and voting stories, and *Twitter* provides users with chances of sharing news, photos, etc. The data from these two sources both consists of two parts, namely graph and *actionlog*. The graph contains social links among users, and *actionlog* records voting/tweeting history of users in the form of <user_id, action_id, action_time>, e.g. in *Twitter*, <001, 002, 1250772872> means that user *001* re-tweeted an URL (numbered as 002) on the date *1250772872* (Unix Timestamps). In [18], the propagation procedure is defined by action follow. For instance, in *Digg*, if user u votes "Cinderella", and later on her follower v also votes it, we consider the action of voting "Cinderella" has propagated from u to v. Table 2 shows the statistics.

Parameter Setting. We set the parameters of Twitter and Digg as follows:
(1) All probabilistic weights assigned on social links are learned from *actionlog* data by the methods proposed in [17] and settings are referred to [18].
(2) Due to the sparsity of our datasets, not all action seeds, i.e. first voters of stories or initial posters of tweets, are included in the social graph. Adopting the strategy in [18], we take the users who first vote stories (post the tweets) among their friends as seeds.

Observations. To assess *LT* and *IC* model, we define the *accumulated influence*, denoted by γ, as respectively $\gamma = \sum w_{u,v}$ and $\gamma = 1 - \prod(1 - w_{u,v})$. Then conditioned on fixed γ and increasing τ, values of *activation probability* are shown in Fig. 2. Specifically, accumulated influence γ is fixed within an interval $(x, x + 0.1]$, and for each γ, average timespan τ is varied from 1 day to 5 days. In experiments, x is set from 0 to 0.9. Due to space constraints, only some representative results are shown here.

Observe that in both of *Digg* and *Twitter*, for each settled τ, with greater γ issued, the activation probability reaches a higher level, which is consistent with the theoretical result concluded by classic models (Section. 2.3). However, for each γ, the activation

Table 2. Statistics of Datasets

	Digg	*Twitter*	*NetHEPT*	*DBLP*
# Nodes	71K	736K	15K	914K
# Edges	1.7M	36M	62K	6.6M
Avg. Degree	24	50	4.1	7.2
# Actionlogs	3M	2.8M	-	-

probability decreases with an increasing τ, which demonstrates that even if the accumulated influence is fixed, activation probability decays when the average timespan grows, however, this is not observed by classic models.

In particular, the activation probability can be overestimated by *LT* and *IC* models. During a specific marketing campaign, these classic models will give an over-prediction on the expected spread.

We next present the awareness threshold (AT) model, which addresses the drawback of the LT and IC.

4 Awareness Threshold (AT)

Brand awareness is a term in the field of marketing science, which quantifies the extent to which a brand is recognized by potential users, and is the primary goal of advertising. Awareness level of user v at time t, denoted by $A_v^{(t)}$, reflects the likelihood of her adoption behaviors [15,25]. By measuring user awareness level, business agents are able to predict their marketing achievement, i.e., the number of users who will adopt specific products.

The awareness has two dimensions, *build* and *decay*, namely as below.

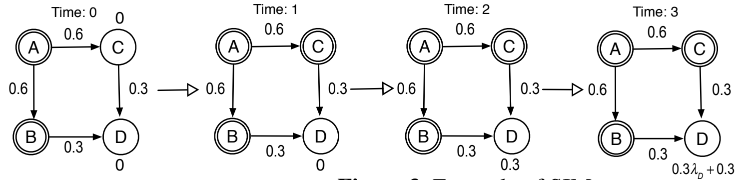
Awareness Build. Viral marketing attempts to spread brand content over the whole social network. When marketing campaign is launched in social networks, advertising messages are broadcast through the social connection between user pairs. Some individuals are in the circles which are full of brand information, they are thus exposed to advertising messages adequately. However, other social groups may only touch these information to a limited extent. User awareness level grows as increasing copies of advertising are exposed to them. With higher awareness level in minds, users tend to adopt the brand (or get activated in the setting of propagation models) [4,7].

Decay Effect. In the absence of further advertising exposures, customer awareness will decline and eventually decay to negligible levels [4,7,16,13]. One common function to model the decay effect is assuming the awareness will decrease exponentially w.r.t time. Formally, for a specific user v, her awareness decay can be mathematically modeled as follows:

$$A_v^{(t)} = \Delta A_v^{(t)} + \lambda_v \cdot\ A_v^{(t-1)}, \qquad \lambda_v \in (0,1) \tag{1}$$

where $A_v^{(t)}$ and $A_v^{(t-1)}$ represents her awareness level at step t and $t-1$, and $\Delta A_v^{(t)}$ is the awareness increment contributed by new advertising exposure.

Our Model. To incorporate the above two effects, we build our Awareness Threshold (*AT*) model. Like *IC* and *LT* models, *AT* model simulates social network as a directed graph $G = \{V, E, W, T, \Lambda\}$, and each user is in the status of active (product adopter) or inactive. Meanwhile, different from *IC* and *LT* models, in *AT* model, there are two additional sets ($T \subseteq \mathbb{Z}^+$ and Λ) assigned to the graph G. T records time delays attached to social links. Since information diffusion progress among friend network will not be finished in a moment, $t^{(u,v)} \in T$ implies that it takes $t^{(u,v)}$ time steps for messages to propagate from u to v by the edge $e_{u,v}$. Many researches [30,10] have been conducted on how to calculate T. Moreover, for each user v, Λ contains her decay factor $\lambda_v \in (0, 1)$.



