JIS

Article

Time-Sensitive Influence Maximization in Social Networks

Journal of Information Science
1–15

DOI: 10.1177/0165551510000000
jis.sagepub.com
SAGE

Azadeh Mohammadi
Department of Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Mohamad Saraee
School of Computing, Science and Engineering, University of Salford-Manchester, Manchester, United Kingdom

Abdolreza Mirzaei
Department of Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Abstract
One of the fundamental issues in social networks is the influence maximization problem, where the goal is to identify a small subset of individuals such that they can trigger the largest number of members in the network. In real-world social networks, the propagation of information from a node to another may incur a certain amount of time delay; moreover, the value of information may decrease over time. So not only the coverage size, but also the propagation speed matters.
In this paper, we propose the Time-Sensitive Influence Maximization (TSIM) problem, which takes into account the time dependence of the information value. Considering the time delay aspect, we develop two diffusion models, namely the Delayed Independent Cascade model and the Delayed Linear Threshold model. We show that the TSIM problem is NP-hard under these models but the spread function is monotone and submodular. Thus, a greedy approximation algorithm can achieve a *1-1/e* approximation ratio. Moreover, we propose two time-sensitive centrality measures and compare their performance to the greedy algorithm. We evaluate our methods on four real-world datasets. Experimental results show that the proposed algorithms outperform existing methods, which ignore the decay of information value over time.

Keywords
Approximation analysis; Influence maximization; Information diffusion; Social networks; Time-Sensitive diffusion;

1. Introduction

A social network is made up of social actors (such as individuals or organizations) and the relationships between them. The advent of online social networks in the last decades provides unprecedented opportunities for social network analysis [1]. One of the main research interests in this expanding area is the analysis of influence and information propagation in social networks. Social influence is a well-known phenomenon in networks and is defined as the change in an individual's behaviours, opinions or actions that results from interaction with others [2]. Social influence has broad applications in different fields including viral marketing [3–6], information dissemination [7,8], recommendation systems [9,10], and trust propagation [11,12].

One of the fundamental issues with information propagation is to find a small subset of influential nodes such that they can attract the largest number of members in a social network, according to a diffusion model. This problem is called influence maximization, which is proved to be NP-hard [13,14]. Different approximation algorithms [13–18] and heuristic methods [18–23] have been proposed to solve this problem. However, in most studies, the role of time in influence propagation is ignored. Only some recent works have considered the temporal aspects in information diffusion [24–29]. In [24–26] the underlying graph is inferred from real propagation data by considering a time delay factor. Chen et al. [27] and Liu et al. [28,29] independently studied the influence maximization problem with a deadline constraint.

Corresponding author:
Azadeh Mohammadi, Department of Computer Engineering, Isfahan University of Technology, Isfahan, Iran
azade.mohammadi@gmail.com

In all of the previous works, the aim is to maximize the number of influenced nodes, either until a deadline is reached or without using any time constraint. In fact, previous studies value all activated nodes equally, regardless of their activation time. However, in real-world applications, the value of propagated items may decrease over time. Therefore, a node that is activated later may worth less. As an application, consider the dissemination of hot topics or urgent information. In these situations, the value of propagated information is dependent on activation time of influenced users. For example, in the event of an epidemic it is important to inform more people as soon as possible. As another application, in viral marketing, the sooner a company sells its product, the faster it receives a return on investment. This means that not only the number of activated customers, but also the rate of activation matters.

In this paper, we define the Time-Sensitive Influence Maximization (TSIM) problem, which aims to find a set of individuals whose spread of influence in the network maximizes the total profit. The profit value of an individual is dependent on its activation time and the rate at which information decays during time. The amount of reduction in information value is determined by a freshness function, which specifies the relative importance of coverage and diffusion time in the propagation process. If the freshness value decreases quickly over time, the activation time of the nodes will have a major impact on the spread value. Conversely, if freshness value decreases slowly, the number of activated nodes will be the determining factor for the total profit.

To incorporate time delay aspect, we consider two diffusion models, Delayed Independent Cascade and Delayed Linear Threshold models. We prove the TSIM problem is NP-hard and show that the spread function is monotone and submodular under the aforementioned models. Thus, we develop a greedy algorithm, which can guarantee a spread value within $(1-1/e)$ of the optimal solution. Furthermore, we devise two time-sensitive heuristic methods and compare their performance with conventional centrality measures.

The major contributions of this paper are as follows:

- We propose a new problem called Time-Sensitive Influence Maximization, which takes into account the time dependency of information value. We incorporate freshness concept to the influence maximization problem.
- We prove the NP-hardness of the TSIM problem and propose a greedy algorithm based on submodularity and monotonicity properties of the spread function. The proposed algorithm considers time delay in estimating propagated value.
- We propose two time-sensitive heuristics for influence maximization problem and compare them with the proposed approximation algorithm and conventional methods.
- We verify the effectiveness of our algorithms by conducting experiments on real social networks. The experimental results show that the selected nodes by our methods can achieve higher spread value in comparison to conventional methods.

The rest of the paper is organized as follows. Section 2 reviews the related works. The problem definition and proposed methods are discussed in Section 3. The experimental results and discussion are presented in Section 4. Finally, Section 5 concludes the paper and indicates our future directions.

2. Related works

The influence maximization problem was first introduced by Domingo and Richardson [3,4]. They modeled the problem using Markov random fields and proposed heuristics for identifying influential nodes. Kempe et al. [13,14] formulated influence maximization as a discrete optimization problem. They proved that this problem is NP-hard under two well-known propagation models, namely the Independent Cascade (IC) and Linear Threshold (LT) models. They proposed a greedy algorithm with an approximation ratio of (1-1/e) that successively selects the node with the maximum marginal influence. However, their algorithm is computationally expensive and cannot scale well with large social networks. A number of studies were carried out to address this scalability issue. Leskovec et al. [15] proposed the Cost-Effective Lazy Forward (CELF) algorithm that is a lazy forward version of the greedy algorithm. It uses the submodularity property of the influence function to reduce greatly the number of influence spread evaluations on the nodes. Goyal et al. [16] presented CELF++ based on the CELF algorithm that avoid unnecessary recomputations of marginal gain incurred by CELF.

In contrast to approximation algorithms, several studies proposed heuristic methods to reduce the complexity of influence spread evaluation. Chen et al. [19] proposed a scalable heuristic called LDAG for the LT model, which computes the influence of each node only within its constructed local directed acyclic graph. Also a scalable heuristic algorithm called PMIA is proposed for the IC model, which exploits maximum influence in-arborescence paths to

estimate the influence spread [20]. Another scalable method is IRIE that is proposed under the IC model. It uses a fast iterative ranking algorithm with a fast influence estimation method to achieve scalability while maintaining good influence coverage [21]. All of the above heuristics are proposed for a specific diffusion model; on the other hand, there are centrality heuristics including degree heuristic, betweenness and closeness centrality that are not dependent on the diffusion model [22,23]. Goyal et al. proposed an alternative approach that directly uses past propagation data to estimate expected influence spread [30].

