Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Time-Critical Influence Maximization in Social Networks with Time-Delayed Diffusion Process

Wei Chen
Microsoft Research Asia
Beijing, China
weic@microsoft.com

Wei Lu
University of British Columbia
Vancouver, B.C., Canada
welu@cs.ubc.ca

Ning Zhang
Univ. of Science and Technology of China
Hefei, Anhui, China
lemonustc@gmail.com

Abstract

Influence maximization is a problem of finding a small set of highly influential users in a social network such that the spread of influence under certain propagation models is maximized. In this paper, we consider time-critical influence maximization, in which one wants to maximize influence spread within a given deadline. Since timing is considered in the optimization, we also extend the Independent Cascade (IC) model to incorporate the time delay aspect of influence diffusion in social networks. We show that time-critical influence maximization under the time-delayed IC model maintains desired properties such as submodularity, which allows a greedy algorithm to achieve an approximation ratio of $1 - 1/e$, to circumvent the **NP**-hardness of the problem. To overcome the inefficiency of the approximation algorithm, we design two heuristic algorithms: the first one is based on a dynamic programming procedure that computes exact influence in tree structures, while the second one converts the problem to one in the original IC model and then applies existing fast heuristics to it. Our simulation results demonstrate that our heuristics achieve the same level of influence spread as the greedy algorithm while running a few orders of magnitude faster, and they also outperform existing algorithms that disregard the deadline constraint and delays in diffusion.

1 Introduction

Recently, the rapidly increasing popularity of online social networking sites such as Facebook, Twitter, and Google+ opens up great opportunities for large-scale viral marketing campaigns. Viral marketing, first introduced to the data mining community by Domingos and Richardson (2001), is a cost-effective marketing strategy that promotes products by giving free or discounted items to a selected group of highly influential individuals (seeds), in the hope that through the word-of-mouth effects, a large number of product adoption will occur. Motivated by viral marketing, *influence maximization* emerges as a fundamental data mining problem concerning the propagation of ideas, opinions, and innovations through social networks.

In their seminal paper, Kempe et al. (2003) formulated influence maximization as a problem in discrete optimization: Given a network graph G with pairwise user influence probabilities on edges, and a positive number k, find k users, such that by activating them initially, the expected spread of influence is maximized under certain propagation models. Two classical propagation models studied in the the literature are the *Independent Cascade* (IC) and the *Linear Threshold* (LT) model.

The family of models considered in Kempe et al. and its follow-ups, including IC and LT, do not fully incorporate important *temporal* aspects that have been well observed in the dynamics of influence diffusion. First, the propagation of influence from one person to another may incur a certain amount of *time delay*, which is evident from recent studies in statistical physics (Iribarren and Moro 2009; Karsai et al. 2011).

Second, the spread of influence may be *time-critical* in practice. In a certain viral marketing campaign, it might be the case that the company wishes to trigger a large volume of product adoption in a fairly short time frame, e.g., a three-day sale. As a motivating example, let us suppose that Alice has bought an Xbox 360 console and Kinect with a good discount, but the deal would only last for three days. Alice wanted to recommend this deal to Bob, but whether her recommendation would be effective depends on whether Alice and Bob can be in touch (e.g., meeting in person, or Bob seeing the message left by Alice on Facebook) before the discount expires. Therefore, when we try to maximize the spread of influence for a viral marketing campaign facing this kind of scenarios, we need to take both the time delay aspect of influence diffusion and the time-critical constraint of the campaign into consideration.

To this end, we extend the influence maximization problem to have a *deadline constraint* to reflect the time-critical effect. We also propose a new propagation model, the *Independent Cascade model with Meeting events (IC-M)* to capture the delay of propagation in time. We show that the IC-M model maintains monotonicity and submodularity, which implies a greedy $(1 - 1/e)$-approximation algorithm to circumvent the **NP**-hardness of the problem. In addition, we design two efficient and effective heuristic algorithms, MIA-M and MIA-C, based on the notion of Maximum Influence Arborescence (MIA) (Chen, Wang, and Wang 2010) to tackle time-critical influence maximization under the IC-M model. Our experiments demonstrate that both algorithms produce seed sets with equally good quality as those mined by approximation algorithm, while being two to three or-

ders of magnitude faster. Moreover, we show that only using standard heuristics such as MIA and disregarding time delays and deadline constraint could result in poor influence spread compared to our heuristics that are specifically designed for this context.

1.1 Related Work

Domingos and Richardson (2001; 2002) first posed influence maximization as an algorithmic problem. They modeled the problem using Markov random fields and proposed heuristic solutions. Kempe et al. (2003) studied influence maximization as a discrete optimization problem. They showed that the problem is **NP**-hard under both the IC and LT models, and relied on submodularity to obtain a $(1-1/e)$ greedy approximation algorithm.

A number of studies following (Kempe, Kleinberg, and Tardos 2003) developed more efficient and scalable solutions, including the cost-effective lazy forward (CELF) optimization (Leskovec et al. 2007) and also work by Kimura et al. (2007), Chen et al. (2009), etc. Specifically for the IC model, Chen et al. (2010) showed that it is **#P**-hard to compute the exact influence of any node set in general graphs. They proposed the MIA model which uses influence in local tree structures to approximate influence propagated through the entire network. They then developed scalable algorithms to compute exact influence in trees and mine seed sets with equally good quality as those found by the approximation algorithm. Goyal et al. (2012) leveraged real propagation traces to derive more accurate diffusion models.

