

# Maximizing Time-Decaying Influence in Social Networks

Naoto Ohsaka<sup>1,4(✉)</sup>, Yutaro Yamaguchi<sup>2,4</sup>, Naonori Kakimura<sup>1,4</sup>,  
and Ken-ichi Kawarabayashi<sup>3,4</sup>

<sup>1</sup> The University of Tokyo, Tokyo, Japan

ohsaka@is.s.u-tokyo.ac.jp, kakimura@global.c.u-tokyo.ac.jp

<sup>2</sup> Osaka University, Osaka, Japan

yutaro\_yamaguchi@ist.osaka-u.ac.jp

<sup>3</sup> National Institute of Informatics, Tokyo, Japan

k\_keniti@nii.ac.jp

<sup>4</sup> JST, ERATO, Kawarabayashi Large Graph Project, Tokyo, Japan

**Abstract.** Influence maximization is a well-studied problem of finding a small set of highly influential individuals in a social network such that the spread of influence under a certain diffusion model is maximized. We propose new diffusion models that incorporate the time-decaying phenomenon by which the power of influence decreases with elapsed time. In standard diffusion models such as the independent cascade and linear threshold models, each edge in a network has a fixed power of influence over time. However, in practical settings, such as rumor spreading, it is natural for the power of influence to depend on the time influenced. We generalize the independent cascade and linear threshold models with time-decaying effects. Moreover, we show that by using an analysis framework based on submodular functions, a natural greedy strategy obtains a solution that is provably within  $(1 - 1/e)$  of optimal. In addition, we propose theoretically and practically fast algorithms for the proposed models. Experimental results show that the proposed algorithms are scalable to graphs with millions of edges and outperform baseline algorithms based on a state-of-the-art algorithm.

## 1 Introduction

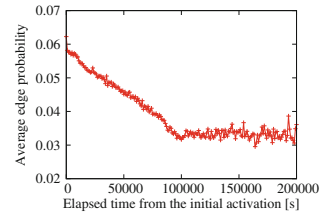
Recently, the rapidly increasing popularity of online social networks has created opportunities to study information diffusion that models the spread of news, ideas, and product adoption throughout the population. Motivated by applications to marketing, Domingos and Richardson [8] introduced *viral marketing*, which is a cost-effective marketing strategy that promotes products through word-of-mouth. Formally, the *influence maximization problem* [17] asks, for a parameter  $k$ , to find a set of  $k$  vertices in a social network such that the expected number of activated vertices is maximized.

The *independent cascade (IC)* model and *linear threshold (LT)* model are two of the most basic and widely studied diffusion models in influence maximization.

The IC model proposed by Goldenberg et al. [10] focuses on individual (and independent) interaction among friends in a social network. The LT model [14] focuses on threshold behavior in influence propagation, where a user is influenced when a sufficient number of friends are influenced. The two models are tractable as they are shown to have submodularity [17], which has motivated substantial theoretical and practical follow-up research [2, 4, 5, 20, 25, 28, 29].

Recent research into empirical social networks [11, 15, 16] reports that time plays an important role in the spread of influence in a network. However, in the IC and LT models, each edge in a network has a fixed power of influence over time. This does not reflect reality. In a practical setting, such as rumor spreading, the power of influence may decay over time. Moreover, the influence may be propagated with delay. In this paper, we incorporate two types of temporal phenomena, i.e., *time-decaying phenomenon* and *time-delay propagation*, into both the IC and LT models.

*Time-decaying Phenomenon.* First, we consider the “freshness” of information. Intuitively, a rumor has a lifetime, and a new idea is often affected by trends. One may also observe that information becomes less attractive over time. To observe such phenomenon in a real-world social network, we estimate edge probabilities that represent the power of influence *at each time* by applying the method of [13] to the Digg dataset.<sup>1</sup> Figure 1 shows the transition of the average edge probabilities among all edges. As can be seen, the average edge probability clearly decreases over time, especially halving in a day. Thus, the power of word-of-mouth effects strongly depends on the elapsed time. This motivates us to introduce a time-decaying phenomenon to information diffusion models.



**Fig. 1.** Time transition of the average edge probability.

*Time-delay Propagation.* In addition to the time-decaying effect, we incorporate a time-delay effect, which has been studied extensively [11, 13]. In many real-world examples, the propagation of influence from one person to another may have a certain time delay due to heterogeneity in human activities. Thus, the speed of influence spread varies. We capture such time-delaying propagation in our model by extending the model [11, 22, 26] with the time-decaying phenomenon.

## 1.1 Contributions

In this paper, we address the above temporal issues and extend well-studied diffusion models. We first propose an IC model that incorporates time-decaying probabilities and time-delay propagation. The salient feature of this model is that the power of influence on edges decays over time, as shown in Fig. 1. This model is

<sup>1</sup> <http://www.isi.edu/~lerman/downloads/digg2009.html>.

simple and sufficiently general to deal with various time-decaying probabilities. It should also be noted that our model includes most previous models with temporal effects, such as [12] (see Sect. 3.2). In addition, we propose an LT model with time-decaying probabilities and time-delay propagation, which is another interpretation of temporal phenomena with threshold behavior.

Our main contributions are summarized as follows.

- **Time-varying IC model (Sect. 3):** We extend the IC model with time-decaying probabilities and time-delay propagation. We show that the expected number of activated vertices under the extended model is monotone and submodular with respect to an initial vertex set. Therefore, we can efficiently find a solution that approximates an optimal solution within the ratio  $(1 - 1/e - \epsilon)$  using a greedy algorithm [24].
- **Time-varying LT model (Sect. 4):** We introduce the LT model with time-decaying probabilities and time-delay propagation, in which the influences from neighbors decay over time. As with the extended IC model, the expected number of activated vertices is monotone and submodular, and we can efficiently find an approximate solution within the ratio  $(1 - 1/e - \epsilon)$  using a greedy algorithm.
- **Scalable algorithms (Sect. 5):** We propose scalable and accurate algorithms for influence maximization under the proposed models by generalizing sketching methods [2, 28]. To this end, we design novel dynamic programming that can deal with time-decaying probabilities efficiently.
- **Experimental evaluations (Sect. 6):** We conduct experiments on real-world social networks and demonstrate that the proposed algorithms outperform baseline methods in terms of both efficiency and accuracy.

Due to space limitations, we omit some of the proofs and the experimental results, which will be found in the full version of this paper.