Some recent works have studied the role of time in the diffusion of information. Saito et al. proposed continuous time diffusion model and addressed the problem of estimating the parameters of the model from the observed data [24]. Goyal et al. [25], presented algorithms for learning the parameters of both static and time-dependent models from propagation data. The proposed methods in [26] find the optimal network and transmission rates that maximize the likelihood of temporal propagation data.

Chen et al. [27] and Liu et al. [28,29] independently studied the problem of influence maximization under a deadline constraint. Given a deadline T, the aim of these works is to find a set of nodes that maximizes the expected number of influenced nodes until T. In [27] a meeting event is added to diffusion models that determine the probability of meeting the neighbors in a specified time. Their proposed model is a special case of the model proposed in [28,29]. However, all of the above works ignore the devaluation of propagated items during time and overlook its effect on the influence maximization problem. Therefore, in this paper we incorporate this aspect into the influence maximization problem and define the Time-Sensitive Influence Maximization problem.

3. Material and methods

A social network can be modeled as a directed graph *G=(V, E)*, where *V* is the set of nodes and *E* is the set of edges or relationships. As stated in the previous section, influence maximization is a fundamental problem in social networks. Given the graph *G* and a number *k* as input, the influence maximization problem aims to find a set of *k* influential nodes, called the seed set, such that by activating them, the expected spread of influence according to a diffusion model is maximized. Diffusion models define how the information propagates on the network. Two widely used diffusion models are the Independent Cascade (IC) and Linear Threshold (LT) models. After selecting a set of initial nodes, the items propagate based on the given diffusion model. When a node adopts the item, it will switch from inactive state to active state. In progressive models, once a node becomes active it will remain so forever.

It is shown in [13] that the influence maximization problem is NP-hard under IC and LT models, but a greedy algorithm is employed that achieves *(1-1/e)* approximation by exploiting the monotonicity and submodularity properties of the influence function. Submodularity and monotonicity are two key theoretical properties for optimization problem and are defined as follows.

Theorem 1. Given a ground set *V*, a function $f: 2^V \to R$ is called monotone if $f(S) \le f(T)$ for all subsets $S \subseteq T \subseteq V$ [31].

In other words, adding an element to the input of a monotone function does not decrease its value.

Theorem 2. Given a ground set *V*, a function $f: 2^V \to R$ is called submodular if it satisfies $f(S \cup \{x\}) - f(S) \ge f(T \cup \{x\}) - f(T)$ for all subsets $S \subseteq T \subseteq V$ and all $x \in V / T$ [31].

Intuitively, a function is submodular if it satisfies the diminishing return property. This property states that the marginal gain from adding an element to a set *S* is at least as high as the marginal gain from adding that element to the superset *T*.

The conventional influence maximization problem does not consider the time sensitivity of information. Thus, to incorporate time sensitivity of propagation process, we extend the IC and LT models to Delayed Independent Cascade (DIC) model and Delayed Linear Threshold (DLT) model and propose the Time-Sensitive Influence Maximization (TSIM) problem. In the next subsection, we formally define TSIM and study its properties under the DIC and DLT models.

3.1. Problem definition and properties

As we mentioned earlier, the value of propagating items may decrease over time. The amount of reduction in the value of propagated items is determined by a freshness function that is given as an input. The freshness function *ff(t)* specifies how the value of an item varies with time. It can be any positive decreasing function of time such as exponential, polynomial or, piecewise linear function. The value of a node is determined by the freshness function, giving as input the activation time of that node. Thus, we define the value of node *v* as *fval(v)=ff(v.acttime)*, where *v.acttime* is the activation time of *v*.

Taking into account the diminishing value of propagating items, we formally define TSIM problem as follows.

Definition 1. Given a social network *G(V, E)*, a diffusion model *m*, a positive integer *k*, and a freshness function *ff(t)*, the TSIM problem is to find a set *S* of *k* seed nodes such that *PV(S)* is maximized under diffusion model *m*, i.e., finding $S^* = \arg\max_{S \subseteq V, |S| \leq K} PV(S)$.

In the above definition, *PV(S)* is the spread function that is equal to the expected value of activated nodes given *S*. Formally we define it as:

$$PV : 2^V \rightarrow R^+,\ PV(S) = E\left[\sum_{i \in A(S)} fval(i)\right] \quad (1)$$

where *A(S)* is the set of activated nodes at the end of diffusion process initiated by *S* and *fval(i)* is the value of activated node *i*.

In the following sections, we discuss the properties of the TSIM problem under DIC and DLT models.

3.1.1. TSIM problem under DIC model

In the IC model, each edge $(u,v) \in E$ is associated with a propagation probability $p_{uv} \in [0,1]$, which represents the probability that *u* activates its neighbor *v*. By activating a set of initial nodes at *t=0*, the items propagate in discrete time steps as follows. Let Act_{cur} be the set of nodes activated at current step. At any time step, each node $u \in Act_{cur}$ has a single chance to activate each of its inactive out-neighbors *v* with probability p_{uv}. The propagation process terminates if and only if $Act_{cur} = \varnothing$.

As discussed before, in real-world social networks the propagation of influence from one node to another may incur a certain amount of time delay. Consequently, we extend the IC model to the Delayed Independent Cascade (DIC) model, which takes into account the propagation delay. As in the IC, each edge $(u,v) \in E$ in the DIC model is associated with a propagation probability, p_{uv}; In addition, a time delay d_{uv} is assigned to edge *(u,v)* that represents the amount of time it takes for *u* to influence *v*.

In contrast to the IC model, each node in the DIC model can be in one of three different states, active, inactive, or latent active. A set of nodes are selected as seed at *t=0*. By activating these nodes, the items propagate in discrete time steps as follows. At any time step *t*, each node $u \in A_t$ has a single chance to activate each of its inactive or latent active out-neighbor *v* with probability p_{uv}. If *v* is in inactive state, it switches to latent active and its activation time is set to $t + d_{uv}$, which means *v* will be active at $t + d_{uv}$. If *v* is already in latent active state, its activation time will be updated with the minimum of its current activation time and $t + d_{uv}$. In the TSIM problem, the values of activated nodes are computed based on their activation time and freshness function. Thus, the Propagation Value (*PV*) is equal to the sum of the value of activated nodes at the end of propagation. The propagation process terminates if and only if $Act_{cur} = \varnothing$ and there is no node in the latent active state. Also, if the freshness function value is equal to *0* at current step, there will be no need to continue the propagation process because *PV* will not change after that. In the following, we examine the TSIM problem and its properties under this model.

Theorem 3. The TSIM problem is NP-hard under the DIC model.

Proof. The conventional influence maximization problem under IC model is a special case of TSIM problem with *ff(t)=1*. Since the influence maximization problem is known to be NP-hard for the IC model [12], TSIM must be NP-hard under the DIC model as well. □

Theorem 4. The spread function *PV(.)* is monotone and submodular under the DIC model.