The time-delay phenomena in information diffusion has been explored in statistical physics. Iribarren and Moro (2009) observed from a large-scale Internet viral marketing experiment that the dynamics of information diffusion are controlled by the heterogeneity of human activities. More recently, using time-stamped phone call records, Karsai et al. (2011) found that the spreading speed of information on social networks is much slower than one may expect, due to various kinds of correlations, such as community structures in the graph, weight-topology correlations, and burstiness.

2 Influence Maximization with Deadline and Meeting Events and Its Properties

2.1 Model and Problem Definition

We first describe the standard Independent Cascade (IC) model in Kempe et al. (2003), and then show our extensions that incorporate deadline and random meeting events. In the IC model, a social network is modeled as a directed graph $G = (V, E)$, where V is the set of nodes representing users and E is the set of directed edges representing links (relationship) between users. Each edge $(u, v) \in E$ is associated with an influence probability $p(u, v)$ defined by function $p : E \to [0, 1]$. If $(u, v) \notin E$, define $p(u, v) = 0$.

The diffusion process under the IC model proceeds in discrete time steps $0, 1, 2, \ldots$. Initially, a *seed set* $S \subseteq V$ is targeted and activated at step 0, while all other nodes are inactive. At any step $t \geq 1$, any node u activated at step $t - 1$ is given a single chance to activate any of its currently inactivate neighbors v with independent success probability $p(u, v)$. Once a node is activated, it stays active. The process continues until no new nodes can be activated. The influence maximization problem under the IC model is to find a seed set S with at most k nodes such that the expected number of activated nodes after the diffusion terminates, called *influence spread* and denoted by $\sigma(S)$, is maximized.

We now describe our extension to the IC model to incorporate time-delayed influence diffusion, which we denote by IC-M (for Independent Cascade with Meeting events). In the IC-M model, each edge $(u, v) \in E$ is also associated with a *meeting probability* $m(u, v)$ defined by function $m : E \to [0, 1]$ (if $(u, v) \notin E$, $m(u, v) = 0$). As in IC, a seed set S is targeted and activated at step 0. At any step $t \geq 1$, an active node u meets any of its currently inactive neighbors v independently with probability $m(u, v)$. If a meeting event occurs between u and v for the *first* time, u is given a *single* chance to try activating v, with an independent success probability $p(u, v)$. If the attempt succeeds, v becomes active at step t and will start propagating influence at $t+1$. The diffusion process quiesces when all active nodes have met with all their neighbors and no new nodes can be activated.

Several possibilities can be considered in mapping the meeting events in the IC-M model to real actions in online social networks. For instance, a user u on Facebook posting a message on her friend v's wall can be considered as a meeting event. Different pairs of friends may have different frequencies of exchanging messages on each other's walls, which is reflected by the meeting probability.

Note that the original IC model is a special case of IC-M with $m(u, v) = 1$ for all edges $(u, v) \in E$. More importantly, for the original influence maximization problem, the meeting probability is not essential, because as long as $m(u, v) > 0$, eventually u will meet with v and try to influence v once. Thus, if we only consider the overall influence in the entire run, there would be no need to introduce meeting probabilities. However, if we consider influence within a *deadline constraint*, then meeting probability is an important factor in determining the optimal seed set.

Formally, for a deadline $\tau \in \mathbb{Z}_+$, we define $\sigma_\tau : 2^V \to \mathbb{R}_+$ to be the set function such that $\sigma_\tau(S)$ with $S \subseteq V$ is the expected number of activated nodes by the end of time step τ under the IC-M model, with S as the seed set. The *time-critical influence maximization with a deadline constraint* τ is the problem of finding the seed set S with at most k seeds such that the expected number of activated nodes by step τ is maximized, i.e., finding $S^* = \arg\max_{S \subseteq V, |S| \leq k} \sigma_\tau(S)$.

Since the original influence maximization problem is **NP**-hard for the IC model (Kempe, Kleinberg, and Tardos 2003) and that problem is a special case of time-critical influence maximization for the IC-M model with all $m(u, v) = 1$ and deadline constraint $\tau = |V|$. This leads to the following hardness result.

Theorem 1. *The time-critical influence maximization problem is **NP**-hard for the IC-M model.*

2.2 Properties of the IC-M Model

Although to find the optimal solution for time-critical influence maximization with deadline τ under IC-M is **NP**-hard (Theorem 1), we show that the influence function $\sigma_\tau(\cdot)$ is monotone and submodular, which allows a hill-climbing-style greedy algorithm to achieve a $(1-1/e)$-approximation to the optimal.

Given a ground set U, a set function $f\colon 2^U \rightarrow \mathbb{R}$ is *monotone* if $f(S_1) \leq f(S_2)$ whenever $S_1 \subseteq S_2$. Also, the function is *submodular* if $f(S_1 \cup \{w\}) - f(S_1) \geq f(S_2 \cup \{w\}) - f(S_2), \forall S_1 \subseteq S_2, \forall w \in U \setminus S_2$. Submodularity captures the law of diminishing marginal returns, a well-known principle in economics.