Threshold	Time Delay
$\theta_A = 0.4$	$t_{A,B} = 1$
$\theta_B = 0.5$	$t_{A,C} = 1$
$\theta_C = 0.4$	$t_{B,D} = 2$
$\theta_D = 0.6$	$t_{C,D} = 2$

Figure 3. Example of SIM.

We denote $t^{(v)}$ as the time at which v gets activated. (If v stays inactive, $t^{(v)} = +\infty$.) Let $\mathbb{1}_{\{.\}}$ denote an indicator function which returns 1 if its argument is true; 0, otherwise. For example, $\mathbb{1}_{\{5=2\}} = 0$ and $\mathbb{1}_{\{5=5\}} = 1$. When time $t \geq 1$, awareness score of inactive user v, denoted by $A_v^{(t)}$, is given by Eq. 1. Specifically, in *AT* model, $\Delta A_v^{(t)}$ is defined as follows:

$$\Delta A_v^{(t)} = \sum\nolimits_{u \in N_v^{(t)}} w_{u,v} \cdot \mathbb{1}_{\{t^{(u)}+t^{(u,v)}=t\}}.$$

The term $t^{(u)} + t^{(u,v)} = t$ indicates that v receives the influence forwarded by u at t. The threshold of v, notated by θ_v, controls her activation condition, and v will be activated only if her awareness score reaches θ_v.

Intuitively, the update of $A_v^{(t)}$ contains two parts: (1) $\Delta A_v^{(t)}$: the awareness increments brought by active neighbors of v; (2) $\lambda_v \cdot A_v^{(t-1)}$: the awareness remained from last step.

Formally speaking, we define the awareness score of inactive user v at time t as follows:

Definition 1 (Awareness Score) *For the time t, awareness score of inactive user $v \in V$, i.e. $A_v^{(t)}$, is defined as*

$$\lambda_v \cdot A_v^{(t-1)} + \sum\nolimits_{u \in N_v^{(t)}} w_{u,v} \cdot \mathbb{1}_{\{t^{(u)}+t^{(u,v)}=t\}} \tag{2}$$

if $t \geq 1$; and $A_v^{(t)} = 0$ if $t = 0$.

In *AT* model, if all seeds ($\forall s \in S$) are activated initially, starting from them, information spreads as time goes in discrete steps:

(1) $t = 0$, $\forall s \in S$, s is active. $\forall v \in V - S$, v is inactive.

(2) $t \geq 1$, the information is broadcast by every newly-activated user u, i.e. $t^{(u)} = t-1$, and propagated to each inactive neighbor v:

- At $t^{(u)} + t^{(u,v)}$, v will capture an awareness increment $w_{u,v}$.
- If $A_v^{(t)} \geq \theta_v$, v is activated.
- Meanwhile, awareness score of every inactive individual j is weakened by her decay factor λ_j, as formalized by Eq. 2.

(3) It terminates when no further activation is possible.

Fig. 3 illustrates this process with an example. Noting that in Fig. 3, the double circle in a node indicates that the user is active, and the value next to an inactive user represents her awareness score at a specific time.

5 Influence Maximization

In this section, we will first prove that under *AT* model, the NP-hardness of *IM* problem still holds, and then provide solution towards *IM* problem based on the *AT* model.

Theorem 1 *Under awareness threshold model, the influence maximization problem is NP-hard.*

Proof. In *AT* model, when there is no decay inside user awareness, i.e. for each user v, $\lambda_v = 1$, then solving influence maximization problem is exactly equivalent to addressing it under *LT* model whose **NP**-hardness is proved in [23]. Thus *AT*-based influence maximization can be proved as **NP**-hard.

Algorithm 1: Sandwich Approach

Input: $G, k, \sigma_{AT}, \sigma_L, \sigma_U$
Output: seed set S
$S_L \leftarrow$ *ALG(G,k,* σ_L*)*;
$S_U \leftarrow$ *ALG(G,k,* σ_U*)*;
$S_{AT} \leftarrow$ *ALG(G,k,* σ_{AT}*)*;
return $\text{argmax}_{S\in\{S_L,S_U,S_{AT}\}}\, \sigma_{AT}(S)$;

We attempt to solve this NP-hard problem approximately. In the literature of *IM*, if influence function $\sigma_m(\cdot)$ is *monotone*, i.e. $\sigma_m(S) \leq \sigma_m(T)$ whenever $S \subseteq T$, and *submodular* which means if $S \subseteq T$, then $\forall v \notin T$ $\sigma_m(S \cup \{v\}) - \sigma_m(S) \geq \sigma_m(T \cup \{v\}) - \sigma_m(T)$, then the greedy algorithm [23] provides a result which approximates the optimal value within a factor of $(1 - \frac{1}{e})$ [33]. However, under *AT* model, influence function $\sigma_{AT}(\cdot)$ is not submodular (refer to Example. 1). Inspired by [31], we exploit the *Sandwich Approximation* technique to solve IM problem with solution-dependent lower bound.

