Diffusion Maximization in Evolving Social Networks

Nathalie T. H. Gayraud  
Department of Computer Science and Engineering  
University of Ioannina  
Ioannina, Greece  
ngairo@cs.uoi.gr

Evaggelia Pitoura  
Department of Computer Science and Engineering  
University of Ioannina  
Ioannina, Greece  
pitoura@cs.uoi.gr

Panayiotis Tsaparas  
Department of Computer Science and Engineering  
University of Ioannina  
Ioannina, Greece  
tsap@cs.uoi.gr

ABSTRACT

Diffusion in social networks has been studied extensively in the past few years. Most previous work assumes that the underlying network is a static object that remains unchanged as the diffusion process progresses. However, there are several real-life networks that change dynamically over time. In this paper, we study diffusion on such evolving networks and extend the popular Independent Cascade and Linear Threshold models to account for network evolution. In particular, we introduce two natural variations, a persistent and a transient one, to capture diffusions of different types. We consider the problem of influence maximization where the goal is to select a few influential nodes to initiate a diffusion with maximum spread. We show that, surprisingly, when considering evolving networks the diffusion function is no longer submodular for the transient models, and not even monotone for the transient Independent Cascade model. We also show that, depending on the model, delaying the activation of the initiators may improve diffusion. Our experiments, using three real datasets, demonstrate the effect of network evolution on the diffusion process, and highlight the importance of timing in the selection process.

Categories and Subject Descriptors

J.4 [Computer Applications]: Social and behavioral sciences; H.2.8 [Database Applications]: Data Mining; H.4 [Information Systems Applications]: Miscellaneous

Keywords

Diffusion Maximization; Evolving Social Networks

1. INTRODUCTION

Information propagation and social influence have long been important topics for communication media and social sciences [10]. The growth of online social networks such as Facebook, Twitter, and Instagram, and the importance of influence and diffusion in viral marketing applications [8][12], has intensified the research interest in the topic. A problem that has attracted considerable attention in this area is that of identifying “influencers”: a small set of individuals that will initiate the diffusion of a trend and maximize its spread in the social network. This is a problem of great research interest, with immediate practical applications.

The problem of diffusion maximization was first defined in the seminal works of Domingos and Richardson [8] and Kempe et al. [15]. The work in [15] laid the theoretical and algorithmic foundations for understanding and addressing the problem. The paper introduced two basic diffusion models, the Independent Cascade (IC) model and the Linear Threshold (LT) model, and it formulated the influence maximization problem as a discrete optimization problem. They showed that the problem is NP-hard, but thanks to the submodularity property of the diffusion spread there exists a greedy algorithm with a constant approximation ratio.

The work in [15] was followed by an avalanche of work that proposed improvements or modifications to the basic models (e.g., [18][9][19][7]). Most of the follow-up work considers the network as a static object that remains unchanged as the diffusion process progresses. However, this assumption is often not true. There are many real-life networks that evolve dynamically, with nodes joining and leaving the network, and edges being formed and destroyed over time. Examples include mobile contact networks, location-based networks, collaboration networks and many more. Many of these networks evolve in predictable ways [6], enabling us to incorporate network evolution in the analysis and modeling of diffusion on the network.

To circumvent the evolving nature of the network, previous approaches aggregate the multiple instances over time into a single static graph. However, such approaches disregard the importance of timing in the diffusion process, that is, the importance of information being at the right place, at the right time, so that there is a path in the network on which to propagate. As we will see, network evolution has a significant effect on the process of information diffusion, and timing is critical in the correct selection of influencers.

In this work, we address the problem of diffusion maximization on evolving graphs, and we make the following contributions:

• We define the Independent Cascade and the Linear Threshold models on evolving networks. We introduce two variants for each model, a persistent and a transient one, to account for diffusions of different temporal nature.
We consider the problem of diffusion maximization on evolving graphs and study theoretically its properties under the different models. We prove that, surprisingly, the optimization function is not submodular for the transient models, and for the transient Evolving Independent Cascade model it is not even monotone. We also show that, for some models, delayed activation of the seed nodes may improve the diffusion spread.

We study experimentally the diffusion process on three real evolving datasets. Our experimental evaluation demonstrates the effect of network evolution on diffusion, as well as the importance of timing of node activations.