Theorem 2. *The influence function $\sigma_\tau(\cdot)$ is monotone and submodular for an arbitrary instance of the IC-M model, given any deadline constraint $\tau \geq 1$.*

To prove the theorem, we can view the random cascade process under IC-M using the "possible world" semantics and the principle of deferred decisions. That is, we can suppose that before the cascade starts, a set of outcomes for all meeting events, as well as the "live-or-blocked" identity for all edges are already determined but not yet revealed.

More specifically, for each meeting event (a (u, v) pair and a time step $t \in [1, \tau]$), we flip a coin with bias $m(u, v)$ to determine if u will meet v at t. Similarly, for each edge $(u, v) \in E$, we flip once with bias $p(u, v)$, and we declare the edge "live" with probability $p(u, v)$, or "blocked" with probability $1 - p(u, v)$. All coin-flips are independent. The identity of the edge (u, v) is revealed *in the event* that u is active and is meeting the inactive v for the first time. Therefore, a certain set of outcomes of all coin flips corresponds to one *possible world*, denoted by X, which is a deterministic graph (with all blocked edges removed) obtained by conditioning on that particular set of outcomes.

Proof of Theorem 2. Fix a set X_M of outcomes of all meeting events ($\forall (u, v) \in E, \forall t \in [0, \tau]$), and also a set X_E of live-or-blocked identities for all edges. Since the coin-flips for meeting events and those for live-edge selections are orthogonal, and all flips are independent, any X_E on top of an X_M leads to a possible world X.

Next, we define the notion of "reachability" in X. Consider a live edge (u, v) in X. Traditionally, without meeting events, v is reachable from u via just one hop. Now with pre-determined meeting sequences, v is reachable from u via $t_v - t_u$ hops, where t_u is the step in which u itself is reached, and t_v is the first step when u meets v, after t_u. Hence, we say that v is *reachable* from a seed set S if and only if (1) there exists at least one path consisting entirely of live edges (called live-path) from some node in S to v, and (2) the *collective number of hops* along the *shortest* live-path from S to v is no greater than τ.

Then, let $\sigma_\tau^X(S)$ be the number of nodes reachable from S in X (by the reachability definition above). Let S_1 and S_2 be two arbitrary sets such that $S_1 \subseteq S_2 \subseteq V$, and let node $w \in V \setminus S_2$ be arbitrary. The monotonicity of $\sigma_\tau^X(\cdot)$ holds, since if some node u can be reached by S_1, the source of the live-path to u must be also in S_2. As for submodularity, consider a certain node u which is reachable from $S_2 \cup \{w\}$ but not from S_2. This implies (1) u is not reachable from S_1 either, and (2) the source of the live-path to u must be w. Hence, u is reachable from $S_1 \cup \{w\}$ but not from S_1. This gives $\sigma_\tau^X(S_1 \cup \{w\}) - \sigma_\tau^X(S_1) \geq \sigma_\tau^X(S_2 \cup \{w\}) - \sigma_\tau^X(S_2)$.

Let $\mathcal{E}_I$ denote the event that I is the true realization (virtually) of the corresponding random process. Taking the expectation over all possible worlds, we have $\sigma_\tau(S) = \sum_X \Pr[\mathcal{E}_X] \cdot \sigma_\tau^X(S), \forall S \subseteq V$, where X is any combination of X_E and X_M, and $\Pr[\mathcal{E}_X] = \Pr[\mathcal{E}_{X_E}] \cdot \Pr[\mathcal{E}_{X_M}]$. Therefore, $\sigma(\cdot)$ is a nonnegative linear combination of monotone and submodular functions, which is also monotone and submodular. This was to be shown. □

Algorithm 1: Greedy $(G = (V, E), k, \sigma_\tau)$

```
S ← ∅;
for i = 1 → k do
    u ← argmax_{v∈V\S} [σ_τ(S ∪ {v}) − σ_τ(S)];
    S ← S ∪ {u};
Output S;
```

With Theorem 2, we can apply the result in Nemhauser et al. (1978) to obtain a greedy algorithm (Algorithm 1: Greedy) that approximates the optimal solution with a factor of $1-1/e$ for the time-critical influence maximization problem under the IC-M model. The greedy algorithm repeatedly grows S by adding u with the largest marginal influence w.r.t S in each iteration until $|S| = k$ or no more node can provide marginal gain greater then zero.

However, Chen et al. (2010) shows that it is **#P**-hard to compute $\sigma(\cdot)$ values in general graphs for the IC model, and this hardness result is applicable to our problem as it subsumes the original one (Sec. 2.1). A common practice is to estimate influence spread using Monte-Carlo (MC) simulations, in which case the approximation ratio of Greedy drops to $1-1/e-\epsilon$, where ϵ is small if the number of simulations is sufficiently large. Due to expensive simulations, the greedy algorithm is not scalable to large data, even the implementation is accelerated by the CELF optimization (Leskovec et al. 2007).

3 Computing Influence in Arborescences

In this section, we present an dynamic programming algorithm that computes exact influence spread in tree structures, which will be used in Sec. 4 to develop MIA-M.