Example 1 *Here a propagation example is taken to show the non-submodularity of* σ_{AT}*. According to the social graph presented in Fig. 3, when seed set is assigned as* $\{A\}$*,* $\{A,C\}$*,* $\{A,B\}$ *and* $\{A,B,C\}$ *respectively, then* $\sigma_{AT}(\{A,B\}) - \sigma_{AT}(\{A\}) < \sigma_{AT}(\{A,B,C\}) - \sigma_{AT}(\{A,C\})$*. As mentioned above, influence function* $\sigma_{AT}(\cdot)$ *is submodular iff* $\forall S \subseteq T$ *and* $v \notin T$*, the equation* $\sigma_{AT}(S \cup v) - \sigma_{AT}(S) \geq \sigma_{AT}(T \cup v) - \sigma_{AT}(T)$ *holds. So* $\sigma_{AT}(\cdot)$ *is not submodular.*

Let σ_L and σ_U be non-negative, monotone and submodular set functions defined on user set V, i.e. $\sigma_L : 2^V \rightarrow \mathbb{R}_{\geq 0}$ and $\sigma_U : 2^V \rightarrow \mathbb{R}_{\geq 0}$, such that $\forall S \subseteq V$, $\sigma_L(S) \leq \sigma_{AT}(S) \leq \sigma_U(S)$. The sandwich approach is described in Alg. 1, where *ALG* represents a greedy-manner approximation algorithm.

Theorem 2 *[31] The sandwich algorithm is able to generate a seed set* S_{sand} *such that* $\sigma_{AT}(S_{sand})$ *is not lower than*

$$max\{\frac{\sigma_{AT}(S_U)}{\sigma_U(S_U)}, \frac{\sigma_L(S^{opt})}{\sigma_{AT}(S^{opt})}\} \cdot (1 - \frac{1}{e}) \cdot \sigma_{AT}(S^{opt})$$

where S^{opt} *is the optimal solution.*

In accordance with theorem. 2, sandwich approach approximates the optimal result within a solution-dependent factor. Under this algorithmic framework, construction of bound functions, namely σ_L and σ_U, is the next critical issue.

One instance of σ_L can be built by a modification on *AT* model, that replacing Eq. 2 with the equation below:

$$\max_{u\in N_v^{(t)}}\{w_{u,v} \cdot \mathbb{1}_{\{t^{(u)}+t^{(u,v)}=t\}}\}, \tag{3}$$

where the decay factor is set as 0, and $\sum$ is replaced with max, i.e., at each step, only the weight which contributes most is picked to update awareness score. Similarly, to generate σ_U, we alter Eq. 2 to the equation:

$$A_v^{(t-1)} + \sum\nolimits_{u\in N_v^{(t)}} w_{u,v} \cdot \mathbb{1}_{\{t^{(u)}+t^{(u,v)}=t\}}, \tag{4}$$

where the decay factor is 1. Intuitively, by adoption of above modifications, for specified seed set S, we have $\sigma_L(S) \leq \sigma_{AT}(S) \leq \sigma_U(S)$.

Theorem 3 *Influence function* σ_L *(*σ_U*), which is built by modifying* AT *model in the way that altering Eq. 2 to Eq. 3 (Eq. 4), is non-negative, monotone and submodular.*

Algorithm 2: Two-Phase Search

Input: $G, k, \sigma_{AT}, \sigma_{LT}$
Output: scheduled seed set U^*

```
S^g ← Selection(G, k, σ_LT);
U^0 ← ∅ ;
foreach s_i ∈ S^g do
    Add (s_i, 0) into U^0
U* ← Schedule(G, σ_AT, U^0) ;
return U*;
```

Proof. Clearly, both σ_L and σ_U output non-negative numbers. We first prove the Monotonicity and submodularity of σ_L.

Suppose the propagation model corresponding to σ_L is called AT_L. Given a social graph G, for each $e_{u,v} \in E$, if $w_{u,v} < \theta_v$, $e_{u,v}$ is cut from G, then the generated graph is denoted by g. Under AT_L model, given a seed set S, $\forall v \in G - S$, v can be activated by S in G iff there is a path from S to v in g. So an intuitive result is that for any seed set $T \subseteq S$, $\sigma_L(T) \leq \sigma_L(S)$, i.e σ_L is monotone. Consider the quantity $\sigma_L(S \cup v) - \sigma_L(S)$ ($v \notin S$), it is the number of nodes which are reachable from v but unreachable from S in g. Obviously, if $T \subseteq S$, we have $\sigma_L(S \cup v) - \sigma_L(S) \leq \sigma_L(T \cup v) - \sigma_L(T)$. So σ_L is submodular.

Similarly, propagation model associated with σ_U is named by AT_U. Conditioned on AT_U model, for any inactive user v, $A_v^{(t)}$ is actually equal to the sum of all $w_{u,v}$ received. So for issued seed set S, its expected influence under AT_U equals the value under *LT*, i.e. $\sigma_U(S) = \sigma_{LT}(S)$. Because of the monotonicity and submodularity of σ_{LT}, it is trivial that σ_U is monotone and submodular.

Obviously, Alg. 1 is an approximation solution of IM problem, with the approximate ratio of $max\{\frac{\sigma_{AT}(S_U)}{\sigma_U(S_U)}, \frac{\sigma_L(S^{opt})}{\sigma_{AT}(S^{opt})}\} \cdot (1 - \frac{1}{e}) \cdot \sigma_{AT}(S^{opt})$.