The rest of the paper is structured as follows. Section 2 reviews related work. In Section 3 we introduce preliminary definitions and in Section 4 we formulate our problem. In Sections 5 and 6 we define the Evolving Independent Cascade and the Evolving Linear Threshold models and study their properties. In Section 7 we report the results of our experimental evaluation. Section 8 concludes the paper.

2. RELATED WORK

The pioneering work of Domingos and Richardson [8] and Kempe et al. [13] generated significant amount of research [15, 4, 5, 7], focusing mostly on variations of the models, and efficient implementations of the algorithms. Surprisingly, there is little research on diffusion on evolving graphs.

Evolving networks: Most closely related to our work are the works of Zhuang et al., [26] and Aggarwal et al., [1] who, as in this paper, view an evolving graph as a sequence of graphs \( \{G_i\} \) at different time instances. However, the work of Zhuang et al., [26] addresses a different problem. They apply diffusion maximization independently in each static graph \( G_i \), and assuming that only the initial graph \( G_0 \) is fully known, they ask which 6 nodes to probe to get the edges incident on these nodes at time \( t \), so as to approximate the diffusion on \( G_i \). Our goal is maximizing diffusion over the sequence of graphs as a whole assuming that diffusion and evolution run in parallel.

The focus of the work of Aggarwal et al., [1] is on the efficient estimation of the influence spread by avoiding calculations among graph instances that are structurally similar. Our focus is not on algorithms, but instead on modeling evolution and understanding diffusion maximization in evolving networks under different models. In fact, we show that depending on the model, submodularity may not hold, thus raising the need for new algorithmic approaches.

Another work that considers diffusion on evolving graphs is that of Albano et al., [2]. They make a distinction between extrinsic time measured in seconds and intrinsic time where a time unit corresponds to a new edge appearing in the graph. Their goal is different from ours: they differentiate between diffusion and graph evolution, e.g., to understand whether an increase in diffusion is due to a sudden growth in the graph.

Time-varying and continuous networks: The notion of time has been introduced in the analysis of information diffusion, as a way of extending the basic diffusion model to capture the duration or latency of diffusion. The duration of diffusion is usually modeled by associating with each edge or node, in addition to its activation probability, a latency function that determines when the node reacts to an activation. The problem of influence maximization and estimation is studied for both discrete time (e.g., Liu et al., [19] and Chen et al., [3]) as well as for continuous time (e.g., Gomez-Rodriguez et al. [14] and Du et al., [2]) where the latency per edge entails a random spreading time drawn from a distribution over the time of activation. The recent work of Xie et al., [25] extends the continuous model to capture dynamic properties, but still the diffusion function remains submodular. An orthogonal line of research focuses on learning the influence graph by inferring the influence probabilities, as well as the latency functions of each edge, e.g., the work of Gomez-Rodriguez et al., [13]. The key difference of our work from previous work on time-varying or temporal graphs is that in time-varying graphs, the effect of time on the propagation probability is with respect to the activation time of each node. In our work, we assume that the network changes over time, independently of the diffusion process.

Epidemics: Another line of research focuses on virus propagation on dynamic networks using epidemics models. Such research addresses different problems such as determining the epidemic threshold [23]. Stattnner et al., [24] studies the spread of infectious diseases by simulating the infection transmission using the SIR model (a model similar to Independent Cascade) on evolving networks. Their experimental results showed that changes of the underlying network greatly affect the spread of diseases.

3. PRELIMINARIES

In this section, we introduce the necessary concepts for describing the graph evolution and diffusion processes on evolving graphs.

Evolving graphs: We model an evolving graph as a sequence of \( n \) graphs \( G = \{G_1, G_2, ..., G_m\} \), defined over the same set of nodes \( V \), where the set of edges differs between time-stamps. That is, \( G_i = (V, E^i) \), where \( E^i \subseteq V \times V \). Essentially, we can think of the graph sequence, as a sequence of sets of edges \( E^1, E^2, ..., E^n \) over the same set of nodes \( V \). Note that our model is general enough to allow for the addition and deletion of both edges and nodes in the graph over time. The set of nodes \( V \) contains all the nodes that appear in any snapshot. If a node is not present in a snapshot, there are no edges incident to it. Furthermore, our model can easily be extended to capture evolving probabilistic graphs, where at every time-step, an edge appears in the graph with some probability, that changes over time.