An *in-arborescence* is a directed tree where all edges point into the root. Given a graph $G = (V, E)$ with influence probability function p and meeting probability function m, consider an in-arborescence $A = (V_A, E_A)$ rooted at v where $V_A \subseteq V$ and $E_A \subseteq E$. We assume that influence propagates to v only from nodes in A. We also assume that there exists at least one $s \in S$ such that $s \in V_A$; otherwise no nodes can be activated in A. Given a seed set S and deadline τ, we show how to compute $\sigma_\tau(S)$ in A in time linear to the size of the graph.

Let $ap(u,t)$ be the *activation probability* of u at step t, i.e., the probability that u is activated at step t after the cascade ends in A. Since the events that u gets activated at different steps are mutually exclusive, the probability that u ever becomes active by the end of step τ is $\sum_{t=0}^{\tau} ap(u,t)$. By linearity of expectation, $\sigma_\tau(S) = \sum_{u\in V}\sum_{t=0}^{\tau} ap(u,t)$. Hence, the focus is to compute $ap(u,t)$, for which we have the following theorem.

Theorem 3. *Given any u in arborescence A, and any $t \in [0,\tau]$, the* activation probability $ap(u,t)$ *can be recursively computed as follows.*

For base cases when $u \in S$ or $t = 0$,

$$ap(u,t) = \begin{cases} 1 & (u \in S \wedge t = 0) \\ 0 & (u \notin S \wedge t = 0) \\ 0 & (u \in S \wedge 1 \le t \le \tau) \end{cases}$$

When $u \notin S \wedge t \in \{1,\ldots,\tau\}$, $ap(u,t) =$

$$\prod_{w\in N^{in}(u)} \Big(1 - \sum_{t'=0}^{t-2} ap(w,t')p(w,u)[1-(1-m(w,u))^{t-t'-1}]\Big)$$
$$- \prod_{w\in N^{in}(u)} \Big(1 - \sum_{t'=0}^{t-1} ap(w,t')p(w,u)[1-(1-m(w,u))^{t-t'}]\Big), \quad (1)$$

where $N^{in}(u) \subseteq V_A$ is the set of in-neighbors of u in A.

Proof. The base cases ($u \in S$ or $t = 0$) are trivial. When $u \notin S$ and $t \in \{1,\ldots,\tau\}$, for any in-neighbor $w \in V_A$ of u and $t' < t$, $p(w,u)\{1-[1-m(w,u)]^{t-t'-1}\}$ is the probability that w meets u at least once from $t'+1$ to $t-1$ and that (w,u) is live. Since the events that w gets activated at different t' are mutually exclusive, $1 - \sum_{t'=0}^{t-2} ap(w,t')p(w,u)\{1-[1-m(w,u)]^{t-t'-1}\}$ is the probability that u has not been activated by w before or at $t-1$. Note that $\sum_{t'=0}^{-1} ap(w,t')p(w,u)\{1-[1-m(w,u)]^{t-t'-1}\} = 0$, so the above still holds for $t = 1$. Similarly, $\prod_{w\in N^{in}(u)}(1 - \sum_{t'=0}^{t-1} ap(w,t')p(w,u)\{1-[1-m(w,u)]^{t-t'}\})$ is the probability that u has not become active before or at t. Hence, Formula 1 is exactly the probability that u is activated at t, which is $ap(u,t)$. □

The recursion given by Formula 1 can be carried out by dynamic programming, traversing from leaves to the root. In a general in-arborescence, given as input a node u and a deadline constraint τ, the time complexity of calculating $\sum_{t=0}^{\tau} ap(u,t)$ by Formula 1 is polynomial to τ, which is exponential to the size of the input: $\Theta(\log\tau)$ bits. In principle, this does not affect efficiency much as τ is small (5 or 10), and in general much smaller than the size of the graph.

To reduce the amount of computations, a few optimizations can be applied in implementation. Let $path(u)$ be the path from some $s \in S$ in A to u that has the minimum length among all such paths. Note that we only need to compute $ap(u,t)$ for $t \in \{|path(u)|,\ldots,\tau\}$, as u cannot be reached earlier than step $|path(u)|$. In other words, $ap(u,t) = 0$ when $t < |path(u)|$. Also, if $path(u) = \emptyset$ (i.e., does not exist), $ap(u,t) = 0, \forall t$.

For computing $ap(u,t)$ on a *chain* of nodes within an in-arborescence, we derive a more efficient method that reduces the time complexity to polynomial to $\log\tau$. We defer the discussions on this to Section 6.

4 MIA Algorithms for IC-M

The greedy approximation algorithm is too inefficient to use in practice as it lacks of a way to efficiently compute influence spread in general graphs (Sec. 2). To circumvent such inefficiency, we propose two MIA-based heuristic algorithms. The first algorithm is MIA-M (Maximum Influence Arborescence for IC-M) which uses the dynamic programming in Theorem 3 to compute exact influence of seeds. The second one is MIA-C (Maximum Influence Arborescence with Converted propagation probabilities) which first estimates propagation probabilities for pairwise users by combining meeting events, influence events, and the deadline τ, and then uses MIA for IC to select seeds.