Proof. Each edge $(u,v) \in E$ is associated with p_{uv} and d_{uv}, which respectively determine the influence probability and influence delay between *u* and *v*. Influence diffusion on this probabilistic graph under DIC model is a stochastic process. The resulting diffusion on this graph can be seen as a set of possible worlds. Each possible world is a deterministic instantiation of the random graph. More specifically, a determined propagation graph is generated by flipping a coin with bias p_{uv} for each edge $(u,v) \in E$ and determining d_{uv}. If coin flip succeeds, the edge is declared as live, otherwise it is declared as blocked. The edge delay determines the value of activated out-neighbor. Therefore, a deterministic instantiation of the graph, where the live edges are preserved and blocked edges are removed, indicates one possible world, *w*. The set of all possible worlds is denoted by *W*.

Taking the expectation over all possible worlds, we can rewrite *PV(S)* as:

$$PV(S) = \sum_{w \in W} pr(w).PV_w(S) \tag{2}$$

where *pr(w)*is the probability of *w* and $PV_w(S)$ is the value of activated nodes given *S* over the determined graph *w*. Based on this redefinition, we prove the monotonicity and submodularity of *PV(.)*.

Let *S* and *T* be two arbitrary sets such that $S \subseteq T \subseteq V$. We say a node *v* is reachable from a seed set *S* if and only if at least one path of live edges exists between a node in *S* and *v*. The value of each reachable node is computed by the freshness function based on the sum of the delays on the shortest live path from *S* to *v*. Let $x \in V \setminus T$ be an arbitrary node. If $\mathrm{U}_{u \in S} R_w(u)$ shows the set of reachable nodes from *S* over deterministic graph *w* and $x \in \mathrm{U}_{u \in S} R_w(u)$, then the source of the live path to *x* also exists in *T*. The shortest path from *T* to *v* is no longer than the shortest path from *S* to *v*. It means the value of reachable nodes from *T* cannot be less than the value of reachable nodes from *S*, therefore $PV_w(.)$ is monotone. Since *PV(.)* is a non-negative linear combination of monotone functions it is monotone as well.

To verify the submodularity property, let *S* and *T* be two arbitrary sets such that $S \subseteq T \subseteq V$, and let $x \in V \setminus T$ be an arbitrary node. In a deterministic graph *w*, $PV_w(S \cup \{x\}) - PV_w(S)$ is the value of nodes reachable from x , but not from *S*. Similarly, $PV_w(T \cup \{x\}) - PV_w(T)$ is the value of nodes reachable from x , but not from *T*. As said before, the value of activated nodes is computed based on the time delay of the shortest live path to each node. Since $S \subseteq T$, the value of reachable nodes from *T* is greater than or equal to the value of reachable nodes from *S;* consequently, we have $PV_w(S \cup \{x\}) - PV_w(S) \geq PV_w(T \cup \{x\}) - PV_w(T)$ **.** It means $PV_w(.)$ is submodular. Since submodularity is closed under nonnegative linear combinations [31], *PV(.)* is submodular as well, and the theorem is proved. □

3.1.2. TSIM problem under DLT model

In the LT model, each edge $(u,v) \in E$ is associated with an influence weight $w_{uv} \in [0,1]$, such that $\sum_{u \in N^{in}(v)} w_{uv} \leq 1$. The weight w_{uv} represents the strength of the influence exerted by *u* on *v*. In addition, each node *v*, has a threshold $\theta_v \in [0,1]$ chosen uniformly at random. By activating a set of initial nodes at *t=0*, the items propagate in discrete time steps as follows. Let Act_{cur} be the set of nodes activated at current step. At any time step *t*, each inactive node becomes activated if the weighted sum of its active in-neighbors exceeds its threshold, i.e., $\sum_{u \in AN_v} w_{uv} \geq \theta_v$, where AN_v represents active in-neighbors of *v* until time step *t*. The propagation process terminates at step *t*, if and only if $Act_{cur} = \varnothing$.

As discussed before, in real-world social networks the propagation of influence from one node to another may incur a certain amount of time delay. Consequently, we extend the LT model to the Delayed Linear Threshold (DLT) model, which takes into account the propagation delay. In the DLT model, in addition to threshold and influence weights, each edge $(u,v) \in E$ is associated with a time delay d_{uv} that represents the amount of time it takes for node *u* to influence *v*.

In contrast to the LT model, each node in the DLT model can be in one of three different states, active, inactive, or latent active. A set of nodes are selected as seed at *t=0*. Each active node *u* affects its out-neighbor *v* after delay d_{uv}. Hence, if *u* becomes active at time step *t*, its impact time for *v* will be $t+d_{uv}$. For each node *v*, its active in-neighbors and their impact time are saved in a list called AN_v. If *v* is in inactive state and the weighted sum of its active in-neighbors becomes greater than or equal to θ_v , it will switch to latent active state. A latent active node will switch to active state at time *t* if the weighted sum of its effective in-neighbors until time *t* exceeds its threshold, i.e., $\sum_{\{u|(u,tiu) \in ANv \wedge tiu \leq t\}} w_{uv} \geq \theta_v$, where *u* is an active in-neighbor of *v* and ti_u is the impact time of *u* on *v*.

In TSIM problem, the values of activated nodes are computed based on their activation time and freshness function. Thus, the Propagation Value (*PV*) is equal to sum of the value of activated nodes at the end of propagation. The propagation process terminates, if and only if $Act_{cur} = \varnothing$ and there is no node in the latent active state. Also if at step *t*, the freshness function value is equal to 0, there will be no need to continue the propagation process because *PV* will not change after that. In the following, we examine the TSIM problem and its properties under this model.

Theorem 5. The TSIM problem is NP-hard under the DLT model.

Proof. The conventional influence maximization problem under the LT model is a special case of the TSIM problem with *ff(t)=1*. Since the influence maximization problem is known to be NP-hard for the LT model [13], TSIM must be NP-hard under the DLT model as well. □

Theorem 6. The spread function *PV(.)* is monotone and submodular under the DLT model.

Proof. Each edge $(u,v) \in E$ is associated with w_{uv} and d_{uv}, which respectively determine the influence weight and influence delay between *u* and *v*. We consider an instance of the DLT model where each node *v* randomly picks one of its incoming edges, say *(u,v)*, with probability w_{uv} and picks no edge with probability $1-\sum_{u\in N^{in}(v)} w_{uv}$. In such a case, for each node, at most one incoming edge is chosen. The selected edges are declared as live and all other edges are declared as blocked.

Kempe et.al. [13] showed that the Linear Threshold model is equivalent to reachability via live-edge paths as defined above. In our case, instead of computing the number of activated nodes, the value of nodes is calculated based on their activation time. Therefore, we incorporate the time delay aspect to live edges. In this case, the effective in-neighbors of node *v* at time *t* are those that their impact time on *v* is smaller than *t*.