An example of a graph sequence \( G = \{G_1, G_2, G_3\} \) with three snapshots is shown in Figure 1. Given a sequence, we define the union graph \( G_U = (V, E_U) \), where \( E_U = \bigcup_{i=1}^{n} E^i \).
Diffusion and network evolution: In the following, we consider two commonly used models for diffusion: Independent Cascade (IC) and Linear Threshold (LT). We describe the models in detail in Sections 5.1 and 5.2 respectively. At a high level, both models assume that nodes are in two states: either active or inactive. Diffusion starts with a set of active nodes \( A^0 \) and then proceeds in discrete steps. At each diffusion step \( t \), given the already active nodes \( A^{t-1} \), depending on the graph topology and the diffusion model, a new set of nodes \( S^t \) is activated, resulting in a new set of active nodes \( A^t = A^{t-1} \cup S^t \). The process continues until no more activations are possible.

Regardless of the diffusion model we consider, in order to define the diffusion process on an evolving graph, the first issue that we need to address is to define the notion of time. We have two distinct time measures that run in parallel: the graph evolution time, where a time-step is defined by a graph instance in the graph sequence, and the diffusion time, where a time-step is defined by one step in the diffusion process. We need to decide how to synchronize these two time-tracks. That is, we need to decide how many diffusion steps can happen on a graph instance \( G^t \), or how many graph instances a diffusion step spans.

In this work, we make the decision to have the evolution time and the diffusion time run in lock-step. In our model, one time-step \( t \) corresponds to one graph instance \( G^t \), on which a single diffusion step takes place. That is, entering time-step \( t \) there is a set of nodes \( A^{t-1} \subseteq V \) that are active. Similar to the case of a static graph, a diffusion step happens on the graph \( G^t \), and a new set of vertices \( S^t \) are activated, defining the set \( A^t = A^{t-1} \cup S^t \). We then move on to time-step \( t + 1 \) and graph \( G^{t+1} \). Note that since the set of nodes is the same for all graph instances, the notion of a node \( u \) that is inactive at time \( t-1 \) and active at time \( t \) is well defined. Once a node becomes active, it remains active for all following steps. The diffusion process continues for as many steps as the graph sequences instances.

Our definition is general enough to include the possibility that the diffusion time runs faster than the evolution time. Assume for a example that \( s \) diffusion steps are executed on a graph instance \( G^t \). We can simulate this process by adding \( s \) copies of the graph \( G^t \) in the sequence, and assume again that evolution and diffusion time are synchronized. Similarly, if diffusion time is slower than evolution time, we can aggregate the multiple graph instances that correspond to a single diffusion step, and assume again that diffusion and evolution time are synchronized.

Transient and persistent diffusion: Another issue that arises when considering diffusion on a time-evolving graph is to determine the temporal nature of diffusion. In all diffusion models, when a node \( u \) gets activated, the model makes a decision as to whether the neighbors of \( u \) will be affected. When the graph is static, this decision can be made at the time that \( u \) is activated. When the graph evolves over time, the neighbors of \( u \) also change over time. What is the time-span in which node \( u \) can affect its neighbors?

This question is not a simple technicality: the answer determines the temporal nature of the diffusion. In this work we consider two cases: (1) Transient diffusion processes, where the effect of a node activation is “local” in time. This models the case where the diffusion capability is short-lived and localized in time; (2) Persistent diffusion processes, where the activation of a node has an effect that lasts beyond a single time instance. This models the case where the diffusion capability can persist over time. We elaborate on these issues when we describe the specific models.

4. PROBLEM DEFINITION

We now define the diffusion maximization problem that we consider in this work. Similar to prior work on diffusion maximization, we assume that there is an item that we want to spread in the network. This may be a product, an idea, or a piece of information. Our goal is to select a small set of influential nodes in the network that will initiate the diffusion, such that the spread of the item is maximized. We will refer to this set of nodes as the initiators, or influencers, and denote it as \( I \).

In the following, we use \( A^t \) to denote the set of nodes that are active in graph \( G^t \) after the diffusion process has been completed. Given the sequence of graphs, and the diffusion model, the set \( A^t \) depends on the set of influencers \( I \) selected to be activated. We define \( \sigma(D, I) \) to be the number of activated nodes under the diffusion model \( D \) for the set of initiators \( I \). We call \( \sigma(D, I) \) the spread of the diffusion for the set \( I \). Our goal is to select a set \( I \) of \( k \) nodes that maximizes \( \sigma(D, I) \).