Both algorithms first construct a maximum influence in-arborescence (MIIA) for each node in the graph, we calculate influence propagated through these MIIAs to approximate the influence in the original network.

4.1 The MIA-M Algorithm

Before describing the algorithm, we first introduce some necessary notations. For a pair of nodes u, v, let $\mathcal{P}(u,v)$ be the set of all paths from u to v in G. Given a path $\mathsf{P} = \langle u = u_1,\ldots,u_l = v\rangle \in \mathcal{P}(u,v)$, its propagation probability $pp(\mathsf{P}) = \prod_{i=1}^{l-1} p(u_i,u_{i+1})$.

Next, we define the *maximum influence path* from u to v to be $\mathsf{MIP}(u,v) = \mathrm{argmax}_{\mathsf{P}\in\mathcal{P}(u,v)}\, pp(\mathsf{P})$. Note that $\mathsf{MIP}(u,v) = \emptyset$ if $u = v$ or $\mathcal{P}(u,v) = \emptyset$. In addition, we require at most one $\mathsf{MIP}(u,v)$ for each u, v pair, with ties broken in a consistent way. To compute MIPs, notice that if we transfer influence probability $p(u,v)$ into edge weight $-\log p(u,v)$, computing $\mathsf{MIP}(u,v)$ is equivalent to finding the shortest path from u to v in G, and this can be done efficiently by Dijkstra's algorithm.

For MIA-M, we also introduce the "augmented" length $\ell_A(\mathsf{P})$ of a path P to take meeting events and the deadline constraint into account. Consider an edge $(u_i,u_j) \in \mathsf{P}$. Due to random meeting events, after u_i activates at step t, its influence will not propagate to u_j exactly at $t+1$. Instead, the propagation may take multiple steps and the number of such steps is a random variable $X_{i,j}$, which can also be interpreted as the number of Bernoulli trials needed to get the first meeting between u_i and u_j after u_i's activation. Clearly, $X_{i,j}$ follows the geometric distribution, with success probability $m(u_i,u_j)$, expectation $\frac{1}{m(u_i,u_j)}$, and standard deviation $\frac{\sqrt{1-m(u_i,u_j)}}{m(u_i,u_j)}$. Here we propose to estimate the value of $X_{i,j}$ by $\frac{1}{m(u_i,u_j)} - \frac{\sqrt{1-m(u_i,u_j)}}{m(u_i,u_j)}$, and define the *augmented path length* $\ell_A(\mathsf{P})$ of P to be the sum of all estimated values of the random variables (one per edge) along P: $\ell_A(\mathsf{P}) = \sum_{(u_i,u_j)\in\mathsf{P}}\left(\frac{1}{m(u_i,u_j)} - \frac{\sqrt{1-m(u_i,u_j)}}{m(u_i,u_j)}\right)$. We

empirically verify that this is a good choice for $\ell_A(\mathsf{P})$.

Constructing Arborescences. For any node v in G, we approximate the influence to v from all $u \in V \setminus \{v\}$ using the *maximum influence in-arborescences (MIIA)* of v. To construct the MIIA rooted at v, we first take the union over the maximum influence paths to v over all $u \in V \setminus \{v\}$. After that, two pruning steps will be done. First, we remove paths whose propagation probability is below a pre-defined influence threshold $\theta \in (0, 1]$, which controls the size of the local influence region and is a trade-off between efficiency and seed set quality. Second, to take the effect of deadline into account, we eliminate paths whose augmented length is greater than τ.

Definition 1 (Maximum Influence In-Arborescence). *Given an influence threshold $\theta \in (0, 1]$ and a deadline constraint $\tau \in \mathbb{Z}_+$, the maximum influence in-arborescence of any node $v \in V$ is*

$$\mathsf{MIIA}_\tau(v, \theta) = \cup_{u \in V, pp(\mathsf{MIP}(u,v)) \geq \theta, \ell_A(\mathsf{MIP}(u,v)) \leq \tau} \mathsf{MIP}(u, v)$$

The full MIA-M is described in Algorithm 2, where $MG(u) = \sigma_\tau(S \cup \{u\}) - \sigma_\tau(S)$ is the marginal influence of u w.r.t to seed set S, $MG(u, v)$ is the marginal influence of u on a specific v, and $realized(v)$ is the cumulative influence realized on v by S. Also, for each $u \in V$, $\mathsf{InfSet}(u) = \{v \in V : u \in \mathsf{MIIA}_\tau(v, \theta)\}$. After constructing $\mathsf{MIIA}_\tau(v, \theta)$ and using Theorem 3 to obtain $\sigma_\tau(\{v\})$ for all $v \in V$ (lines 4-10), the algorithm selects k seeds iteratively in a greedy manner, and uses Theorem 3 to update the marginal gain of nodes in related MIIAs (lines 11-20). Specifically, after u is picked as a seed, the activation probability of all $v \in \mathsf{InfSet}(u)$ goes up, and thus we need to update the marginal gain of all $w \in \mathsf{MIIA}_\tau(v), \forall v \in \mathsf{InfSet}(u)$.