Let A_t be the set of nodes that are active at time *t*. Suppose $v \in V \setminus A_t$. In the DLT model, the probability that *v* becomes active at *t+1* is equal to the probability that the total weight of effective neighbors of *v* exceeds θ_v , given that the threshold was not exceeded already. We call this probability $p_{DLT}(v,t+1)$ and it is equal to:

$$p_{DLT}(v,t+1) = \frac{\sum_{u\in eff_{t+1}(v)\setminus eff_t(v)} w_{uv}}{1-\sum_{u\in eff_t(v)} w_{uv}} \tag{3}$$

where $eff_t(v)$ shows the set of active in-neighbors of *v* whose impact time on *v* is less than or equal to *t*.

For the live-edge process, the probability that an inactive *v* will be activated at *t + 1*, given that it has not been active yet by the end of *t*, can be computed as follows. We start with a set of initial nodes A_0. For each node *v*, if its incoming live-edge is incident to a node in A_t and impact time of that node on *v* is less than or equal to *t+1*, then *v* will be reachable at stage *t+1*. Therefore, A_{t+1} is the union of A_t and new reachable nodes. If node *v* is not reachable by time *t*, then the probability that *v* becomes reachable in *t + 1* is the probability that *v*'s effective live-edge is incident on some node in $eff_{t+1}(v) \setminus eff_t(v)$, given that *v* is unreachable in *t*, which is equal to:

$$p_{DLE}(v,t+1) = \frac{\sum_{u\in eff_{t+1}(v)\setminus eff_t(v)} w_{uv}}{1-\sum_{u\in eff_t(v)} w_{uv}} \tag{4}$$

This probability is exactly same as $p_{DLT}(v,t+1)$.

Considering this equivalence, the arguments will be similar to the proof of Theorem 4. Using the live-edge process, we obtain a possible world. For two arbitrary sets *S* and *T* such that $S \subseteq T \subseteq V$, let $\mathrm{U}_{u\in S} R_w(u)$ shows the set of reachable nodes from *S* over deterministic graph *w*. Let $x \in V \setminus T$ be an arbitrary node. If $x \in \mathrm{U}_{u\in S} R_w(u)$, then the source of the live path to *x* also exists in *T*. The value of each reachable node is computed by the freshness function based on the sum of the delays on the shortest path from *T* to *v*. The shortest path from *T* to *v* is no longer than the shortest path from *S* to *v*. It means the value of reachable nodes from *T* cannot be less than the value of reachable nodes from *S*, therefore $PV_w(.)$ is monotone. Since *PV(.)* is a non-negative linear combination of monotone functions it is monotone as well.

Also, in a deterministic graph *w*, $PV_w(S\cup\{x\}) - PV_w(S)$ is the value of nodes reachable from *x*, but not from *S*. Similarly, $PV_w(T\cup\{x\}) - PV_w(T)$ is the value of nodes reachable from *x*, but not from *T*. As said before, the value of activated nodes is computed based on the time delay of the shortest live path to each node. Since $S \subseteq T$, the value of reachable nodes from *T* is greater than or equal to the value of reachable nodes from *S;* consequently, we have $PV_w(S\cup\{x\}) - PV_w(S) \geq PV_w(T\cup\{x\}) - PV_w(T)$ **.** It means $PV_w(.)$ is submodular. Since submodularity is closed under nonnegative linear combinations [31], *PV(.)* is submodular as well, and the theorem is proved. □

3.2. Approximation algorithm for the TSIM problem

As we showed in Theorem 3 and Theorem 5, the TSIM problem is NP-hard under both the DIC and DLT models. However, a celebrated result by Nemhauser et.al. [31] states that for any monotone submodular function *f* with $f(\emptyset)=0$, the problem of finding a set *S* of size *k* that maximizes *f(S)*, can be approximated within a factor of $(1-1/e)$ by a greedy algorithm. Based on this theorem, we adapt the greedy algorithm [13] to solve the TSIM problem.

3.2.1. Time-Sensitive Greedy algorithm

We have proved the monotonicity and submodularity of *PV(.)* function under DIC and DLT models in Theorem 4 and Theorem 6, respectively. Based on these theorems and the theorem stated by Nemhauser et.al. [31], in this section we propose Time-Sensitive Greedy algorithm to approximately solve the TSIM problem with (1-1/e) approximation guarantee.

As shown in Algorithm 1, the greedy algorithm iteratively selects a node with the largest marginal gain and adds it to the seed set *S*, until *k* seeds are selected. The marginal gain is the difference of the propagated value initiated by *S* and the propagated value initiated by *S* union the selected node.

Algorithm 1. Time-Sensitive Greedy algorithm

Input: $G, k, ff(t)$
Output: S

initialize $S=\emptyset$
for $i=1$ *to* k do
    $u \leftarrow \arg\max_{v\in V\setminus S} PV(S\cup\{v\})-PV(S)$
    $S \leftarrow S\cup\{u\}$
end
return S

It is proved that computing the marginal influence is NP-hard for the IC and LT models [19,20]. Thus, Monte-Carlo simulation is used as a common approach for estimating influence spread. This leads to $(1-1/e-\varepsilon)$ approximation ratio, where ε depends on the accuracy of Monte–Carlo estimation [13]. Since IC and LT are special case of DIC and DLT models respectively, computing marginal value for these models is NP-hard as well. To estimate marginal gain for the TSIM problem, we adapt Monte-Carlo simulation to accommodate time sensitivity of information value in the computation of propagation value. The main idea of Monte-Carlo simulation is to run the diffusion model by selected nodes for many times and take the average of obtained values at each iteration.

Each simulation run is the same as what we described before in section 3.1.1 and 3.1.2 about influence propagation in the DIC and DLT models. Algorithm 2 and Algorithm 3 present the computation of propagation value in DIC and DLT models, respectively.

Algorithm 2. Computing propagation value for the DIC model

Input: *G, S, ff(t)*
Output: *PV(S)*

$PV \leftarrow 0$
$t \leftarrow 0, Act_0 \leftarrow S$
$v.status \leftarrow inactive, v.acttime \leftarrow +\infty$ *for all* $v \in V / S$
$v.status \leftarrow active, v.acttime \leftarrow 0,\ PV \leftarrow PV + ff(0)$ *for all* $v \in S$
do
    $cur \leftarrow t$
    for $u \in Act_{cur}$ do
        for $(u,v) \in E$ *and* $v.status \neq active$ do
            if *rand* $< p_{uv}$ then
                if $d_{uv} = 0$ then

```
                                                    v.status ← active
                                                    v.acttime ← cur
                                                    Act_cur ← Act_cur ∪ {v}
                                                    PV = PV + ff(cur)
                                            else if  v.status = inactive  then
                                                    v.status ← latent active
                                                    v.acttime ← t + d_uv
                                            else if  t + d_uv < v.acttime  then
                                                    v.acttime ← t + d_uv
            t ← t + 1
            Act_t ← {v | v.status = latent active ∧ v.acttime = t}
            v.status ← active, PV ← PV + ff(v.acttime)   for all v ∈ Act_t
    while (Act_t ≠ ∅  or  |{v | v.status = latent active}| ≠ 0) and ff(t) ≠ 0
    return PV
```