Since we have an evolving graph, when selecting a node \( v \) to activate, we must also select the time \( t \) at which we want to activate it. Activating node \( v \) at time \( t \) means that the node \( v \) is added to the set of active nodes \( A^t \), and it can influence its neighbors in future time-steps. It is also possible to activate node \( v \) at time \( t = 0 \) which means that \( v \) is active entering the graph evolution and diffusion process. We use \( v^t \) to denote the instance of node \( v \) in graph \( G^t \) at time \( t \). The selection algorithm is thus required to select appropriate instances of \( k \) nodes from the set \( V_T = \{ v_1, ..., v_k \} \), \( v_i \neq v_j \), of \( k \) node instances to be activated, such that \( \sigma(D, I) \) is maximized.

We can now define the following problem, which we call the Spread Maximization on Evolving Graphs problem (EvolveMaxSpread).

Problem 1 (EvolveMaxSpread). Given a sequence of graphs \( G = \{ G^1, G^2, ..., G^n \} \) and an integer \( k \), for a given diffusion model \( D \), find a set \( I = \{ v_1, ..., v_k \} \), \( v_i \neq v_j \), of \( k \) node instances to be activated, such that \( \sigma(D, I) \) is maximized.

Our problem contains as a special case the problems defined in [15], since we can simulate the diffusion process in a static graph \( G \), as the diffusion on a sequence of graphs, where all graph instances are copies of \( G \), and the length of the sequence is sufficient for the diffusion to be completed. Therefore, we can conclude that the problem is NP-hard.

Following the work in [15], most works that consider variations of the diffusion maximization problem on a static graph are able to derive a constant factor approximation algorithm by making use of the fact that the spread function is monotone and submodular. Let \( f: 2^V \rightarrow \mathbb{R} \) denote a set function that maps a subset \( S \subseteq V \) of the nodes to a real number. We say that the function \( f \) is monotone if \( f(S \cup \{v\}) - f(S) \geq 0 \)
for all $S \subseteq V$, $v \in V \setminus S$. We say that function $f$ is submodular if $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for all $A \subseteq B$, $v \in V \setminus B$. The problem of finding a set $S$ of size $k$ that maximizes $f(S)$ is NP-hard for several submodular functions that arise in practice [19]. However, it is well known [21] that a greedy hill-climbing algorithm that builds a set incrementally by adding each time the element that yields the maximum increase in $f$, produces a solution that has approximation factor $(1 - 1/e)$ of the optimal, where $e$ is the base of the natural logarithm. In the following, we show that, surprisingly, depending on the diffusion model, the spread function is not always submodular, and in some cases not even monotone.

As we have already discussed, when selecting the initiator set $I$, we need to select not only the nodes to activate but also the time at which to activate them. For some of the diffusion models we consider, the best time to activate a node so as to maximize the spread is as early as possible, that is, at time $t = 0$. In this case we say that the model is timing-insensitive. Formally, a diffusion model $D$ is timing-insensitive if for any graph sequence $\mathcal{G}$, and any initiator set $I = \{v_1^i, \ldots, v_k^i\}$, for the initiator set $I^0 = \{v_0^0, \ldots, v_k^0\}$ we have $\sigma_D(I) \leq \sigma_D(I^0)$. We will otherwise say that the diffusion model is timing-sensitive.

5. EVOLVING MODELS

In this section, we introduce our diffusion models that extend the Independent Cascade (IC) and the Linear Threshold (LT) models for evolving networks. We also study the properties of the diffusion spread function for each of the models, and the sensitivity to the timing of the activation of the initiators.

5.1 Evolving IC Model

In the case of a static graph, diffusion under the IC model proceeds in discrete steps, where at step $t$ a new set of nodes $S^t$ is activated. Entering time-step $t$, the nodes in the set $S^{t-1}$ (where $S^{t-1} = A^0$ for $t = 1$, i.e., the set of active nodes at time zero) are said to be infectious. During time-step $t$, the nodes in $S^{t-1}$ have a single chance to activate their inactive neighbors. Node $u \in S^{t-1}$ activates an inactive node $v$ over the edge $(u, v)$ with probability $p_{uv}$. If the activation is successful then $v$ is added to the set $S^t$ (and $A^t$). After step $t$, node $u$ does not attempt to activate any of its neighbors.

We will now define two variants of the IC model for the case of evolving graphs. We will collectively refer to these models as the Evolving Independent Cascade model and denote it by EIC.