Time Complexity: Let $n_{m\theta} = \max_{v \in V} |\mathsf{MIIA}_\tau(v, \theta)|$, and $n_{s\theta} = \max_{v \in V} |\mathsf{InfSet}(v)|$. Also suppose that the maximum running time to compute $\mathsf{MIIA}_\tau(v, \theta)$ for any $v \in V$ by Dijkstra's algorithm is $t_{m\theta}$. Thus, MIA-M runs in $O(|V|(t_{m\theta} + n_{m\theta}\tau^3) + kn_{m\theta}n_{s\theta}(n_{m\theta}\tau + \log |V|))$.

Algorithm 2: MIA-M ($G = (V, E)$, k, θ, τ)

```
1  S ← ∅;
2  ∀v ∈ V, MG(v) ← 0 and realized(v) ← 0;
3  ∀v ∈ V, MIIA_τ(v, θ) ← ∅ and InfSet(v) ← ∅;
4  foreach v ∈ V do
5      Compute MIIA_τ(v, θ) (Definition 1);
6      foreach u ∈ MIIA_τ(v, θ) do
7          InfSet(u) ← InfSet(u) ∪ {v};
8          Compute ap(v, t, {u}, MIIA_τ(v, θ)), ∀t ≤ τ
           (Theorem 3);
9          MG(u, v) ← Σ_{t=0}^{τ} ap(v, t, {u}, MIIA_τ(v));
10         MG(u) ← MG(u) + MG(u, v);
11 for i = 1 → k do
12     u ← argmax_{v ∈ V\S} MG(v);
13     S ← S ∪ {u};
14     foreach v ∈ InfSet(u) do
15         realized(v) += MG(u, v);
16         foreach w ∈ MIIA_τ(v, θ) do
17             Compute ap(v, t, S ∪ {w}, MIIA_τ(v, θ)),
               ∀t ≤ τ (Theorem 3);
18             MG_new(w, v) ← [Σ_{t=0}^{τ} ap(v, t, S ∪
               {w}, MIIA_τ(v, θ))] − realized(v);
19             MG(w) += MG_new(w, v) − MG(w, v);
20             MG(w, v) ← MG_new(w, v);
```

4.2 The MIA-C Algorithm

We now discuss our second algorithm, MIA with Converted propagation probability (MIA-C). It consists of two steps. First, for each $(u, v) \in E$, we estimate a converted propagation probability $p_c(u, v)$ that incorporates meeting probability $m(u, v)$, influence probability $p(u, v)$, and deadline τ, with the intention to *simulate* the influence spread under the IC-M model in the original IC model. Second, after obtaining all $p_c(u, v)$, we treat these converted probabilities as parameters for the IC model and run the MIA algorithm proposed for IC to select k seeds.

In the IC-M model with deadline τ, the value of $p_c(u, v)$ depends on $p(u, v)$, $m(u, v)$, and τ. We use the following conversion function to obtain $p_c(u, v)$:

$$p_c(u, v) = p(u, v) \cdot [1 - (1 - m(u, v))^\beta], \qquad (2)$$

where $\beta \in [1, \tau]$ is the parameter used to estimate the number of meeting attempts. If β is 1, $p_c(u, v) = p(u, v) \cdot m(u, v)$, in which case we are pessimistic that u has only one chance to meet v (the minimum possible, assuming u itself activates before τ). On the other hand, if β is τ, $p_c(u, v) = p(u, v) \cdot [1 - (1 - m(u, v))^\tau]$, for which we are optimistic that u has τ chances to meet v (the maximum possible). To achieve a balanced heuristic, we let $\beta = \frac{\tau}{2}$ for all pairs of u, v, and experiments show that this estimation turns out to be more effective than other choices in most cases.

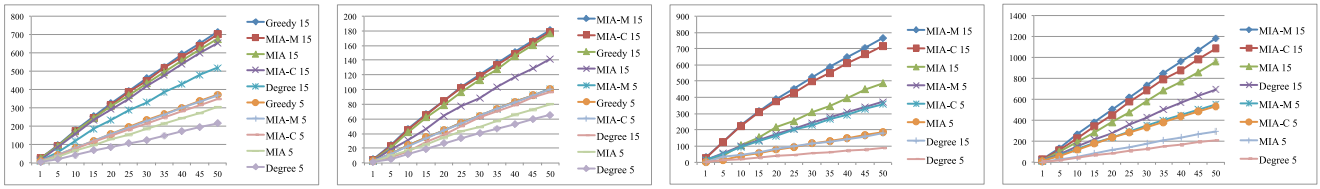
After the probability conversion step, we utilize MIA (Algorithm 4, (Chen, Wang, and Wang 2010)) to find the seed set, making MIA-C take the advantage of updating marginal gains of nodes in an extremely efficient manner.

The time complexity of converting probabilities is $O(|E|)$, and second part of MIA-C has the same time complexity as MIA, which is $O(|V|t_{m\theta} + kn_{m\theta}n_{s\theta}(n_{m\theta} + \log |V|))$.