6 Scheduled Influence Maximization

Viral marketing is to maximize brand awareness or product adoption over social networks, where the level of 'brand awareness or product adoption' is equivalent as user activation by influence propagation study. However, we observed that In *AT* model, the reach of information is also highly related to the schedule of seeds, i.e. the time step at which each seed is initially activated. Intuitively, the schedule of activating seeds is also essential in Viral marketing. Motivated by this issue, Sec. 6 proposed a new IM problem: *Scheduled Influence Maximization (SIM) problem* and addressed it with an approximation algorithm.

6.1 Scheduled Influence Maximization

Let $S = \{s_i | 1 \leq i \leq k\} \subseteq V$ be a k-element seed set, and $\Gamma = \{t_i | 1 \leq i \leq k\} (t_i \in N)$ be the corresponding schedule set, i.e. during a specific marketing campaign, each seed s_i is artificially activated at t_i. A *scheduled seed set* $U = \{(s_i, t_i) | s_i \in S, t_i \in \Gamma, 1 \leq i \leq k\}$ contains every selected seed and its activation time. In this paper, taking a scheduled seed set U as input, influence function $\sigma_{AT}(U)$ outputs the number of active users finally generated. Under *AT* model, formal definition of *Scheduled Influence Maximization (SIM)* problem is given as follows.

Definition 2 *(SIM Problem) Given a directed social graph $G = \{V, E, W, T, \Lambda\}$, and an integer budget $k \leq |V|$, search a scheduled seed set U, where $|U| \leq k$, that maximizes $\sigma_{AT}(U)$.*

Intuitively, for each tuple $(s_i, t_i) \in U$, if $t_i = 0$, SIM is trivially reduced to IM problem. As proved by Theorem. 4, the hardness result on SIM still holds.

Theorem 4 *SIM Problem under AT model is **NP**-hard.*

Proof. In *AT* model, if $\forall v \in V$, decay factor $\lambda_v = 1$. Then for any possible seed user set, arranging their activation time cannot change final output. So scheduled influence maximization problem is actually identical to the conventional influence maximization problem. As proved before, under *AT* model, influence maximization problem is **NP**-hard. Thus addressing scheduled influence maximization is also **NP**-hard.

6.2 Two-Phase Search

An intuitive observation is that, compared to IM problem, the search space of SIM is enlarged dramatically. To reduce the time cost of searching solution result, we design the *Two-Phase Search (TPS)* algorithm which divides the result exploration procedure into two phases, namely seed selection and seed schedule. The algorithmic details are described in Alg. 2. To be specific, seed set S^g is obtained by running greedy-manner approximation approach (*Selection*), like *IMM* [36], and their activation steps are managed by the *Schedule* module. In this work, three schedule algorithms, *Breadth First Schedule (BFS)*, *Depth First Schedule (DFS)*, and *Bucket Schedule (BS)* are proposed to implement the module.

Algorithm 3: Breadth First Schedule

Input: G, σ_{At}, U
Output: scheduled seed set U

1 $loop \leftarrow true$;
2 **while** $loop$ **do**
3     $loop \leftarrow false$;
4   **foreach** $(s_i, t_i) \in U$ **do**
5         $\varphi \leftarrow \sigma_{AT}(U), t_i \leftarrow t_i + 1$;
6   **if** $\sigma_{AT}(U) \leq \varphi$ **then**
7             $t_i \leftarrow t_i - 1$;
8   **else**
9             $loop \leftarrow true$;
10 **return** U;

BFS & DFS. As detailed in Alg. 3 and 4, taking U^0 (as mentioned in Alg. 2, in which every seed is activated at step 0) as input, *BFS* and *DFS* operate the schedule by deferring activation time heuristically. Specifically, in both of these two methods, seed-step pairs (s_i, t_i) are picked in a round-robin fashion for deferment test, and each test will increase t_i by one step if expected spread is raised accordingly.

The difference is that as shown in *BFS* (Alg. 3 : line 4 to 9), when one deferment test on (s_i, t_i) is finished, the next seed-step pair will be chosen. Differently, for specified tuple (s_i, t_i), *DFS* keeps deferring t_i, namely setting $t_i \leftarrow t_i + 1$, until no improvement on $\sigma_{AT}(U)$ is achieved, then next tuple will be loaded, shown in Alg. 4 : line 3 to 11. Both of these two methods terminate if no further promotion on $\sigma_{AT}(U)$ is possible.

BS. Another heuristic is that assuming U^+ is generated from U^0 by a decent schedule, and $\forall (s_i, t_i) \in U^+$, $t_i \in [0, B]$, where interval $[0, B]$ is called a *bucket*. *BS* targets to

Algorithm 4: Depth First Schedule

Input: G, σ_{AT}, U
Output: scheduled seed set U

```
loop ← true ;
while loop do
    foreach (s_i, t_i) ∈ U do
        loop ← true ;
        while loop do
            loop ← false ;
            φ ← σ_AT(U), t_i ← t_i + 1 ;
            if σ_AT(U) ≤ φ then
                t_i ← t_i − 1;
            else
                loop ← true;
return U;
```

Algorithm 5: Bucket Schedule

Input: G, σ_{AT}, U, B
Output: scheduled seed set U

```
loop ← true ;
while loop do
    loop ← false ;
    φ ← σ_AT(U) ;
    foreach (s_i, t_i) ∈ U do
        t_i ← argmax_{t_i ∈ [0,B]} {σ_At(U)} ;
        if σ_AT(U) > φ then
            loop ← true;
return U;
```

return a result which approaches to U^+ as much as possible (pseudo-code is listed in Alg. 5.). Similarly, *BS* also schedules seeds in rotation. When pair (s_i, t_i) is picked, t_i will be assigned as the value which maximizes $\sigma_{AT}(U)$ among all integer elements in $[0, B]$. If no spread improvement is achieved in the whole round, schedule iteration exits and final result is returned.