## 2 Related Work

Inspired by the work of Domingos and Richardson [8], Kempe et al. [17] formulated the influence maximization problem as a discrete optimization problem. They showed that the influence maximization problem is NP-hard for the IC and LT models and that the expected number of activated vertices is a monotone and submodular function with respect to an initial vertex set. This implies that an optimal solution for the influence maximization problem can be approximated efficiently within the ratio  $(1 - 1/e - \epsilon)$  with a greedy-type algorithm [24].

Since Kempe et al.’s greedy algorithm suffers from poor scalability, a plethora of scalable algorithms have been proposed. Existing approaches (for the IC and LT models) can be roughly classified into three types. *Simulation-based* methods [5, 17, 20, 25] conduct Monte-Carlo simulations of the diffusion process to estimate the influence spread accurately; however, they suffer from inefficiency. *Heuristic-based* methods [4, 5, 18] avoid using Monte-Carlo simulations by restricting the spread of influence in a particular group, which often results

in poor-quality solutions due to an absence of accuracy guarantees. *Sketch-based* methods [2] resolved the inefficiency of Monte-Carlo simulations while preserving accuracy guarantees. Rather than directly simulating the diffusion process, sketch-based methods build sketches in advance based on an outcome of *reverse simulations*, and efficiently estimate the influence spread. Subsequently, several strategies for bounding the sketch size have been developed [28, 29]. In this paper, we generalize sketch-based methods to our proposed models without significant deterioration of efficiency.

Various information diffusion models with time-delay propagation have been proposed in different contexts [3, 11, 13, 22, 26, 27] to resemble actual cascade distribution. The influence maximization problem in such models has also been studied [3, 9, 12, 28]. We show in Sect. 3.2 that most previous models are included in the proposed model. Note that, the existing models only consider the time difference between two vertices. In contrast, our model considers the time reached from the seeds, which allows us to introduce the time-decaying probabilities, as well as the time difference.

### 3 Time-Varying IC Model

#### 3.1 Model Definition

Here we define the time-varying IC model formally. Let  $G = (V, E)$  be a directed graph, where  $V$  is a vertex set of size  $n$  and  $E$  is an edge set of size  $m$ . For a vertex  $v$  in  $V$ ,  $N^+(v)$  denotes the set of out-neighbors of  $v$ . Each individual vertex can be either *active* (an adopter of the innovation) or *inactive*. In the *time-varying (TV) IC* model, we begin with a seed set  $A$  of active vertices. Then, the process unfolds according to the following randomized rule. When a vertex  $u$  becomes active at time  $t_u$  for the first time, it is given a single chance to activate each current inactive neighbor  $v$  of  $u$  through the edge  $e = (u, v)$ . Here unlike the standard IC model, both the distance to  $v$  and the probability to activate  $v$  depend on time. That is, the conditional likelihood that the influence reaches  $v$  at time  $t$  is defined by  $f_e(t | t_u)$ . We assume that the likelihood is shift invariant, i.e.,  $f_e(t | t_u) = f_e(t - t_u)$ , and nonnegative, i.e.,  $f_e(s) = 0$  for  $s < 0$ . Moreover, when  $v$  receives the influence at time  $t$ , the probability to be activated is given by a nonincreasing function  $p_e : \mathbb{R}_+ \rightarrow [0, 1]$  of the arrival time, i.e.,  $p_e(t)$ . Thus, the probability that  $v$  becomes active at time  $t$  is

$$\Pr[v \text{ becomes active at time } t \mid u \text{ is active at time } t_u] = p_e(t)f_e(t - t_u). \quad (1)$$

When  $v$  receives influence from more than one newly activated neighbors simultaneously, their attempts to activate  $v$  are sequenced independently in arbitrary order. The process runs until no further activations are possible.

Intuitively, the term  $p_e(t)$  represents the decrease in power of influence as time passes, because  $p_e$  is a nonincreasing function on elapsed time. On the other hand,  $f_e(t - t_u)$  represents the time-delay effect on the edge  $e$ . Note that if  $p_e(t)$  is a constant  $c_e$  for any  $t$ , then this model is identical to the IC model.

### 3.2 Examples of TV-IC Models

Here we present examples of the TV-IC model. The first example is the influence maximization problem with deadline.

**Example 1 (Influence maximization with deadline).** Let  $c_e \in [0, 1]$  be a constant for each edge  $e$  and let  $T$  be a positive number. Consider the TV-IC model where a function  $p_e$  is given by  $p_e(t) = c_e$  if  $t \leq T$  and  $p_e(t) = 0$  if  $t > T$ . This case means that the influence will expire at time  $T$ . Therefore, the influence maximization problem over such a model is to maximize the expected number of vertices activated before deadline  $T$ .

Moreover, the TV-IC model extends previous models properly.

**Example 2 (Continuous-time independent cascade (CTIC) model [11, 22, 26]).** The TV-IC model where a function  $p_e$  is a constant includes various previously proposed models. For example, Saito et al. [26, 27] considered the case where  $f_e$  is an exponential function. In their model, the time-delay parameters  $r_e > 0$  and diffusion parameters  $c_e \in (0, 1)$  for each edge  $e = (u, v)$  are given. When  $u$  is activated at time  $t_u$ ,  $u$  will activate an inactive neighbor  $v$  with probability  $c_e$ . If it succeeds, a delay time  $\delta$  is sampled from the exponential distribution  $r_e \exp(-r_e \delta)$ , and  $v$  will become active at time  $t_u + \delta$ . Gomez-Rodriguez et al. [11] dealt with more general functions of  $f_e$ . However, in their model,  $p_e(t)$  is set to 1 for all  $e$  and  $t$ , which means that a vertex  $u$  always activates its neighbor  $v$  at some time. A similar model was also proposed in [13, 22]. However, in all these models,  $p_e$  is a constant independent of time  $t$ .

**Example 3 (Independent cascade model with meeting events [3]).** In this model, we are given meeting probabilities  $m_e$  and propagation probabilities  $c_e$ . When a vertex  $u$  is activated at time  $t_u$ ,  $u$  will attempt to meet an inactive neighbor  $v$  with probability  $m_e$ , where  $e = (u, v)$ . Thus a time-delay  $\delta \in \{1, 2, \dots\}$  occurs with probability  $m_e(1 - m_e)^{\delta-1}$ . When they meet,  $u$  will activate  $v$  with probability  $c_e$  at that time. This means that the probability that  $u$  will activate  $v$  at time  $t_u + \delta$  is  $c_e m_e (1 - m_e)^{\delta-1}$ . Thus, it is included in our model, where  $p_e$  is a constant.