Algorithm 3. Computing propagation value for the DLT model

```
    Input: G, S, ff(t)
    Output: PV(S)
    PV ← 0
    t ← 0, Act_t ← S
    v.status ← inactive, v.acttime ← +∞, AN_v = ∅        for all v ∈ V / S
    v.status ← active, v.acttime ← 0, PV ← PV + ff(0)    for all v ∈ S
    do
            cur ← t
            for u ∈ Act_cur  do
                    for (u,v) ∈ E and v.status ≠ active  do
                            ti_u ← t + d_uv
                            AN_v ← AN_v + {(u, ti_u)}
                            if   d_uv = 0  and  Σ_{u|(u,ti_u)∈AN_v ∧ ti_u ≤ cur} w_uv ≥ θ_v  then
                                    v.status ← active
                                    v.acttime ← cur
                                    Act_cur ← Act_cur ∪ {v}
                                    PV = PV + ff(cur)
                            else if  v.status = inactive and Σ_{u|(u,ti_u)∈AN_v} w_uv ≥ θ_v  then
                                    v.status ← latent active
            t ← t + 1
            Act_t ← {v | (v.status = latent active) ∧ (Σ_{u|(u,ti_u)∈AN_v ∧ ti_u ≤ t} w_uv ≥ θ_v)}
            v.status ← active, v.acttime = t, PV ← PV + ff(v.acttime)   for all v ∈ Act_t
    while (Act_t ≠ ∅  or  |{v | v.status = latent active}| ≠ 0) and ff(t) ≠ 0
    return PV
```

3.3. Heuristic methods for the TSIM problem

Although greedy algorithm guarantees approximation ratio for the TSIM problem, the running time is large. A possible approach is to use heuristic methods. Since in this paper the TSIM problem is discussed under two different diffusion models, namely DIC and DLT, we apply model-independent heuristics for comparison. Consequently, we propose two time-sensitive heuristic methods, which are described in this section.

3.3.1. Time-Sensitive Degree heuristic

Degree heuristic is frequently used for selecting seeds in the influence maximization problem [22,23]. It selects the k nodes with the highest degrees in the social network as initial nodes. Experimental results have shown that degree

centrality has good performance in the influence maximization problem [32]. However, it does not consider time aspect in the diffusion of information.

Since the value of propagating items is dependent on the activation time, a node that can activate more nodes in a shorter time is more desirable. In other words, a node that activates its neighbors with high freshness value can be a good spreader. Therefore, we define the Time-Sensitive Degree of node *u* as:

$$TSDeg(u) = \sum_{v \in N^{out}(u)} ff(d_{uv}) \tag{5}$$

where *v* is out-neighbor of *u* and d_{uv} is the propagation delay from *u* to *v*. We sort the nodes in descending order based on their Time-Sensitive Degree and select the first *k* nodes as seed set.

3.3.2. Time-Sensitive Betweenness heuristic

Betweenness centrality is an indicator for the importance of a node in the network. Betweenness of a node is defined as the number of shortest paths from all vertices to all others that pass through that node. It has shown good performance in identifying influential nodes in social networks [22]. In this paper, we extend the betweenness measure to capture the dependency of propagated value to time delay. Considering the time aspect, a path with less delay can be more valuable.

As we discussed before the value of a propagated item is defined by a freshness function. Thus, we weight each edge *(u,v)* by $1/(ff(d_{uv})+1)$ and define Time-Sensitive Betweenness of node *u* as:

$$TSBet(u) = \sum_{x,y \in V} \frac{nsp_u(x,y)}{nsp(x,y)} \tag{6}$$

where *nsp(x,y)* is the number of shortest paths from node *x* to *y*, and $nsp_u(x,y)$ is the number of shortest paths from node *x* to *y* that pass through node *u*. The shortest paths between nodes are computed on the weighted graph we defined above.

4. Results and Discussion

We evaluated the efficiency and the effectiveness of our approximation algorithms and the proposed heuristics on four real-world social networks for both the DIC and DLT models. In addition, we compared our time-sensitive methods with conventional influence maximization methods. The experimental setups and the corresponding experimental results are presented in section 4.1 and 4.2, respectively.

4.1. Experimental setups

In this section, we describe the datasets we used and the configurations of the conducted experiments.

4.1.1. Datasets

We used four real networks for our empirical study, namely Twitter, WikiVote, HEP-PH, and Epinions. Twitter is one of the most popular social networks for online communications. We used the dataset extracted by Hashmi et al. in [33] which contains following relationships of 2492 users. The second dataset, WikiVote, contains all the Wikipedia voting data from the inception of Wikipedia until January 2008. Nodes in the network represent Wikipedia users and an edge from node *u* to node *v* represents user *u* voted for user *v*, which means that *v* has influence on *u*. This dataset can be obtained from [34]. The third dataset, HEP-PH, is a citation graph that covers all the Arxiv Highenergy Physics papers from January 1993 to April 2003. In this graph, the nodes represent papers and the edges represent citation. If paper *u* cites paper *v*, it means *v* has influence on *u*. This dataset is downloadable from [34]. The last dataset, Epinions, is extracted from Epinions.com, which is a general consumer review site. It shows who-trust-whom relationships between members. Each node represents a user of the site and an edge from *u* to *v* means that *u* trusts *v*, i.e., *v* has influence on *u*. This dataset is available at [34]. The statistics of the above datasets are presented in Table 1.

Table 1. Statistics of real-world social networks.

Dataset	Number of Nodes	Number of Edges
Twitter	2492	17658
WikiVote	7115	103689
HEP-PH	34546	421578
Epinions	75879	508837

4.1.2. Configurations

We evaluated our Time-Sensitive Greedy Algorithm (TSGreedy) and our proposed heuristic algorithms, which are, Time-Sensitive Degree heuristic (TSDeg) and Time-Sensitive Betweenness heuristic (TSBet), in terms of propagation value and running time. In addition, we compared them with conventional influence maximization algorithms including Greedy Algorithm (Greedy), Degree heuristic (Deg), Betweenness heuristic (Bet), and Random (Rnd) method that acts as the baseline method and select *k* seeds randomly.

For Greedy and TSGreedy algorithm, we exploited CELF++ optimization [16]. For estimating influence spread under both the DIC and DLT models, we performed Monte Carlo simulations 10000 times as in [13,16] and took the average of simulations as result. We compared the amount of propagated value of aforementioned algorithms with different sizes of seed set, ranging from 1 to 50.

The influence probability of edges in the DIC model is set using the Weighted Cascade model [13,20]. In the Weighted Cascade model the probability of edge *(u,v)* is assigned to $1/|N^{in}(v)|$, where $|N^{in}(v)|$ is the number of in-neighbors of *v*. Similarly, in the DLT model, we set the influence weight of each incoming edges of *v* to be $1/|N^{in}(v)|$ as in [13,19]. To determine the influence delay of edges, we used discrete uniform distribution, where we set the maximum value of d_{uv} to 20 for simplicity, i.e., $d_{uv}: U(0,20)$. It should be noted that the determination of influence probability and influence delay are orthogonal to our defined problem.