5.1.1 Transient EIC Model

In the first variant of the model, we assume that a node $u$ can activate its neighbors only immediately after the time instance that it becomes active. In this case, the diffused item and the activation capability of the nodes in the network are transient. For example, consider an infectious disease that is transmitted through a human contact network. When a node becomes infected it has a probability of infecting its neighbors, and then it becomes inoculated. We refer to this model as the Transient Evolving Independent Cascade model, and denote it by tEIC.

Formally, similar to the static case, at step $t$ the elements that are infectious in $S^{t-1}$ are given a single chance to activate their inactive neighbors, and node $u \in S^{t-1}$ activates the inactive node $v$ over the edge $(u, v) \in E^t$ with probability $p_{uv}$. This yields a new set of recently activated and infectious nodes $S^t$. The difference in the evolving case is that at each time-step the graph is different, and the neighbors of node $u$ are defined over the graph $G^t$.

This seemingly small variation makes a big difference in the properties of the model. We show that for the transient EIC model, the spread function is no longer monotone and submodular.

**Lemma 1.** The function $\sigma_{EIC}$ is neither monotone nor submodular.

**Proof.** For the proof we will construct a graph sequence $\mathcal{G}$ for which the function $\sigma_{EIC}$ is neither monotone nor submodular. For simplicity we will assume that all diffusion probabilities $p_{uv}$ are 1, that is, if an edge $(u, v)$ is present in the graph then it will cause the activation of a node. The set of nodes $V$ consists of $N + 6$ nodes, $V = \{v_1, \ldots, v_6, u_1, \ldots, u_N\}$, and we have a sequence of 4 graphs $G^1, G^2, G^3, G^4$ on these nodes. The sequence for $N = 3$ is shown in Figure 2. The four graphs are identical except for the fact that in $G^2$ there is also the edge $(v_3, v_4)$. A key property of the sequence is that nodes $u_1, \ldots, u_N$ are connected only to node $v_4$.

Figure 3 shows the diffusion process for the initiator set $I = \{v_1^0\}$. The dark (red) colored nodes are active nodes that are infectious when entering a given step. The light
time becomes active at the \( t_{EIC} \) model is graph act competitively, and it is possible for one diffusion is given a chance to activate another node not successful it will not attempt to activate further, the diffusion that stared from node \( v_1 \) now stops at node \( v_2 \) and does not proceed any further. Intuitively, the activation of node \( v_5 \) at time \( t = 0 \) causes a premature activation of the node \( v_3 \) which then blocks the diffusion initiated at node \( v_1 \). Therefore, we have that \( \sigma_{EIC}(\{v_0^0, v_1^0\}) = 4 < \sigma_{EIC}(\{v_0^0\}) \) proving that \( \sigma_{EIC} \) is not monotone.

The same sequence can be used to prove that \( \sigma_{EIC} \) is not submodular. Consider the addition of node \( v_2^0 \) to the Initiator set \( I = \{v_0^0\} \). The activation of node \( v_0 \) will cause the nodes \( u_4 \) and \( u_1, \ldots, u_5 \) to be activated earlier, however it has no effect on the overall spread since these nodes would have been activated anyway. Therefore, the increase in spread is \( \sigma_{EIC}(\{v_0^0, v_2^0\}) - \sigma_{EIC}(\{v_0^0\}) = 1 \), corresponding to the activation of \( v_6 \). However, adding \( v_0 \) to the Initiator set \( I = \{v_0^0, v_2^0\} \) results in activating \( N + 1 \) additional nodes, whose activation was previously blocked. Thus, \( \sigma_{EIC}(\{v_0^0, v_2^0, v_0^0\}) - \sigma_{EIC}(\{v_0^0, v_2^0\}) = N + 3 \), meaning that \( \sigma_{EIC} \) is not submodular.

The example demonstrates the importance of timing in the activation of nodes in an evolving graph. Node \( v_5 \) must become active at exactly time \( t = 2 \) in order to activate \( v_1 \) at \( t = 3 \), which in turn can activate nodes \( u_4, \ldots, u_N \) at time \( t = 4 \). Diffusions originated from different nodes in the graph act competitively, and it is possible for one diffusion to block another, thus reducing the overall spread. Clearly, the \( tEIC \) model is timing-sensitive.