The following example is one in which the probability decays at arrival time  $t$ . However, the time delay effect is not considered.

**Example 4.** Assume that  $p_e$  is given by  $p_e(t) = r_e \alpha(t)$  for some constant  $r_e$  and a nonincreasing function  $\alpha : \mathbb{R}_+ \rightarrow [0, 1]$ , which represents the influence decay factor. This is the IC model wherein the probability is decreased by a factor of  $\alpha(t)$  when the influence is reached at time  $t$ . This case includes the temporal factor proposed by Cui et al. [7], where  $p_e(t) = r_e \exp(-ct)$  for some constant  $c$ . Note that the model proposed by Cui et al. [7] is more general in order to resemble actual cascade distribution, which does not clearly possess submodularity. Thus, their model has no theoretical guarantee for influence maximization.

### 3.3 Submodularity of the Influence Spread Function

We say that a set function  $f : 2^V \rightarrow \mathbb{R}$  is *monotone* if  $f(S) \leq f(T)$  for all  $S \subseteq T \subseteq V$ , and *submodular* if  $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$  for all  $S \subseteq T \subseteq V$  and  $v \in V \setminus T$ .

Let  $\sigma(A)$  be the expected number of vertices activated after running the process of the TV-IC model with an initial seed set  $A$ . The following theorem is the main technical result in this section, which generalizes Kempe et al. [17].

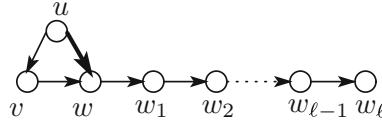
**Theorem 1.** *For the TV-IC model,  $\sigma$  is a monotone submodular function.*

We here present the proof idea. The detailed proof is deferred to the next section. First, we remove the time-delay factor  $f_e$ , similar to the proof of [12]. Consider the probability distribution obtained by  $f_e$  of all possible time differences between each pair of nodes in the network and sample a length  $d_e$  of each edge  $e$  from the probability space. Let  $\sigma_d$  be the expected number of vertices activated, assuming that the length of an edge  $e$  is  $d_e$ . Since  $\sigma$  is the expected value of  $\sigma_d$ , it is sufficient to show that  $\sigma_d$  is monotone and submodular.

Here we focus on the time-decay factor  $p_e$ . Note that a standard “coin flipping” technique for the IC model [17] would not work to show the submodularity when  $p_e$  depends on time. The key observation was that a set of activated vertices corresponds to the reachability of a random graph generated by “coin flipping” on each edge. Then, the expected size of the reachable vertices is shown to be monotone and submodular. This technique tells us that we do not have to consider time in the IC model. However, due to the time dependency of probability, we cannot directly apply this observation to our problem setting.

To overcome this difficulty, we prepare a random variable  $x_e$  in the range  $[0, 1]$  on each edge  $e$  before the process. Based on these values, we construct a graph in a deterministic manner such that the reachability of the graph is equal to the activated vertices. Note that the obtained graph depends on a seed set, in contrast to Kempe et al. [17]. This requires more careful analysis in the proof.

It should also be noted that the time decay of probabilities is essential to satisfy the submodularity of  $\sigma_d$ . To demonstrate this, consider a graph consisting of a directed triangle  $(u, v), (v, w), (u, w)$  with a directed path  $(w, w_1), (w_1, w_2), \dots, (w_{\ell-1}, w_\ell)$  of length  $\ell > 1$ . The length  $d$  of the edges is defined as  $d_{uw} = 3$  and  $d_e = 1$  for any edge  $e \neq (u, w)$ . The probabilities are set to  $p_{vw}(t) = 0$  for  $t < 2$ ,  $p_{vw}(t) = 1$  for  $t \geq 2$ ,  $p_{ww_1}(t) = 0$  for  $t < 4$ ,  $p_{ww_1}(t) = 1$  for  $t \geq 4$ , and  $p_e(t) = 1$  for any other edge  $e$  and any time  $t$ . This is illustrated in Fig. 2. For this graph, if we take  $\{u\}$  as the seed set, then  $u$  activates  $v$  in time  $t = 1$ ,  $v$  activates  $w$  in time  $t = 2$ , and  $w$  fails to activate  $w_1$  in time  $t = 3$ , which stops diffusion. If we take  $\{v\}$ , then  $v$  fails to activate  $w$  in time  $t = 1$  and diffusion terminates. However, if we take  $\{u, v\}$ , then  $v$  fails to activate  $w$  in time  $t = 1$ , but  $u$  succeeds in time  $t = 3$ , and  $w$  activates  $w_1$  in time  $t = 4$ , which eventually results in influence spreading to all vertices. Thus, we have  $\sigma_d(\{u, v\}) - \sigma_d(\{u\}) = (\ell + 3) - 3 > 1 = \sigma_d(\{v\}) - \sigma_d(\emptyset)$ , which violates submodularity.



**Fig. 2.** Example violating submodularity when probability increases.

It follows from Theorem 1 that, using Monte-Carlo simulations to estimate  $\sigma(A)$ , we can maximize  $\sigma$  within  $(1 - 1/e - \epsilon)$  approximation factor by a greedy algorithm [24]. However, naive Monte-Carlo simulations require significant time to estimate  $\sigma(A)$ . Because the proof of Theorem 1 has a nice combinatorial structure, we provide a theoretically efficient algorithm in Sect. 5.

**3.4 Proof of Theorem 1**

Here we prove Theorem 1. Let  $G = (V, E)$  be a directed graph and  $p_e : \mathbb{R}_+ \rightarrow [0, 1]$  be a nonincreasing function for each edge  $e$ . As described in Sect. 3.3, we may assume that if  $u$  becomes active at time  $t$ , then the influence reaches a neighbor  $v$  at time  $t + d_e$ , and the probability that  $v$  becomes active is  $p_e(t + d_e)$ .

For each edge  $e$ , we choose a number  $x_e$  in the range  $[0, 1]$  uniformly at random. We assume that we can use the edge  $e = (u, v)$  to activate  $v$  in the TV-IC model if the arrival time  $t$  satisfies  $p_e(t) > x_e$ . Then, the probability that  $e$  can activate  $v$  in time  $t$  is equal to  $p_e(t)$ , which is the case when  $x_e$  is the range  $[0, p_e(t)]$ .