As the freshness function, we employed exponential decay function and constant freshness function. An exponential decay function is of the form $ff(t) = ff(0)e^{-\lambda t}$, where *ff(0)* is the initial value and $\lambda > 0$ is the decay rate. We set the initial value of each propagating item equal to 1 and considered two different decay rates, $\lambda = 0.2$ and $\lambda = 2$ for comparison purposes. The larger the λ, the faster the freshness function decreases. A constant freshness function means that no decay in the value of the item occurs over time. Therefore, when *ff(t)*=1 the TSIM problem is equivalent to the conventional influence maximization problem.

4.2. Experimental results

In this section, we present the experimental results of the proposed methods on four real-world datasets. We evaluated different methods with respect to influence spread (propagation value) and running time. In addition, we compared the effectiveness of our methods with conventional influence maximization methods. We examined the effect of the freshness function on the TSIM problem in section 4.2.3.

4.2.1. Influence spread

We varied the seed set size, *k*, from 1 to 50 and for each *k* we estimated the expected value of activated nodes in different algorithms. The expected value of activated nodes measures the effectiveness of the algorithms for the TSIM problem. The larger the expected value of activated nodes returned by an algorithm, the better the algorithm is.

Figure 1 and Figure 2 illustrate the propagation value of different algorithms achieved on four datasets with freshness function $ff(t) = e^{-0.2t}$ under the DIC and DLT models, respectively. We can see from the results that TSGreedy outperforms other methods in all cases and the second best algorithm is TSDeg. TSBet performs worse than TSGreedy and TSDeg, but it outperforms conventional methods. Rnd has the worst performance.

Greedy, Deg and Bet methods consistently achieve lower propagation values than our proposed time-sensitive methods. This shows that conventional influence maximization methods are not effective in the TSIM problem.

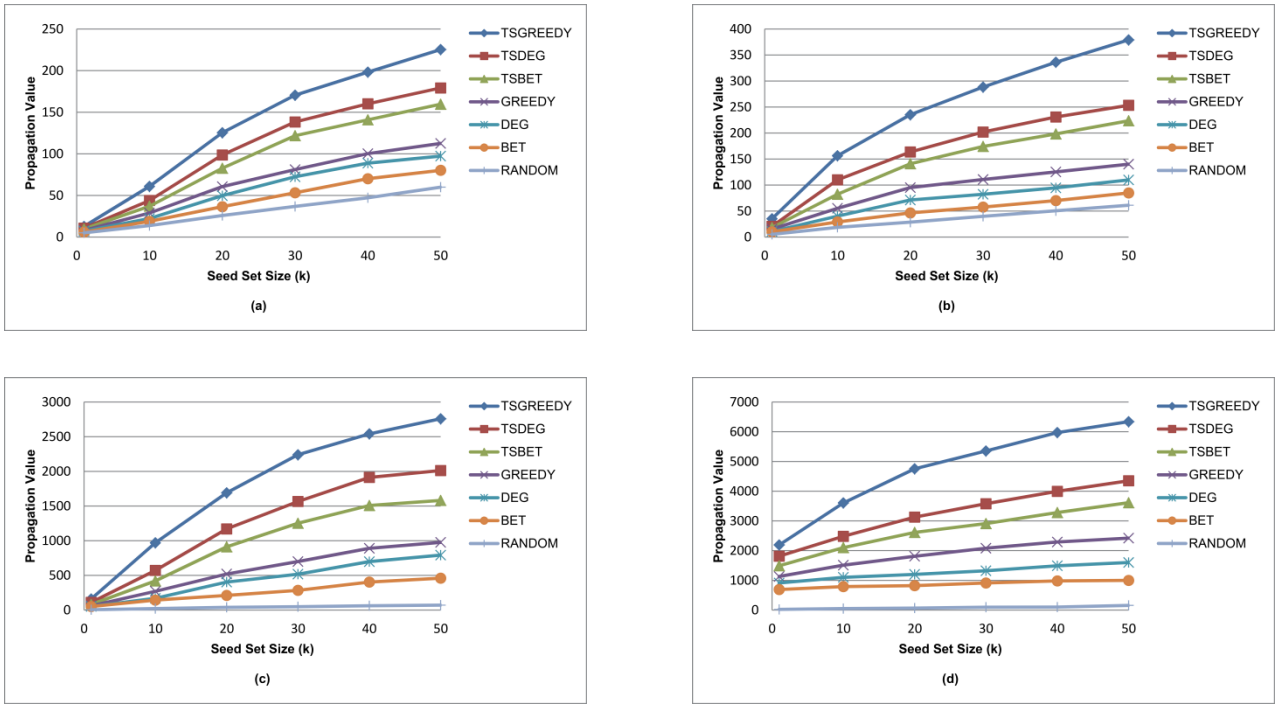

Figure 1. Propagation value of different algorithms under DIC model on four dataset (a) Twitter; (b) WikiVote; (c) HEP-PH; (d) Epinions, with freshness function $ff(t) = e^{-0.2t}$.