5 Empirical Evaluations

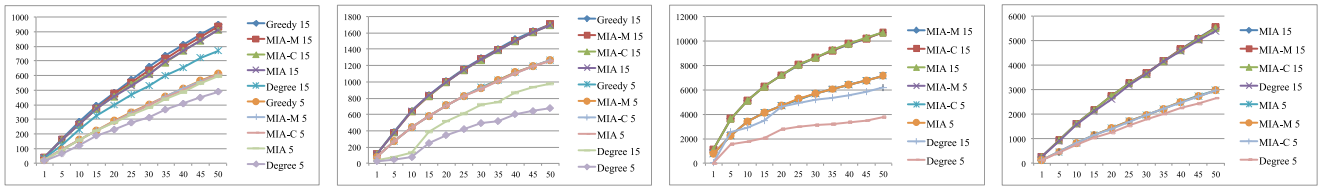
We conduct experiments on four real-world network datasets to evaluate MIA-M and MIA-C, and compare them to the greedy approximation algorithm and a few other baselines.

Dataset Preparation. We use four network datasets. The statistics of the datasets are presented in Table 1. NetHEPT, standard in this area, is a collaboration network extracted from arXiv.org's High Energy Physics Theory section. DBLP is a much larger collaboration network from the DBLP Computer Science Bibliography. Nodes in NetHEPT and DBLP are authors; if u and v have collaborated, we draw arcs in both directions. WikiVote is a who-vote-whom network from Wikipedia. If v voted on u for promoting u to adminship, we draw an arc (u, v). Epinions is a who-trust-whom social network from Epinions.com, for which



(a) NetHEPT (b) WikiVote (c) Epinions (d) DBLP

Figure 1: Influence spread (#nodes, Y-axis) against seed set size (X-axis) on graphs with weighted meeting probabilities.



(a) NetHEPT (b) WikiVote (c) Epinions (d) DBLP

Figure 2: Influence spread (#nodes, Y-axis) against seed set size (X-axis) on graphs with uniform random meeting probabilities.

we draw an arc (u, v) if v expressed her trust in u.

Graph Parameters. Influence probabilities are assigned using the Weighted Cascade model[1] (Kempe, Kleinberg, and Tardos 2003), in which $p(u, v) = 1/d^{in}(v)$ where $d^{in}(v)$ is the in-degree of v. Similarly, for meeting probabilities, we also adopt the weighted method: $m(u, v) = c/(d^{out}(u)+c)$, since it is reasonable to deem that the more friends u have, the less probable that u could meet a certain individual in one time unit. Here c is a smoothing constant and we choose it to be 5. In addition, we test on cases where $m(u, v)$ is chosen uniformly at random from $\{0.2, 0.3, \ldots, 0.7, 0.8\}$.

Algorithms Compared. We evaluate MIA-M, MIA-C, the greedy algorithm (Greedy) and the other two: Degree and MIA. For Greedy, we apply CELF and run Monte Carlo for 10000 times. Degree is a heuristic based on the notion of degree centrality that considers high degree nodes as influential ones. It outputs the top-k highest out-degree nodes as seeds. We also test MIA[2], one of the state-of-the-art heuristic algorithms for the standard IC model. For the purpose of comparisons, we let MIA select seeds disregarding meeting probabilities and the deadline constraint entirely, i.e., treating $m(u, v) = 1$ for all edges and $\tau = |V|$. MIA-M, MIA-C, and MIA all use $1/320$ as the influence threshold θ, as recommended by Chen et al. (2010). For MIA-C, we choose $\beta = \frac{\tau}{2}$ since it gives more stable performance (compared to 1 and τ) in most cases.

All experiments were conducted on a server running Microsoft Windows Server 2008 R2 with 2.33GHz Quad-Core Intel Xeon E5410 CPU and 32G memory.

[1]We also do experiments on the Trivalency model (Chen, Wang, and Wang 2010). Since the results are similar, we omit it here.

[2]We also test the Prefix-excluding MIA algorithm (Chen, Wang, and Wang 2010). Since the results are similar to those for MIA, we omit it here.

Dataset	NetHEPT	WikiVote	Epinions	DBLP
Num. of nodes	15K	7.1k	75K	655K
Num. of edges	62K	101K	509K	2.0M
Average degree	4.12	26.6	13.4	6.1
Maximum degree	64	1065	3079	588
#Connected components	1781	24	11	73K
Largest component size	6794	7066	76K	517K
Avg. component size	8.55	296.5	6.9K	9.0

Table 1: Statistics of Network Datasets.

5.1 Experimental Results and Analysis

We compare the five algorithms on quality of seeds sets and running time. The deadline τ is set to 5 (relatively short time horizon) and 15 (relatively long time horizon) in all results reported. Greedy is too slow to finish on Epinions and DBLP within a reasonable amount of time (three days).

Quality of Seed Sets. The quality of seed sets is evaluated based on the expected influence spread achieved. To ensure fair and accurate comparisons, we run MC simulations 10000 times to get the "ground truth" influence spread of all seed sets obtained by various algorithms. Fig. 1 and 2 illustrate influence spread achieved on datasets with weighted and uniform random meeting probabilities, respectively.

On graphs with weighted meeting probabilities, except for Greedy, MIA-M has the highest seed set quality, while MIA-C is the second best in most test cases. MIA-M performs consistently better than Degree and MIA, e.g., on Epinions, the influence of 50 seeds by MIA-M is 99.4% ($\tau = 5$) and 53.6% ($\tau = 15$) higher than those by MIA. On NetHEPT and WikiVote, MIA-M produces seed sets with equally good quality as Greedy does, e.g., on WikiVote, when $\tau = 5$ they both achieve influence spread of 101; when $\tau = 15$, MIA-M (181) even achieves 3% higher than Greedy (175).