Let  $X = (x_e)$  be a choice of random numbers  $x_e$  for all edges  $e \in E$ . Then, the number of activated vertices is determined uniquely by such  $X$ .  $\sigma_X(A)$  is defined as the total number of vertices activated by a seed set  $A$  by running the process with  $X$ . Since each edge is used at most once in the process,  $\sigma(A)$  can be described by the functions  $\sigma_X(A)$ :

$$\sigma(A) = \int \Pr[X] \sigma_X(A) dX.$$

The function  $\sigma_X$  can be characterized by the reachability of a graph. For each edge  $e$ , we say that  $e$  is *live in time t* if  $p_e(t) > x_e$ . For each vertex  $v$ , we denote  $N_t^+(v) = \{w \in N^+(v) \mid (v, w) \text{ is live in } t + d_{vw}\}$ . For a seed set  $A$ , we construct a graph  $G_X(A)$  as follows.

**Procedure to obtain  $G_X(A)$  from  $X$  and  $A$ .**

**Step 0.** Set  $r_v = 0$  for each  $v \in A$  and  $r_v = +\infty$  for each  $v \in V \setminus A$ . Set  $t = 0$ .

**Step 1.** While  $t < +\infty$  do the following:

**1-1.** Define  $V_t = \{v \in V \mid r_v = t\}$ .

**1-2.** For each  $v \in V_t$  and each  $w \in N_t^+(v)$ , replace  $r_w$  with  $\min\{r_v + d_{vw}, r_w\}$ .

**1-3.** Increase  $t$  to  $\min\{r_v \mid r_v > t\}$ .

**Step 2.** Return  $G_X(A) = (R, F)$ , where  $R = \bigcup_{t < +\infty} V_t$  and  $F = \{(v, w) \mid r_w = r_v + d_{vw}\}$ .

Note that this procedure simulates the TV-IC model when we fix a choice  $X$ . The obtained vertex set  $R$  is the set of vertices activated by  $A$ .

By the above procedure, we show in Lemma 1 that  $\sigma_X$  is monotone and submodular, which implies Theorem 1. Note that the construction of  $G_X(A)$  depends on the given seed set  $A$ ; thus, we cannot extend the proof in [17] directly, and we must consider the dynamics of reachability.

**Lemma 1.** *The function  $\sigma_X$  is monotone and submodular.*

## 4 Time-Varying LT Model

Let  $G = (V, E)$  be a directed graph. Each vertex  $v$  chooses a *threshold*  $\theta_v \in [0, 1]$  uniformly at random. Each edge  $e$  has a nonincreasing function  $q_e : \mathbb{R}_+ \rightarrow [0, 1]$  and has a function  $f_e : \mathbb{R} \rightarrow [0, 1]$  that represents the shift invariant conditional likelihood as in the TV-IC model. We suppose that  $\sum_{e:e=(u,v)} q_e(0) \leq 1$  for each  $v \in V$ .

Given a seed set  $A$ , the diffusion process in the *time-varying (TV) LT* model unfolds, similar to the LT model. The difference is that the distance to a neighbor and the amount of influences from neighbors depend on arrival times. Consider the case wherein a vertex  $u$  becomes active in time  $t_u$ . Then, each edge  $(u, v)$  delivers an influence to  $v$ , where the likelihood that the influence reaches  $v$  at time  $t$  is  $f_e(t - t_u)$ . When the influence reaches  $v$  at time  $t$ , the amount of influence that  $v$  receives is  $q_e(t)$ . The vertex  $v$  becomes activated once the total influence exceeds the threshold  $\theta_v$ .

Similar to the TV-IC model,  $q_e(t)$  represents the time decay of influence, and  $f_e(t - t_u)$  represents the time-delay of propagation. Note that if  $q_e(t)$  is a constant  $c_e$  for any  $t$ , this model coincides with the LT model. Moreover, we can consider the same situations as given in Sect. 3.2.

**Example 5 (Influence maximization with deadline).** Let  $T$  be a positive number. For each edge  $e$ , define a function  $q_e$  to be  $q_e(t) = c_e$  if  $t \leq T$  and  $q_e(t) = 0$  if  $t > T$ , where  $c_e \in [0, 1]$ . The TV-LT model with such a function  $q_e$  represents that the influence will expire at time  $T$ .

**Example 6 (Continuous-time diffusion model).** For each edge  $e = (u, v)$ , we are given the time-delay parameters  $r_e > 0$  and the power of influence  $c_e \in (0, 1)$ . Consider the TV-LT model where  $q_e(t) = c_e$  and  $f_e(t - t_u) = r_e \exp(-r_e(t - t_u))$  for each edge  $e$ . The model is a continuous-time variant of the LT model, in which the time-delay on edges occurs based on exponential distribution.

Let  $\sigma(A)$  be the expected number of vertices activated after running the process of the TV-LT model with an initial seed set  $A$ . Using a technique similar to Theorem 1,  $\sigma$  is shown to be monotone and submodular. As a corollary, an optimal solution for the influence maximization problem under the TV-LT model can be approximated efficiently within the ratio  $(1 - 1/e - \epsilon)$ .

**Theorem 2.** *For the TV-LT model,  $\sigma$  is a monotone submodular function.*



## 5 Scalable Greedy Algorithms for the Proposed Models

In this section, we propose scalable greedy algorithms for influence maximization under the proposed diffusion models by extending a sketching method [2]. We first describe the sketching method and its generalization, and then discuss how to extend it to the proposed models.

### 5.1 Sketching Method and Generalization

The pseudocode of the sketching method is presented in Algorithm 1. Given a directed graph  $G = (V, E)$ , a diffusion model  $\mathcal{M}$ , and a seed size  $k$ , the sketching method performs the following two stages. In the first stage, beginning with an empty family  $\mathcal{R} = \emptyset$ , it repeats the following procedure: sample a *target* vertex  $z$  from  $V$  uniformly at random, compute the vertex set  $R$  that would influence  $z$  in an outcome of the diffusion process of  $\mathcal{M}$ , and add  $R$  to  $\mathcal{R}$ . The above repetition terminates when  $\mathcal{R}$  includes a sufficient number of vertex sets for accurate influence estimation. In the second stage, it computes an approximate solution  $A$  of the *maximum coverage problem*, which seeks to select a set of  $k$  vertices from  $V$  that intersects the maximum number of vertex sets in  $\mathcal{R}$ , by the greedy algorithm. Finally, it returns a solution  $A$ .