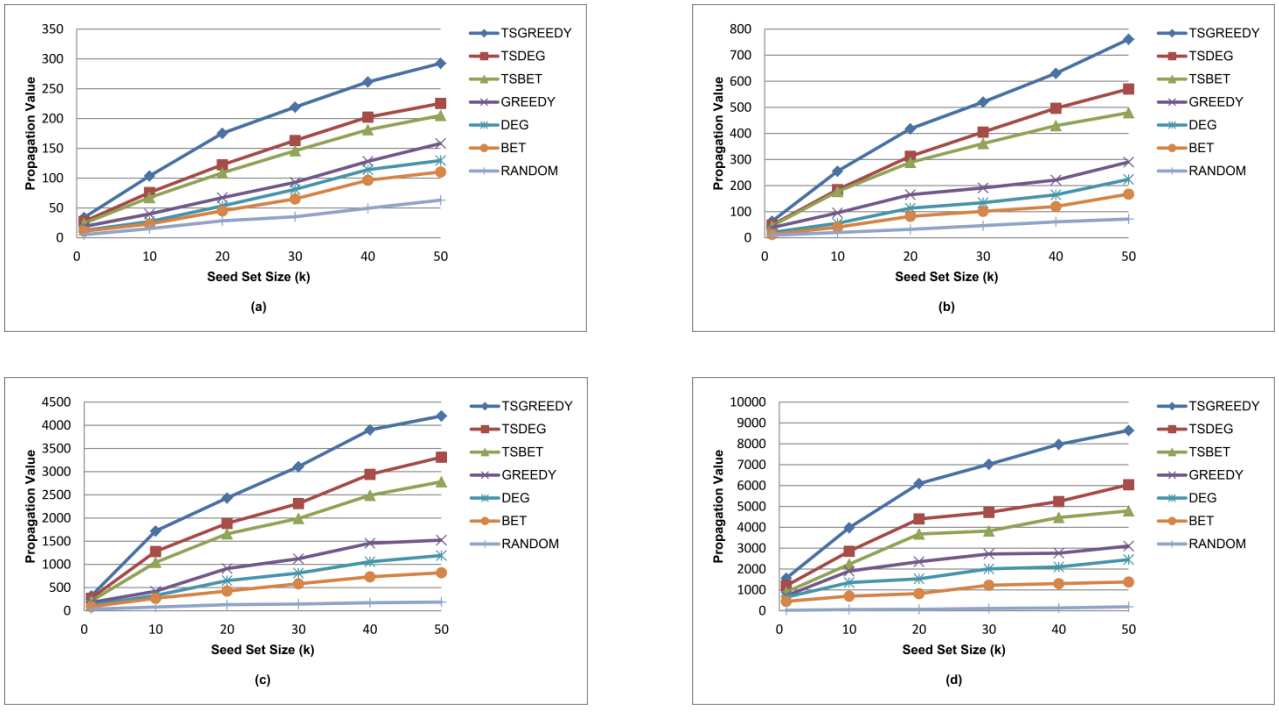

Figure 2. Propagation value of different algorithms under DLT model on four dataset (a) Twitter; (b) WikiVote; (c) HEP-PH; (d) Epinions, with freshness function $ff(t) = e^{-0.2t}$.

4.2.2. Running time

We compared the running time of our proposed algorithms, under both the DIC and DLT models. Table 2 shows the running time of different algorithms for selecting 50 seeds in four datasets when the freshness function is $ff(t) = e^{-0.2t}$. Column 1 represents the running time of TSGreedy under the DIC model, indicated by TSGreedy(DIC). Columns 2 to 4 indicate the running time of TSDeg, TSBet, and Rnd methods, respectively. Since TSDeg, TSBet and Rnd methods are model-independent, their running time is the same under both the DIC and DLT models. The running time of TSGreedy under the DLT model is represented in column 5 as TSGreedy(DLT).

Table 2. Running times of proposed methods on four real social networks in second (k=50, $ff(t) = e^{-0.2t}$).

Algorithms	TSGreedy (DIC)	TSDeg	TSBet	Rnd	TSGreedy (DLT)
Twitter	1.92*10^{3}	0.96*10^{-3}	1.12*10^{3}	2.43*10^{-4}	2.18*10^{3}
WikiVote	1.51*10^{4}	1.81*10^{-3}	1.03*10^{4}	4.38*10^{-4}	1.72*10^{4}
HEP-PH	2.01*10^{5}	3.52*10^{-3}	1.27*10^{5}	6.01*10^{-4}	2.23*10^{5}
Epinions	2.69*10^{5}	4.9*10^{-3}	2.28*10^{5}	7.92*10^{-4}	2.91*10^{5}

As the results show, TSGreedy is the slowest algorithm amongst the proposed algorithms. This is reasonable since the greedy algorithm has to calculate the marginal gain of non-seed nodes at each iteration. However, it can achieve the highest propagation values among the proposed algorithms. The running time of the TSBet heuristic is in the range of the greedy algorithm while its propagation value is much worse than the greedy algorithm. The TSDeg and Rnd run extremely fast. As the results in section 4.2.1 show, TSDeg is the second best algorithm in maximizing expected value amongst the proposed methods. Therefore, it can be a good choice for large networks and when short running time is a major concern. On the other hand, TSGreedy outperforms other proposed methods in terms of propagation value, but it takes more time and is not scalable for large networks.

4.2.3. Effect of freshness function

We examined the impact of the freshness function on the TSIM problem. We evaluated the influence spread of TSGreedy under the DIC model on our datasets with three different freshness functions that are $ff(t) = 1$, $ff(t) = e^{-0.2t}$, and $ff(t) = e^{-2t}$.

It is obvious that when the freshness function decreases faster, the total propagation value will reduce further. For example, the propagation value of TSGreedy(DIC) on Twitter dataset with k=50 is equal to 448, 225.3, and 74.6, for $ff(t) = 1$, $ff(t) = e^{-0.2t}$, and $ff(t) = e^{-2t}$, respectively. As the results show, when the freshness function is constant we have highest propagation value. This is because when $ff(t) = 1$ there is no decay in the value of information and all activated nodes have a value of 1, but in $ff(t) = e^{-0.2t}$ and $ff(t) = e^{-2t}$ the value of activated nodes are dependent on their activation time and it decreases over time. The rate of decay in $ff(t) = e^{-2t}$ is greater than $ff(t) = e^{-0.2t}$.

In addition, we investigated the overlaps of seed sets returned by TSGreedy(DIC) for different freshness functions. Table 3 represents the results for k=50 in four datasets. Each entry in the table shows the number of common seeds returned by TSGreedy(DIC) for the two freshness functions in corresponding columns and rows.

Table 3. The overlaps of seed sets returned by TSGreedy(DIC) with different freshness function when k=50.

	freshness function	$ff(t)=1$	$ff(t)=e^{-0.2t}$	$ff(t)=e^{-2t}$
Twitter	$ff(t)=1$	50	20	11
	$ff(t)=e^{-0.2t}$		50	21
	$ff(t)=e^{-2t}$			50
WikiVote	$ff(t)=1$	50	18	9
	$ff(t)=e^{-0.2t}$		50	20
	$ff(t)=e^{-2t}$			50
HEP-PH	$ff(t)=1$	50	15	6
	$ff(t)=e^{-0.2t}$		50	12
	$ff(t)=e^{-2t}$			50
Epinions	$ff(t)=1$	50	13	4
	$ff(t)=e^{-0.2t}$		50	11
	$ff(t)=e^{-2t}$			50

The results show that the seed sets, which maximizes the propagation value, differs significantly for different freshness functions, i.e., the set of nodes maximizing the propagation value for a given freshness function do not necessarily maximize the value for a different decay pattern. This indicates that considering the time sensitivity of propagation value plays an important role in the influence maximization problem.

5. Conclusions

In this paper, we introduced a novel problem, called Time-Sensitive Influence Maximization (TSIM) problem that considers the effect of time on the value of propagated items. Time-Sensitive Influence Maximization is important in applications where the value of propagated items decreases over time. For example, in dissemination of hot topics or urgent information, it is important to influence more people as fast as possible. Also, in viral marketing, the sooner a product is sold, the more valuable it is, because the sooner the company receives a return on investment.

To address TSIM problem, we extended the standard Independent Cascade and Linear Threshold models to the Delayed Independent Cascade (DIC) and Delayed Linear Threshold (DLT) models, respectively. We proved that the spread function of TSIM is monotone and submodular under the DIC and DLT models, and proposed a greedy algorithm with an approximation guarantee. We also modified degree and betweenness centrality to adapt to the time-sensitive nature of the propagation process. Experimental results on real datasets verified the effectiveness of our formulation and the proposed algorithms.