6.3 Theoretical Analysis

To evaluate the performance of U^* outputted by *TPS*, we first give Lemma. 1 by which Theorem. 5 next guarantees that *TPS* achieves an approximation ratio of $\alpha(1 - 1/e)$, where $\alpha = \sigma_{AT}(U^0)/\sigma_{LT}(S^g)$.

Lemma 1 *Given arbitrary scheduled seed set U and the corresponding seed set S, i.e. $|U| = |S|$ and $\forall(s_i, t_i) \in U$, $s_i \in S$, the expected spread of U is bounded by $\sigma_{LT}(S)$, formally $\sigma_{AT}(U) \leq \sigma_{LT}(S)$.*

Proof. Under *AT* model, for an issued scheduled seed set U, its expected spread $\sigma_{AT}(U)$ is computed by counting active users finally generated. An observation is that, if each decay factor $\lambda_v = 1$, since awareness score $A_v^{(t)}$ never declines over time, the expected spread of U reaches its peak value, formally $\sigma_{AT}(U) \leq \sigma_{AT}^{(\lambda=1)}(U)$. Meanwhile, conditioned on $\lambda_v = 1$, an intuitive result is that for specified seed set, its expected spread $\sigma_{AT}^{(\lambda=1)}$ is free of seeds schedule. Suppose U^0 is the scheduled seed set that $|U^0| = |U|$, and $\forall (s_i, t_i) \in U, (s_i, 0) \in U^0$. Thus we have, $\sigma_{AT}^{(\lambda=1)}(U) = \sigma_{AT}^{(\lambda=1)}(U^0)$. Moreover, with cutting of decay factor, user awareness score equals sum of the probabilistic weights received, which is actually equivalent to the pattern defined in *LT* model. So $\sigma_{AT}^{(\lambda=1)}(U^0) = \sigma_{LT}(S)$. Based on the above analysis, we have $\sigma_{AT}(U) \leq \sigma_{LT}(S)$. Thus the Lemma is proved.

Theorem 5 *Suppose* $\alpha = \sigma_{AT}(U^0)/\sigma_{LT}(S^g)$. *Alg. 2 approximates the optimum to within a factor of* $\alpha(1-1/e)$. *Formally* $\sigma_{AT}(U^*) \geq \alpha(1-1/e)\sigma_{AT}(U^{opt})$.

Proof. Let $S' = \{s_i | (s_i, t_i) \in U^{opt}\}$. For the submodularity, monotonicity and non-negativity of σ_{LT} [23], thus

$$\sigma_{LT}(S^g) \geq (1 - \frac{1}{e})\sigma_{LT}(S^{opt}) \geq (1 - \frac{1}{e})\sigma_{LT}(S')$$

By Lemma. 1, we have $\sigma_{LT}(S') \geq \sigma_{AT}(U^{opt})$, thus $\sigma_{LT}(S^g) \geq (1 - \frac{1}{e})\sigma_{AT}(U^{opt})$. Since $\sigma_{LT}(S^g) = \frac{\sigma_{AT}(U^0)}{\alpha}$, and $\sigma_{AT}(U^*) \geq \sigma_{AT}(U^0)$. Therefore, we can derive $\sigma_{AT}(U^*) \geq \alpha(1 - \frac{1}{e})\sigma_{AT}(U^{opt})$.

7 Experiments

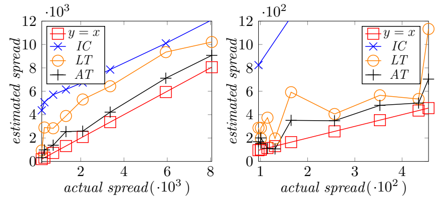
In this paper, we have tested the proposed solution of the SIM problem based on *AT* model. The general experiments are divided into two parts, the spread prediction and spread promotion (For space limitation, spread computation details are mentioned in our full technical report [2]). We have conducted intensive experiments on public datasets to demonstrate the effectiveness and efficiency of our solution.

7.1 Dataset

In addition to the dataset utilized in Section. 3, other real social graphs [3] are also used in our experimental evaluation, and their details are illustrated in Table. 2. All the graphs are constructed with directed edges and commonly-used for influence maximization research [20]. Specifically, *NetHEPT* and *DBLP* contains co-author(s) data in High Energy Physics (Theory) section of arXiv and DBLP Computer Science Bibliography respectively. Since the datasets of *NetHEPT* and *DBLP* do not include *actionlogs*, which means actual propagation size is unavailable, Experiment. 7.3 is only conducted on *Digg* and *Twitter*. For other experiments, all 4 datasets are used.