Here we discuss why  $A$  is influential. Let  $F_{\mathcal{R}}(A)$  be the fraction of sets in  $\mathcal{R}$  intersecting  $A$ , i.e.,  $F_{\mathcal{R}}(A) = \frac{|\{R \in \mathcal{R} \mid R \cap A \neq \emptyset\}|}{|\mathcal{R}|}$ . Then, for any vertex set  $A$ ,  $n \cdot F_{\mathcal{R}}(A)$  is an unbiased estimator of  $\sigma(A)$ , i.e.,  $\mathbb{E}[n \cdot F_{\mathcal{R}}(A)] = \sigma(A)$  [2], where  $\sigma(A)$  is the influence spread of  $A$  under  $\mathcal{M}$ . Therefore, as long as this estimator gives accurate influence estimations,  $A$  is likely to have a large influence spread.

Now we consider applying the sketching method to the diffusion models proposed in this paper. There are two main challenges. The first one is to devise a procedure for generating a (random) vertex set that would influence a certain target vertex (line 4 in Algorithm 1) under the proposed models. The second is guaranteeing the accuracy and time complexity of the sketching method with the devised procedure. For the purpose, we adopt *reverse influence (RI) sets*, a model-independent notion introduced by Tang et al. [28], defined as follows.

---

#### Algorithm 1. Sketching method for influence maximization.

---

**Require:** a directed graph  $G = (V, E)$ , a diffusion model  $\mathcal{M}$ , a seed size  $k$ .

- 1:  $\mathcal{R} \leftarrow \emptyset$ . ▷ Building sketches.
  - 2: **repeat**
  - 3:    $z \leftarrow$  a vertex chosen from  $V$  uniformly at random.
  - 4:    $R \leftarrow$  a vertex set that would influence  $z$  in an outcome of the process of  $\mathcal{M}$ .
  - 5:    $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$ .
  - 6: **until**  $\mathcal{R}$  includes a sufficient number of vertex sets for accurate influence estimation.
  - 7:  $A \leftarrow \emptyset$ . ▷ Selecting a seed set.
  - 8: **while**  $|A| < k$  **do**
  - 9:    $s \leftarrow \operatorname{argmax}_{v \in V} F_{\mathcal{R}}(v)$ . ▷  $F_{\mathcal{R}}(v)$  is defined as  $\frac{|\{R \in \mathcal{R} \mid R \cap \{v\} \neq \emptyset\}|}{|\mathcal{R}|}$ .
  - 10:    $A \leftarrow A \cup \{s\}$  **and** remove vertex sets including  $s$  from  $\mathcal{R}$ .
  - 11: **return**  $A$ .
-

**Definition 1 (Reverse influence set from Definition 3 in [28]).** For a graph  $G = (V, E)$  and a diffusion model  $\mathcal{M}$ , a reverse influence (RI) set for a vertex  $z$  in  $V$  is a random vertex set  $R \subseteq V$  such that for any vertex set  $S \subseteq V$ , the probability that  $R \cap S \neq \emptyset$  is equal to the probability that the initial activation of vertices in  $S$  results in the activation of  $z$  under the diffusion process of  $\mathcal{M}$ . A random RI set is defined as an RI set for a vertex randomly sampled from  $V$ .

Thus, if we are given a family  $\mathcal{R}$  of random RI sets for  $\mathcal{M}$ , then we have  $n \cdot F_{\mathcal{R}}(S) = \sigma(S)$  for every set  $S$  [28]. Furthermore, given a procedure for generating random RI sets under  $\mathcal{M}$ , Tang et al. [28] proved the time complexity and approximation ratio of a sketching algorithm IMM [28] shown as follows.

**Theorem 3 (Theorem 5 in [28]).** Under a diffusion model for which a random RI set takes  $O(\text{EPT})$  expected time to generate, IMM returns a  $(1 - 1/e - \epsilon)$ -approximation with probability at least  $1 - \frac{1}{n^\epsilon}$ , and runs in  $O(\frac{\text{EPT}}{\text{OPT}}(k + \ell)(n + m)^{\frac{\log n}{\epsilon^2}})$  expected time, where  $\text{OPT} = \max_{S \subseteq V: |S|=k} \sigma(S)$ .

In summary, it suffices to design efficient and correct computation of RI sets. Remark that such a procedure may not exist depending on  $\mathcal{M}$ . In the following, we describe an algorithm that produces RI sets under each proposed model and analyze its correctness and computation time.

## 5.2 Efficient RI Set Generation Under TV-IC Model

*Algorithm Description.* Here we describe an efficient algorithm for generating RI sets under the TV-IC model. Note that existing approaches for RI set generation, such as a BFS-like algorithm for the IC and LT models [2, 28, 29] and a Dijkstra-like algorithm for the CTIC model [28], cannot be applied to the TV-IC model due to the time dependency of probability.

For this purpose, we exploit the graph introduced in the proof of Theorem 1. Given the choice of  $d_e$ 's and  $x_e$ 's, a target vertex  $z$  will be activated in the diffusion process with an initial seed vertex  $v$  if  $v$  can reach  $z$  in  $G_X(\{v\})$ , which is obtained by the procedure discussed in Sect. 3.4. However, a naive implementation of the procedure requires at least quadratic time.

We now present a more efficient algorithm. The key idea is to introduce the *latest activation time*  $\tau[v]$  of  $v$ , which is defined as the maximum number  $\tau[v]$  such that the activation of  $v$  within time  $\tau[v]$  results in the activation of  $z$  given the choice of  $x_e$ 's and  $d_e$ 's. Obviously,  $\tau[z] = +\infty$ . For each vertex  $u$  ( $\neq z$ ),  $u$ 's influence must pass through one of its out-going edges in order to influence  $z$ . Specifically,  $u$  influences  $z$  by passing through  $(u, v)$  if  $u$  was activated within time  $\tau[v] - d_{uv}$  and  $p_{uv}(\tau[u] + d_{uv}) > x_{uv}$ . Thus, the latest activation time  $\tau[u]$  of  $u$  is determined by

$$\tau[u] = \max_{v \in N^+(u)} \min\{\tau[v] - d_{uv}, p_{uv}^{-1}(x_{uv})\},$$

where  $p_{uv}^{-1}(x_{uv})$  is the maximum number  $t$  such that  $p_{uv}(t + d_{uv}) > x_{uv}$  (note that  $p_e^{-1}(x)$  can be  $\pm\infty$ ). From the equation, the values  $\tau[v]$  for all vertices  $v$  can be obtained efficiently by performing *dynamic programming*.

**Algorithm 2.** Efficient RI set generation under the TV-IC model.

---

**Require:** a directed graph  $G = (V, E)$ , edge probability functions  $p_e : \mathbb{R}_+ \rightarrow [0, 1]$ , edge length likelihoods  $f_e : \mathbb{R} \rightarrow [0, 1]$ , a target vertex  $z$ .