One drawback of the greedy algorithm is that it is computationally expensive and is not scalable for large social networks. On the other hand, the heuristic methods do not guarantee the quality of results. Therefore, one potential direction for future work is to consider the scalability issue for the TSIM problem and design effective and efficient algorithms that are scalable on large networks.

Also in this paper, we assumed that the influence probability and the influence delay of edges are given as input. However, in real social networks the edges are not labeled with influence probabilities or influence delays. Thus, learning these parameters from propagation data can be considered as another direction for future work.

Acknowledgements

The authors like to thank Professor David Parsons for proof reading the manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

[1] Aggarwal CC. Social Network Data Analytics. First ed. Boston, MA: Springer Publishing Company, 2011.
[2] Chen W, Lakshmanan LVS, Castillo C. Information and Influence Propagation in Social Networks. First ed. Cleveland: Morgan & Claypool, 2013.

[3] Domingos P, Richardson M. Mining the Network Value of Customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01), San Francisco, USA, 2001, pp. 57– 66.
[4] Richardson M, Domingos P. Mining Knowledge-sharing Sites for Viral Marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, 2002, pp. 61– 70.
[5] Momtaz NJ, Aghaie A, Alizadeh S. Social Networks for Marketing : Benefits and Challenges. In: 5th Symposium on Advances in Science & Technology, Mashhad, Iran, 2011, pp. 1– 9.
[6] Long C, Wong RC-W. Viral marketing for dedicated customers. Information Systems 2014; 46: 1–23.
[7] Romero DM, Meeder B, Kleinberg J. Differences in the Mechanics of Information Diffusion Across Topics: Idioms , Political Hashtags , and Complex Contagion on Twitter. In: Proceedings of the 20th international conference on World wide web, Hyderabad, India, 2011, pp. 695– 704.
[8] Cha M TJ, Haddadi H. The Spread of Media Content through Blogs. Social Network Analysis and Mining 2011; 2: 1– 16.
[9] Song X, Tseng BL, Lin CY, Sun MT. Personalized Recommendation Driven by Information Flow. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 2006, pp. 509– 516.
[10] Song X, Chi Y, Hino K, Tseng BL. Information Flow Modeling based on Diffusion Rate for Prediction and Ranking. In: Proceedings of the 16th international conference on World Wide Web (WWW '07), Banff, AB, Canada, 2007, pp. 191– 200.
[11] Ziegler C-N, Lausen G. Propagation Models for Trust and Distrust in Social Networks. Information Systems Frontiers 2005; 7: 337– 358.
[12] Golbeck J, Hendler J. Inferring Binary Trust Relationships in Web based Social Networks. ACM Transactions on Internet Technology 2006; 6: 497– 529.
[13] Kempe D, Kleinberg J, Tardos E. Maximizing the Spread of Influence through a Social Network. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '03, Washington DC, USA, 2003, pp. 137–146.
[14] Kempe D, Kleinberg J, Tardos É. Influential Nodes in a Diffusion Model for Social Networks. Automata, Languages and Programming 2005; 3580: 1127–1138.
[15] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-effective Outbreak Detection in Networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07, San Jose, USA, 2007, pp. 420– 429.
[16] Goyal A, Lu W. CELF ++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks. In: Proceedings of the 20th International Conference Companion on World Wide Web, Hyderabad, India: ACM, 2011, pp. 47– 48.
[17] Cheng S, Shen H, Huang J, Zhang G, Cheng X. StaticGreedy : Solving the Scalability-Accuracy Dilemma in Influence Maximization. In: Proceedings of the 22nd ACM International Conference on Information and knowledge Management, San Francisco, USA, 2013, pp. 509– 518.
[18] Chen W, Wang Y. Efficient Influence Maximization in Social Networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 199– 208.
[19] Chen W, Yuan Y, Zhang L. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In: Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 2010, pp. 88–97.
[20] Wang C, Chen W, Wang Y. Scalable Influence Maximization for Independent Cascade Model in Large-scale Social Networks. Data Mining and Knowledge Discovery 2012; 25: 545–576.
[21] Jung K, Heo W, Chen W. IRIE: A Scalable Influence Maximization Algorithm for Independent Cascade Model and Its Extensions. In: Proceedings of the 12th IEEE International Conference on Data Mining (ICDM'2012), Brussels, Belgium, 2012, pp. 1–20.
[22] Mochalova A, Nanopoulos A. On the Role of Centrality in Information Diffusion in Social Networks. In: Proceedings of the 21st European Conference on Information Systems, Utrecht, Netherlands, 2013, pp. 1–12.
[23] LC F. Centrality in social networks: Conceptual Clarification. Social Networks 1979; 1: 215–239.
[24] Saito K, Kimura M, Ohara K, Motoda H. Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis. In: Proceedings of the 1st Asian Conference on Machine Learning: Advances in Machine Learning, Nanjing, China, 2009, pp. 322 – 337.
[25] Goyal A, Bonchi F, Lakshmanan LVS. Learning Influence Probabilities In Social Networks. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, New York, NY, USA, 2010, pp. 241– 250.
[26] Gomez-rodriguez M, Balduzzi D, Schölkopf B. Uncovering the Temporal Dynamics of Diffusion Networks. In: Twenty-eighth International Conference on Machine Learning, Bellevue, Washington, USA, 2011, pp. 561– 568.
[27] Chen W, Lu W, Zhang N. Time-Critical Influence Maximization in Social Networks with Time-Delayed Diffusion Process. In: Proceedings of the 26th Conference on Artificial Intelligence (AAAI'12), Toronto, Canada, 2012, pp. 592– 598.
[28] Liu B, Dong Xu GC, Zeng Y. Time Constrained Influence Maximization in Social Networks. In: IEEE 12th International Conference on Data Mining (ICDM), , 2012, pp. 439– 448.
[29] Liu B, Cong G, Zeng Y, Xu D, Chee YM. Influence Spreading Path and its Application to the Time Constrained Social Influence Maximization Problem and Beyond. IEEE Transactions on Knowledge & Data Engineering 2014; 26: 1904–1917.
[30] Goyal A, Bonchi F, Lakshmanan LVS. A Data-Based Approach to Social Influence Maximization. PVLDB 2011; 5: 73–84.
[31] Nemhauser G, Wolsey L, Fisher M. An Analysis of the Approximations for Maximizing Submodular Set Functions. Mathematical Programming 1978; 14: 265– 294.

[32] Hussain O, Anwar Z, Saleem S, Zaidi F. Empirical Analysis of Seed Selection Criterion in Influence Mining for Different Classes of Networks. In: Proceedings of the 2013 International Conference on Cloud and Green Computing, Karlsruhe, Germany, 2013, pp. 1– 8.

[33] Hashmi A, Zaidi F, Sallaberry A, Mehmood T. Are all Social Networks Structurally Similar? In: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), Istanbul, Turkey, 2012, pp. 310–314.

[34] Leskovec J, Krevl A. 'SNAP Datasets: Stanford Large Network Dataset Collection', http://snap.stanford.edu/data (2014).