7.2 Parameter Setting and Implementation

All the weights associated on social edges are learned from user historical data by the method mentioned in Section 3. Specifically, for *NetHEPT* and *DBLP*, the weights are provided by the author of [20]. Refer to [30], for each edge $e_{u,v}$, the attached time delay $t^{(u,v)}$ follows a Poisson distribution where the mean is randomly sampled from $\{1, 2, \cdots, 20\}$. In reality, decay factor varies among different individuals. The best way to decide each decay factor is to learn from the historical data of each user. Unfortunately we do not have access to sufficient such real data sets for the purpose of experiments. Thus, we use synthesized setting. For each user v, her decay factor λ_v is set as a randomly-sampled value from $[1 - \theta_v, 1)$, which means lower threshold is more likely to generate a higher decay factor. Since easily-activated individuals usually show more interest on the propagated information, we thus assign them a slower speed of awareness-decay. For the computation of $\sigma_{AT}(U)$ and $\sigma_m(S)$, we adopt the Monte Carlo approach [23], instead of using a fixed number of samples, we determine the sample number dynamically, which is detailed in our technical report.



(a) Digg (b) Twitter

Figure 4. Spread Estimation

7.3 Evaluation Results

The evaluation can be generally grouped into two parts. In the first part, we measured the spread estimation of propagation models (including classic models, and *AT* model (Section. 7.3)). Second, we study the improvement of influence in the social network, and the efficiency of each scheduling method (Section. 7.3 and 7.3). All experiments are performed on an Ubuntu-installed machine equipped with an Intel Xeon(R) CPU E5-1620 v2 @ 3.70GHz $\times 8$ and 16G memory. Results are obtained by running each experiment 10 times and taking the average.

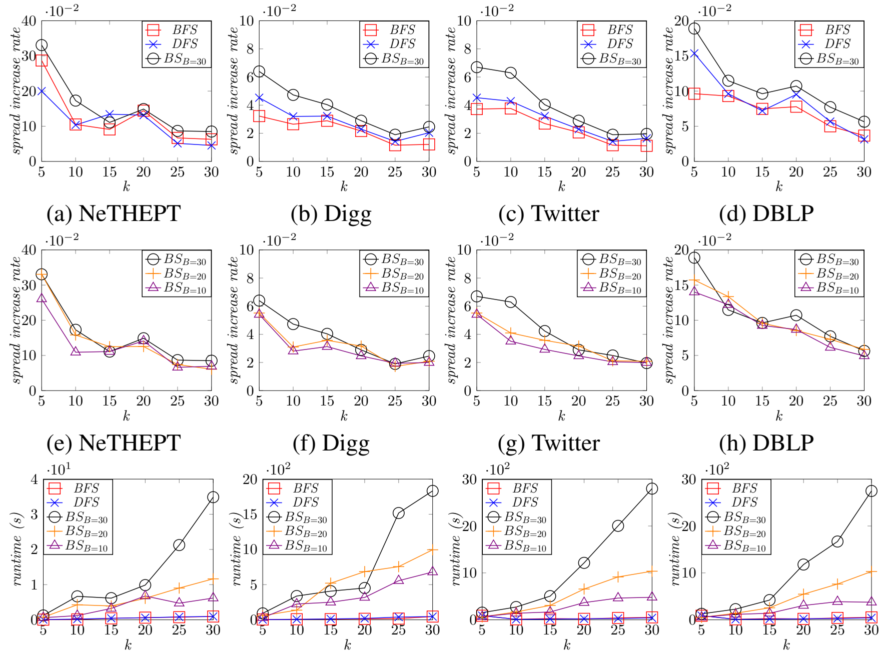
Model Evaluation Since the aforementioned decay effect is not adopted in model construction, classic propagation models usually over-estimates user activation probability, which eventually causes over-prediction in influence spread. This section targets to evaluate propagation models in the way that comparing the real influence spread with the spread values estimated by models. By observing the estimation gap, the effectiveness of propagation models can be evaluated. Experimentally, for specified seed set S, its *expected spread* is compared with the *actual spread*. For ground truth, the seeds of each action are set according to Section 3, and actual spread is the number of users who performed the same action, namely propagation size.

As shown in Fig. 4, in contrast to actual propagation size, the expected spread given by *IC* and *LT* models severely suffers from over-prediction. In *Digg*, the model-caused extra prediction reaches to 60% of the actual spread. While in *Twitter*, expected spread even exceeds the actual value by numerous times.

In *AT* model, activation steps of seeds are set according to their action timestamps. Specifically, timestamp of the earliest action is recorded as $t = 0$, then $t = 1$ denotes the next timestamp. (in our dataset, there are 12 hours between two timestamps.) As demonstrated by empirical results, expected spread given by *AT* model approaches the actual value closer. Since *AT* model captures decay nature inside user awareness, when it is utilized for spread estimation purpose, over-prediction problem can be eased much.

Spread Promotion As addressed above, decent schedule on seeds (activation time) can contribute to enlarge influence spread. Our second set of experiments compare spread promotion created by different schedule algorithms. To measure the spread promotion, *spread increase rate* is defined first. Formally, with fixed budget issued, *spread increase rate*= $\frac{\sigma_{AT}(U)-\sigma_{AT}(S)}{\sigma_{AT}(S)}$, where U denotes scheduled seed set returned by *TPS* (Alg. 2), and S represents seed set outputted by non-scheduling *sandwich approach* (Alg. 1). Fig. 5(a) to (d) report spread increase rate achieved, varying k from 5 to 30. Each line is labeled by the schedule algorithm invoked in *TPS*.