- 1:  $\tau[z] \leftarrow +\infty$  **and**  $\tau[v] \leftarrow -\infty$  **for all**  $v \in V \setminus \{z\}$ .
- 2:  $Q \leftarrow$  a queue with only one element  $z$ .
- 3: **while**  $Q \neq \emptyset$  **do**
- 4:   Dequeue  $v$  with the maximum  $\tau[v]$  from  $Q$ .
- 5:   **for all**  $u$  with  $e = (u, v) \in E$  **do**
- 6:      $d_e \leftarrow$  an edge length sampled according to  $f_e$ .
- 7:      $x_e \leftarrow$  a uniform random number in  $[0, 1]$ .
- 8:      $\rho \leftarrow \min\{\tau[v] - d_e, p_e^{-1}(x_e)\}$ .
- 9:     **if**  $\rho > \tau[u]$  **and**  $\rho \geq 0$  **then**
- 10:        $\tau[u] \leftarrow \rho$  **and** enqueue  $u$  onto  $Q$ .  $\triangleright O(\log n)$  time
- 11: **return** the set of visited vertices, i.e.,  $\{v \in V \mid \tau[v] \geq 0\}$ .

---

The pseudocode of the RI set generation under the TV-IC model is given in Algorithm 2. Beginning with a queue with a target vertex  $z$  with  $\tau[z] = +\infty$ , we determine the latest activation time of each vertex iteratively. For each iteration, we extract a vertex  $v$  with the maximum  $\tau[v]$  ( $\geq 0$ ) from the queue (thereafter,  $v$ 's latest activation time will not be updated), sample a random number  $x_{uv}$  and an edge length  $d_{uv}$  of each vertex  $u$  in the in-neighbors of  $v$ , and update its latest activation time  $\tau[u]$  if  $\min\{\tau[v] - d_{uv}, p_{uv}^{-1}(x_{uv})\} > \tau[u]$ . When  $\tau[u] \geq 0$  at that time, we insert  $u$  into the queue. When the queue is empty, we return the set of vertices  $v$  with  $\tau[v] \geq 0$  as an RI set for  $z$ . Note that by using a binary heap, both selecting a vertex from the queue (line 4) and inserting a vertex into the queue (line 10) can be performed in  $O(\log n)$  time.

*Theoretical Analysis.* We first give the correctness and time complexity.

**Lemma 2.** *Algorithm 2 produces an RI set for  $z$  for the TV-IC model.*

*Proof.* We show that for any vertex  $z$  and any vertex set  $S \subseteq V$ , the probability  $p_1$  that the algorithm's output intersects  $S$  is equal to the probability  $p_2$  that the initial activation of vertices in  $S$  leads to the activation of  $z$ .

From the construction of the algorithm,  $p_1$  is the probability of the following event over the choice of  $x_e$ 's and  $d_e$ 's: For some vertex  $s$  in  $S$ , there is a path  $v_1 = s, v_2, \dots, v_{\ell-1}, v_\ell = z$  of length  $\ell$  such that  $\tau_1 \geq 0$  where  $\tau_\ell = +\infty$  and  $\tau_i = \min\{\tau_{i+1} - d_{v_i v_{i+1}}, p_{v_i v_{i+1}}^{-1}(x_{v_i v_{i+1}})\}$  ( $1 \leq i \leq \ell - 1$ ).

From the procedure to obtain  $G_X(A)$  in Sect. 3.4,  $p_2$  is the probability of the following event over the choice of  $x_e$ 's and  $d_e$ 's: For some vertex  $s$  in  $S$ , there is a path  $v_1 = s, v_2, \dots, v_{\ell-1}, v_\ell = z$  of length  $\ell$  such that  $p_{v_i v_{i+1}}(\tau'_i + d_{v_i v_{i+1}}) > x_{v_i v_{i+1}}$  ( $1 \leq i \leq \ell - 1$ ) where  $\tau'_1 = 0$  and  $\tau'_{i+1} = \tau'_i + d_{v_i v_{i+1}}$  ( $1 \leq i \leq \ell - 1$ ).

It is easy to see that the two events given the choice of  $x_e$ 's and  $d_e$ 's are equivalent. Therefore,  $p_1 = p_2$  and thus the lemma holds.  $\square$

**Lemma 3.** *Algorithm 2 runs in  $O(\frac{m \cdot \text{OPT}}{n} \log n)$  expected time for a randomly selected vertex  $z$ .*

Then, by Theorem 3 and Lemmas 2 and 3, we obtain the following.

**Theorem 4.** *Under the TV-IC model, IMM with Algorithm 2 returns a  $(1 - 1/e - \epsilon)$ -approximation with probability at least  $1 - \frac{1}{n^\ell}$  and runs in  $O((k + \ell)(m + \frac{m^2}{n}) \frac{\log^2 n}{\epsilon^2})$  expected time.*

Although a factor  $m^2/n$  in the time complexity can be  $O(m\sqrt{m})$  for dense graphs, real-world social networks are sparse, i.e.,  $m/n$  is small, and thus the proposed algorithm scales approximately linearly to real-world social networks.

### 5.3 Efficient RI Set Generation Under TV-LT Model

Similar to the TV-IC model, we develop an efficient algorithm for generating random RI sets under the TV-LT model and obtain the following theorem.

**Theorem 5.** *Under the TV-LT model, IMM with the above procedure for RI set generation returns a  $(1 - 1/e - \epsilon)$ -approximation with probability at least  $1 - \frac{1}{n^\ell}$  and runs in  $O((k + \ell)(m + \frac{m^2}{n}) \frac{\log^2 n}{\epsilon^2})$  expected time.*