When meeting probabilities are assigned uniformly at random, seed sets by MIA-M, MIA-C, and MIA tend to have matching influence, all being close to Greedy and better than Degree. Most often, MIA-M is marginally better than MIA-

Algorithm	NetHEPT		WikiVote		Epinions		DBLP	
	5	15	5	15	5	15	5	15
Greedy	40m	1.3h	22m	28m	-	-	-	-
MIA-M	1.6s	15s	7.9s	43s	47s	5.1m	6.6m	10m
MIA-C	0.3s	0.3s	0.4s	0.5s	2.7s	3.3s	24s	33s
MIA	0.3s	0.3s	1.4s	1.4s	12s	13s	40s	41s

Table 2: Running Time (Weighted Meeting Probability)

C and MIA. The reason why MIA catches up is that it picks seeds assuming that all $m(u, v) = 1$ and $\tau = |V|$, and in expectation those seeds still have high influence under uniform random $m(u, v)$'s, where the expectation is taken over all possible meeting events outcomes. Also, when τ is large, the time-critical effect of the deadline is diminishing, so MIA tends to perform better with larger τ.

In reality, however, meeting probabilities between individuals in social networks may be quite different from being uniform random, and over all test cases it can be seen that MIA-M and MIA-C are more stable than MIA. For certain meeting probabilities, MIA has poor performance.

Running Time. We demonstrate the running time results on weighted meeting probability datasets in Table 2 (the uniform random cases are similar). Greedy takes 0.5 to 1.3 hours to finish on NetHEPT and WikiVote, and fails to complete in a reasonable amount of time (three days) on Epinions and DBLP with $\tau = 5$. Degree finishes almost instantly in all test cases so it is not included in the table.

MIA-C and MIA are three orders of magnitude faster than Greedy, since both benefit from the linearity rule of activation probabilities when updating marginal gains (Chen, Wang, and Wang 2010). MIA-C is more efficient because its converted probabilities are smaller than the original influence probabilities used in MIA, and hence arborescences are smaller for MIA-C under the same influence threshold $(1/320)$. MIA-M is two orders of magnitude faster than Greedy, and is scalable to large graphs like Epinions and DBLP. It is slower than MIA-C and MIA because its dynamic programming procedure computes activation probabilities associated with steps, and hence is not compatible with the linearity rule of activation probabilities.

6 Conclusions and Discussions

In this work, we extend the classical Independent Cascade model to study time-delayed influence diffusion and we consider the time-critical influence maximization problem under our proposed IC-M model. We prove the submodularity of IC-M, and propose fast heuristics MIA-M and MIA-C to find seed sets efficiently and effectively, and we validate them experiments on four network datasets.

Future Work. There are a number of extensions and future directions on time-critical influence maximization. One problem is to look into more efficient computation of influence spread in tree structures, with time complexity in polynomial to $\log \tau$. We have obtained results on chain graphs (see our full technical report (Chen, Lu, and Zhang 2012)). Since chain cases are common in the execution of MIA-based algorithms, this could already improve the running time of MIA-M.

Another extension is to use login events to model time-delayed influence diffusion, which could fit better into online social networks. Specifically, each user has a probability of logging on to the system, and only after logging in, users could be influenced by the word-of-mouth of active friends. Incorporating login probabilities into the IC model turns out to be more challenging than doing so for meeting probabilities, as it introduces dependencies in activation events. We have obtained partial results using more complicated dynamic programming methods to deal with this case.

The third extension is to consider time delays in the Linear Threshold (LT) model or even more general diffusion models. We are able to show submodularity for the LT model extension (Chen, Lu, and Zhang 2012), and are looking into extensions to the General Threshold model.

References

Chen, W.; Lu, W.; and Zhang, N. 2012. Time-critical influence maximization in social networks with time-delayed diffusion process (full technical report). *CoRR* abs/1204.3074.

Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 1029–1038.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *KDD*, 199–208.

Domingos, P., and Richardson, M. 2001. Mining the network value of customers. In *KDD*, 57–66.

Goyal, A.; Bonchi, F.; and Lakshmanan, L. V. S. 2012. A data-based approach to social influence maximization. *PVLDB* 5(1):73–84.

Iribarren, J. L., and Moro, E. 2009. Impact of human activity patterns on the dynamics of information diffusion. *Physics Review Letters* 103:038702.

Karsai, M.; Kivelä, M.; Pan, R. K.; Kaski, K.; Kertész, J.; Barabási, A.-L.; and Saramäki, J. 2011. Small but slow world: How network topology and burstiness slow down spreading. *Physics Review Letters* 83:025102.

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *KDD*, 137–146.

Kimura, M.; Saito, K.; and Nakano, R. 2007. Extracting influential nodes for information diffusion on a social network. In *AAAI*, 1371–1376.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J. M.; and Glance, N. S. 2007. Cost-effective outbreak detection in networks. In *KDD*, 420–429.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294.

Richardson, M., and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *KDD*, 61–70.