Overall, when $B = 30$, *BS*, notated as $BS_{B=30}$, outperforms *BFS* and *DFS* over all budget values. Moreover, under $k = 5$, invocation of $BS_{B=30}$ raises spread increase rate to its maximal value. In *NetHEPT*, the maximum reaches to 33%, while in *DBLP*, it is 18%. Another observation is that as growing budget issued, spread increase rate tends to go downward. Experiment results show that when incremental seeds are allocated on a social graph, $\sigma_{AT}(S)$ increases faster than $\sigma_{AT}(U) - \sigma_{AT}(S)$ which denotes



(a) NeTHEPT (b) Digg (c) Twitter (d) DBLP

(e) NeTHEPT (f) Digg (g) Twitter (h) DBLP

(i) NeTHEPT (j) Digg (k) Twitter (l) DBLP

Figure 5. Spread Increase Rate & Efficiency Comparison

the additional spread created by *TPS*. Hence issuing k incrementally causes the rate decreases.

We are also interested in investigating how assignment of B affects performance of *BS*. Theoretically, with a greater B inputted, *BS* explores solutions in a larger search space, so better results may be returned. In practice, as shown in Fig. 5(e) to (h), a bigger B generally boosts spread increase rate, but merely in a limited extent. By contrast, bucket size addition results in much extra computation time (discussed in Section. 7.3).

Efficiency Analysis This section evaluates computation cost of running schedule algorithms. As a function of seed number, runtime of schedule algorithms is reported in Fig. 5 (i) to (l). It can be seen that *BFS* and *DFS* scale well with respect to k. For instance, in the largest dataset *Twitter* and *DBLP*, *DFS* only takes around 40 seconds to schedule 30 seeds, and *BFS* performs equally. Comparatively, as seeds are incrementally imported, *BS* requires significantly more time to handle the schedule task, especially when k is greater than 20. In addition to seed number, if bucket size B is enlarged, *BS* also heavily suffers from consequent inefficiency issue. Fig. 5(i) and (l) show that conditioned on $k = 30$, time cost of BS_{20} is even tripled when B is increased to 30.

8 Related Work

8.1 Influence Propagation Model

Classic models, LT and IC model have been introduced in Sec. 2. Apart from these two models, there are also extensive research on other models simulating the influence propagation.

Time-dependent Model In *Latency Aware Independent Cascade (LAIC)* model [30], for each piece of propagation (e.g. at time t_a user u sends a specific message to her friend v, then at t_b, v receives it), the time delay, i.e. $t_b - t_a$, is expressed as a delay weight associated with each social link. Similarly, this time delay is also incorporated by *Independent Cascade model with Meeting events (IC-M)* [10]. In the context, at each time step newly-activated users perform activation on their inactive neighbors by some pre-determined probabilities. If it fails to occur, active users will continue activation attempt in subsequent steps until success. The time span from initial attempt to the final reflects this time delay. Other time-dependent models like *Delayed Linear Threshold* and *Delayed Independent Cascade* [32] are constructed in similar manners. All these models integrate the aforementioned time delay into influence propagation mechanism, so they can track information propagation progress over time. But the decay effect inside user awareness is omitted, which means the fundamental problem in classic models is unsolved.

Decay Model Motivated by the intuition that fresh news exhibits more attractiveness than out-of-date information, *Independent Cascade with Novelty Decay (IC_{ND})* [14] is proposed to model information diffusion conditioned on that novelty inside the diffused message decays as its exposure frequency increasing. For instance, the initial weight assigned on edge $e_{u,v}$ is w, where v is inactive. If n neighbors (except u) have attempted to activated v (and failed), i.e. a specific message *M* has been exposed to v n times, then w will keep decreasing as n grows, which indicates that *M* becomes less attractive to v. In IC_{ND},edge weights decay is dependent on the times of information exposure, which are actually related to the network structure and diffusion trace. So the time-dependent nature of decay effect is not revealed.

Other models like *Time-varying Independent Cascade model (TV-IC)* [34] and *Cascade Model with Diffusion Decay (CMDD)* [38] all build edge weights as time-decay functions, i.e. as time lapse, the weights vary non-increasingly. These models are constructed under the hypothesis that information freshness decays as a function of time, i.e. it becomes less attractive over time. For instance, at step t, the weight associated on $e_{u,v}$ is w, at subsequent steps, even if the propagated message *M* has never been exposed to v, value of w still decreases. However, above hypothesis is challenged by experimental results given in [35], it turns out that edge weights diminishing only occurs if messages are not fresh to the social circles. So back to the last instance, weight of $e_{u,v}$ should keep invariant if v has not touched *M*. Previous time-decay models only simply relate influence relationship decay and time lapse together, and omit the rationale behind this decay effect, which causes that the fundamental defined by these models is inconsistent with influence propagation nature. Thus existing decay mechanisms are unable to depict propagation dynamics.