## 6 Experimental Evaluations

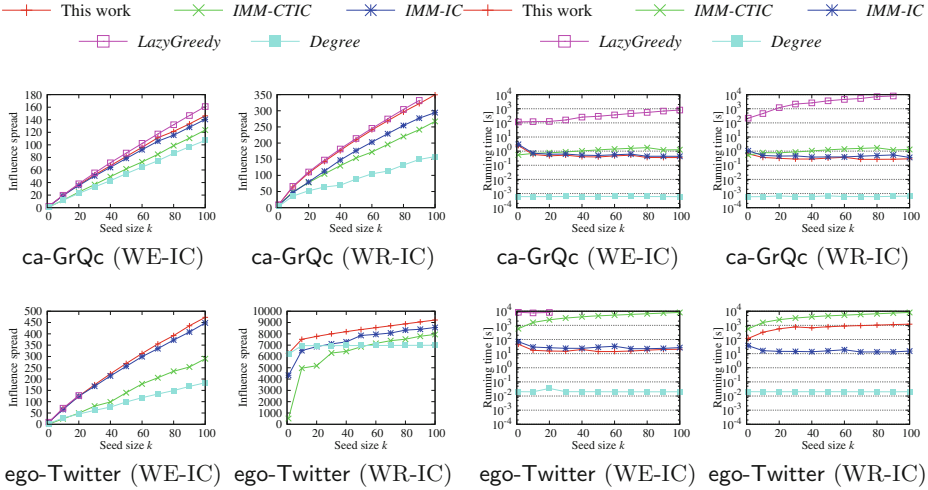
In this section, we demonstrate the efficiency and accuracy of our algorithms through experiments on real-world networks. We conducted the experiments on a Linux server with an Intel Xeon E5540 2.53 GHz CPU and 48 GB memory. All algorithms were implemented in C++ and compiled using g++ 4.8.2 with the -O2 option. We used five real-world social networks (Table 1).

### 6.1 Experiments with TV-IC Model

*Settings of Edge Probability Functions and Edge Length Likelihoods.* Motivated by the empirical evidence shown in Fig. 1, we adopt two nonincreasing functions for edge probabilities. One is the *weighted exponential (WE)* IC model, which assigns  $p_{uv}(t) = \frac{1}{d^-(v)} \exp(-ct)$  to each edge  $(u, v)$ , where  $c$  is sampled randomly in the range  $[1, 10]$ . Here  $d^-(v)$  is the in-degree of a vertex  $v$ . The other is the *weighted reciprocal (WR)* IC model, which assigns  $p_{uv}(t) = \frac{1}{d^-(v)ct}$ , where  $c$  is sampled randomly in the range  $[1, 10]$ . Note that these models represent fast and

Table 1. Datasets.

Dataset	$n$	$m$	Type
Physicians [1]	241	1, 098	Social
ca-GrQc [21]	5, 241	28, 968	Collaboration
wiki-Vote [21]	7, 115	103, 689	Social
soc-Epinions1 [21]	75, 879	508, 837	Social
ego-Twitter [21]	81, 306	2, 420, 744	Social



**Fig. 3.** Influence spreads for TV-IC.

**Fig. 4.** Running times for TV-IC.

slow decay of the power of influence, respectively. We show that such differences in the speed of time-decaying are crucial to the expected size of the cascades.

For each edge  $e$ , we set the edge length likelihood to the Weibull distribution [19], whose probability distribution function is defined as:

$$f_e(\delta) = \frac{\alpha_e}{\beta_e} \cdot \left(\frac{\delta}{\beta_e}\right)^{\alpha_e-1} \cdot \exp\left(-\left(\frac{\delta}{\beta_e}\right)^{\alpha_e}\right), \quad (2)$$

where  $\alpha_e$  and  $\beta_e$  are randomly sampled in the range  $[0, 10]$ . Note that this distribution has been adopted in continuous-time diffusion model literature [9, 28].

*Comparative Algorithms.* For the proposed algorithm for the TV-IC model, i.e., IMM with Algorithm 2, we set  $\epsilon = 0.5$  and  $\ell = 1$ , as described in [28]. Here we compare the proposed algorithm with the following baseline algorithms.

- *LazyGreedy* [23]: An accelerated simulation-based greedy algorithm for monotone submodular function maximization. We conducted Monte-Carlo simulations 10,000 times to estimate the influence spread.
- *IMM-CTIC* [28]: A sketching method for the CTIC model. Since this method takes care of “deadlines” rather than time-decaying edge probabilities, we set its deadline to 1.
- *IMM-IC* [28]: A sketching method for the IC model. We set the probability of each edge  $e$  to  $p_e(\bar{d}_e)$ , where  $\bar{d}_e$  is the average edge length.
- *Degree*: Select  $k$  vertices in decreasing degree order.

*Results.* Figure 3 shows the influence spreads for seed sets of sizes 1, 10, 20, . . . , 100 computed by each algorithm.<sup>2</sup> We omitted the results for Physicians, wiki-Vote, and soc-Epinions1, which exhibit similar behaviors, due to space limitations.

<sup>2</sup> We take the average after conducting simulations of the TV-IC 10,000 times.

*LazyGreedy* did not finish in 10,000 seconds with *ca-GrQc* (WR-IC,  $k = 100$ ), *ego-Twitter* (WE-IC,  $k \geq 30$ ), and *ego-Twitter* (WR-IC). Consequently, we were unable to obtain seed sets with these settings. Our method and *LazyGreedy* returned nearly the best results for most settings. Although *IMM-IC* is close to the best results, its influence spread ( $=4,336$ ) with *ego-Twitter* (WR-IC,  $k = 1$ ) is 30 % worse than the best ( $=6,279$ ). *IMM-CTIC* provided ineffective seed sets, e.g., with *ego-Twitter* (WR-IC,  $k = 1$ ). As expected, *Degree* gave poor seed sets. We can also see that the WR-IC setting gives larger influence spreads compared to the WE-IC setting, which demonstrates the critical importance of the time-decaying phenomenon.

Figure 4 shows the running times required to select seed sets of sizes 1, 10, 20,  $\dots$ , 100 for each algorithm. Note that the running times do not include the time required to read the input graph from secondary storage. *LazyGreedy* even did not finish in 10,000 seconds with *ca-GrQc* ( $k = 100$ ), which is a small network, due to the computation cost of the Monte-Carlo simulations. Our method and *IMM-IC* required only several thousands of seconds for each graph, which is several orders of magnitude faster than *LazyGreedy*.

## 6.2 Experiments for TV-LT Model

*Settings of Edge Weight Functions and Edge Length Likelihoods.* Similar to the TV-IC model, we adopt two nonincreasing functions for edge weights, i.e., the *weighted exponential* (WE) LT model, which assigns  $q_{uv}(t) = \frac{1}{d^-(v)} \exp(-ct)$ , and the *weighted reciprocal* (WR) LT model, which assigns  $q_{uv}(t) = \frac{1}{d^-(v)c(t+1)}$ , where  $c$  is randomly sampled in the range  $[1, 10]$ .

We set the edge length likelihood to the Weibull distribution (2).

*Comparative Algorithms.* For the proposed algorithm for the TV-LT model, i.e., *IMM* with the algorithm in Sect. 5.3, we set  $\epsilon = 0.5$  and  $\ell = 1$  [28]. Since there are no algorithms for continuous-time LT models, we compare our method to *LazyGreedy* [23] and *Degree*, the same as for the TV-IC model, and *IMM-LT* [28], which is a sketching method for the LT model with edge weights  $q_e(\bar{d}_e)$ .

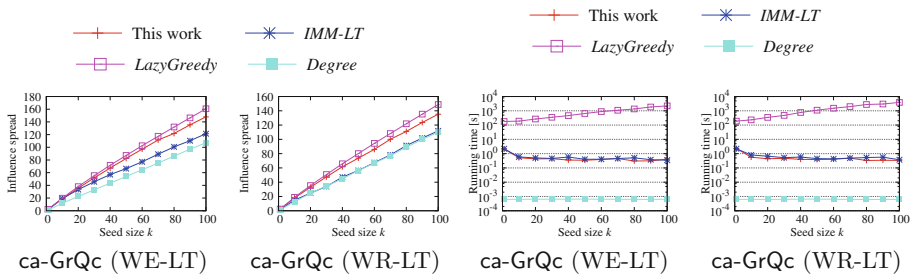


Fig. 5. Influence spreads for TV-LT.

Fig. 6. Running times for TV-LT.

*Results.* Figure 5 shows the influence spreads for seed sets computed by each algorithm. We observe similar behaviors as TV-IC model, *LazyGreedy* gave the best solutions, and the proposed method significantly outperformed *IMM-LT* and *Degree* for most settings.

Figure 6 shows the running times required to select seed sets for each algorithm. As in the case of the TV-IC model, the proposed method has much better scalability than *LazyGreedy*.

## 7 Conclusions

In this paper, we proposed diffusion models that incorporate time-decaying phenomenon and time-delay propagation by generalizing two standard diffusion models, i.e., independent cascade and linear threshold. We demonstrated that our models include most previous models with temporal effects, and the influence functions are monotone and submodular. Moreover, we devised scalable algorithms for influence maximization under the proposed models and experimentally verified their efficiency and accuracy compared to baseline algorithms.

A possible future direction is to learn edge probability functions from cascade logs. It might also be interesting to consider the influence maximization over diffusion models where cascades may recur [6], i.e., the power of influence is not necessarily nonincreasing. Note, however, that the case does not fall into submodular maximization as described in Sect. 3.3.

**Acknowledgments.** N. O. was supported by JSPS Grant-in-Aid for JSPS Research Fellow Grant Number 16J09440. Y. Y. was supported by JSPS Grant-in-Aid for JSPS Fellows Grant Number 13J02522. N. K. was partially supported by JSPS KAKENHI Grant Numbers 25730001, 24106002.

## References

1. Physicians network dataset - KONECT, March 2016. [http://konect.uni-koblenz.de/networks/moreno\\_innovation](http://konect.uni-koblenz.de/networks/moreno_innovation)
2. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: SODA, pp. 946–957 (2014)
3. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. In: AAAI, pp. 592–598 (2012)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD, pp. 1029–1038 (2010)
5. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: KDD, pp. 199–208 (2009)
6. Cheng, J., Adamic, L.A., Kleinberg, J.M., Leskovec, J.: Do cascades recur? In: WWW, pp. 671–681 (2016)
7. Cui, B., Yang, S.J., Homan, C.: Non-independent cascade formation: temporal and spatial effects. In: CIKM, pp. 1923–1926 (2014)
8. Domingos, P., Richardson, M.: Mining the network value of customers. In: KDD, pp. 57–66 (2001)

9. Du, N., Song, L., Gomez-Rodriguez, M., Zha, H.: Scalable influence estimation in continuous-time diffusion networks. In: NIPS, pp. 3147–3155 (2013)
10. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: a complex systems look at the underlying process of word-of-mouth. *Market. Lett.* **12**(3), 211–223 (2001)
11. Gomez-Rodriguez, M., Balduzzi, D., Schölkopf, B.: Uncovering the temporal dynamics of diffusion networks. In: ICML, pp. 561–568 (2011)
12. Gomez-Rodriguez, M., Schölkopf, B.: Influence maximization in continuous time diffusion networks. In: ICML, pp. 313–320 (2012)
13. Goyal, A., Bonchi, F., Lakshmanan, L.V.: Learning influence probabilities in social networks. In: WSDM, pp. 241–250 (2010)
14. Granovetter, M.: Threshold models of collective behavior. *Am. J. Sociol.* **83**(6), 1420–1443 (1978)
15. Iribarren, J.L., Moro, E.: Impact of human activity patterns on the dynamics of information diffusion. *Phys. Rev. Lett.* **103**, 038702 (2009)
16. Karsai, M., Kivela, M., Pan, R.K., Kaski, K., Kertész, J., Barabási, A.L., Saramäki, J.: Small but slow world: how network topology and burstiness slow down spreading. *Phys. Rev. E* **83**, 025102 (2011)
17. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
18. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS, vol. 4213, pp. 259–271. Springer, Heidelberg (2006). doi:[10.1007/11871637\\_27](https://doi.org/10.1007/11871637_27)
19. Lawless, J.F.: *Statistical Models and Methods for Lifetime Data*. Wiley-Interscience, New York (2002)
20. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD, pp. 420–429 (2007)
21. Leskovec, J., Krevl, A.: SNAP datasets: stanford large network dataset collection, June 2014. <http://snap.stanford.edu/data>
22. Liu, B., Cong, G., Xu, D., Zeng, Y.: Time constrained influence maximization in social networks. In: ICDM, pp. 439–448 (2012)
23. Minoux, M.: Accelerated greedy algorithms for maximizing submodular set functions. In: Stoer, J. (ed.) *Optimization Techniques*. Lecture Notes in Control and Information Sciences, vol. 7, pp. 234–243. Springer, Heidelberg (1978)
24. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of the approximations for maximizing submodular set functions. *Math. Program.* **14**, 265–294 (1978)
25. Ohsaka, N., Akiba, T., Yoshida, Y., Kawarabayashi, K.: Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In: AAAI, pp. 138–144 (2014)
26. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 322–337. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-05224-8\\_25](https://doi.org/10.1007/978-3-642-05224-8_25)
27. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Selecting information diffusion models over social networks for behavioral analysis. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS, vol. 6323, pp. 180–195. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-15939-8\\_12](https://doi.org/10.1007/978-3-642-15939-8_12)
28. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: A martingale approach. In: SIGMOD, pp. 1539–1554 (2015)
29. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: near-optimal time complexity meets practical efficiency. In: SIGMOD, pp. 75–86 (2014)