8.2 Scalable Influence Maximization

By exploiting submodularity of influence functions, Leskovec *et al.* propose *CELF* technique [27] which accelerates the classic Greedy dramatically without compromise on accuracy guarantee. Then an improved version, called *CELF++* [19], incrementally contributes to efficiency enhancement. However, running of *CELF* (and *CELF++*) triggers thousands of sampling operation, which causes heavy time cost in practice. Then several algorithms aim to solve IM heuristically, e.g. PMIA [11], IRIE [24] and Simpath [20]. Subsequently, the technique breakthrough, namely reverse reachable subgraph sampling, is achieved by Borgs *et al.*, theoretically, the novel approach *RIS* [6] is able to provide elegant performance guarantee in both of efficiency and effectiveness. Inspired by *RIS*, Tang *et al.* develop *TIM* [37] and *IMM* [36]. Also, there are many problems motivated by IM problem. Competitive Influence Maximization (CIM) [5], a widely discussed problem, considers multiple agents attempting to maximize their influence over the social network competitively. Another instance is Influence Blocking Maximization (IBM) [21] which targets to minimize the influence triggered by the competing seeds.

However, all these problems assume that seeds are activated initially, and seeds scheduling has not been well studied.

9 Conclusion

In this paper, based on the *decaying* observation and verified experiments, we propose a new model *Awareness Threshold (AT)*. This model enables a more accurate estimation of the propagation process. Conditioned on *AT* model, a tailored algorithm is introduced to address influence maximization (IM) problem approximately. Further, under the *AT* model, selection and schedule of seeds collaboratively impact final information spread. Structured on this investigation, *scheduled influence maximization (SIM)* problem is proposed next. To tackle this challenge, *Two-Phase Search* method is developed to approximate the optimal value with a lower bound guarantee. Finally, the approach utility is evaluated by intensive experiments.

Acknowledgment We would like to thank the reviewers for the insightful comments. Haiqi Sun, Jing Yan, Yudian zheng, and Reynold Cheng were supported by the Research Grants Council of Hong Kong (RGC Projects HKU 17229116 and 17205115) and the University of Hong Kong (Projects 104004572, 102009508 ,104004129).

References

1. `http://www.isi.edu/integration/people/lerman/downloads.html`.
2. `https://www.dropbox.com/s/g8bfafun0n055ou/main.pdf?dl=0`.
3. `https://www.dropbox.com/s/zhqnvsy5a5xocfa/simpath_icdm11_datasets.zip?dl=0`.
4. S. Almon. The distributed lag between capital appropriations and expenditures. *Econometrica: Journal of the Econometric Society*, 1965.
5. S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, 2007.
6. C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, 2014.
7. S. Broadbent. *Accountable advertising: A handbook for managers and analysts*. admap publications, 1997.
8. G. Brown. Modelling advertising awareness. *The Statistician*, 1986.
9. S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang. Online topic-aware influence maximization. *PVLDB*, 2015.
10. W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI*, 2012.
11. W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 2010.
12. W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, 2009.
13. J.-P. Dubé, G. J. Hitsch, and P. Manchanda. An empirical model of advertising dynamics. *Quantitative marketing and economics*, 2005.

14. S. Feng, X. Chen, G. Cong, Y. Zeng, Y. M. Chee, and Y. Xiang. Influence maximization with novelty decay in social networks. In *AAAI*, 2014.
15. R. Ferguson. Word of mouth and viral marketing: taking the temperature of the hottest trends in marketing. *Journal of Consumer Marketing*, 2008.
16. T. R. Fry, S. Broadbent, J. M. Dixon, et al. Estimating advertising half-life and the data interval bias. Technical report, Monash University, Department of Econometrics and Business Statistics, 1999.
17. A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.
18. A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 2011.
19. A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *WWW*, 2011.
20. A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, 2011.
21. X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474. SIAM, 2012.
22. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
23. D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *Automata, languages and programming*. 2005.
24. J. Kim, S.-K. Kim, and H. Yu. Scalable and parallelizable processing of influence maximization for large-scale social networks? In *ICDE*, 2013.
25. J. Kirby. Viral marketing. *Connected marketing*, 2012.
26. S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart. Online influence maximization. In *KDD*, 2015.
27. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.
28. G. Li, S. Chen, J. Feng, K.-l. Tan, and W.-s. Li. Efficient location-aware influence maximization. In *SIGMOD*, 2014.
29. S.-C. Lin, S.-D. Lin, and M.-S. Chen. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *KDD*, 2015.
30. B. Liu, G. Cong, D. Xu, and Y. Zeng. Time constrained influence maximization in social networks. In *ICDM*, 2012.
31. W. Lu, W. Chen, and L. V. Lakshmanan. From competition to complementarity: comparative influence diffusion and maximization. *PVLDB*, 2015.
32. A. Mohammadi, M. Saraee, and A. Mirzaei. Time-sensitive influence maximization in social networks. *Journal of Information Science*, 2015.
33. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 1978.
34. N. Ohsaka, Y. Yamaguchi, N. Kakimura, and K.-i. Kawarabayashi. Maximizing time-decaying influence in social networks. In *ECML PKDD*, 2016.
35. G. V. Steeg, R. Ghosh, and K. Lerman. What stops social epidemics? In *ICWSM*, 2011.
36. Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *SIGMOD*, 2015.
37. Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*, 2014.
38. Z. Zhang, H. Wu, K. Yue, J. Li, and W. Liu. Influence maximization for cascade model with diffusion decay in social networks. In *International Conference of Young Computer Scientists, Engineers and Educators*, 2016.