5.1.2 Persistent EIC Model

In the second variant of the model, we assume that the item to be diffused, and the interest of the nodes in the item are persistent. A node \( u \) that becomes active at time \( t \) is given a chance to activate another node \( v \) at the first time instance after \( t \) that \( u \) and \( v \) become connected. For example, in a social network, a user that adopts a product will show it to her friends the first time that they meet, affecting their decision process. We refer to this model as the Persistent Evolving Independent Cascade model, and we denote it by \( pEIC \).

Formally, consider a node \( u \) that becomes active at time \( t \). For a node \( v \), let \( t_{uv} \geq t \) denote the earliest time instance after time \( t \) where there is an edge between \( u \) and \( v \) \( (t_{uv} \) is not defined if there is no such edge). If \( v \) is not active at time \( t_{uv} \), node \( u \) tries to activate \( v \) with probability \( p_{uv} \). If not successful it will not attempt to activate \( v \) again for any \( t' > t_{uv} \).

For the persistent EIC model we can prove that the spread function is monotone and submodular when the activation probabilities per edge are constant over time, that is, \( p_{uv} = p_{vu} \) for every edge \((u, v)\), for all graph instances \( G_t \), where \((u, v) \in E^t \).

For the proof we make use of the union graph \( G_U = (V, E_U) \) defined in Section 3 consisting of the union of all the graphs in the sequence. Similar to the work in [15], we assume that all random choices are made in advance. That is, for each edge \((u, v) \in E_U \), we make it “live” (active) with probability \( p_{uv} \). Note that although an edge \((u, v)\) may appear in multiple graph instances, it is used for the diffusion process exactly once, at time \( t_{uv} \). Since, the probability \( p_{uv} \) is the same for all graph instances, we can assume that the decision to make the edge live is made in advance. We use \( E_U^t \subseteq E_U \) to denote the set of live edges.

We will now create a graph that unfolds the graph sequence and the diffusion process into a single graph. A similar construction is described in [15]. We refer to this graph as the expanded graph \( G_X = (V_X, E_X) \). The graph \( G_X \) consists of \( n + 1 \) layers of \( |V| \) nodes, where the edges of graph \( G^t \) are placed between the nodes of layer \( i - 1 \) and \( i \). Formally, let \( V_i \) denote the \( i \)-th layer of nodes, where \( i = 0, 1, \ldots, n \). For each node \( v \in V \) there is a corresponding node \( v^i \) in layer \( i \). For every (directed) edge \((u, v)\) in graph \( G^t \) we add an edge \((u^{i-1}, v^i)\) to the set of edges \( E_X^t \) if it is also one of the live edges in \( E_U^t \). Furthermore, we add a set of transition edges of the form \((v^{i-1}, v^i)\) for all \( v \in V \) and all layers \( i = 0, 1, \ldots, n \).

An example of our construction for the sequence of three graphs in Figure 4 is shown in Figure 5. The bold edges in the union graph \( G_U \) correspond to the live edges \( E_U^t \). The expanded graph \( G_X \) has four layers of nodes \( V_0^1, V_1^1, V^2, V^3 \). The dashed edges correspond to the transition edges of the graph \( G_X \), and the solid edges correspond to the live edges in \( E_U^t \). The shaded nodes show an example of the diffusion, when \( I = \{v_0^1\} \). As we will show below, the activated nodes are the ones reachable from node \( v_0^1 \) in graph \( G_X \).

\textbf{Theorem 1.} For all instances of the persistent EIC model on a graph sequence with fixed probabilities, the spread function \( \sigma_{EIC} \) is monotone and submodular.

\textbf{Proof.} Given a set of live edges, we will prove by induction that for a set of initiators \( I \), the set of active nodes at time-step \( t \) is the same as the set of nodes in \( V^t \) in graph \( G_X \) that are reachable from \( I \). The claim is trivially true for time-step \( t = 0 \). Assume that it is true at time \( t - 1 \). Consider now time-step \( t \). First note that thanks to the transition edges, any node \( u \) that is reachable at \( t - 1 \) will remain reachable at \( t \). If a node \( v \) becomes active at time \( t \)
then there must be an active node $u$ in $A^{t-1}$ that gets connected with $v$ for the first time since $u$ became active, and edge $(u, v)$ is live. Since $u$ is reachable, $v$ will also become reachable. If $v$ becomes reachable at time $t$ then this means that at time $t$ it became connected with a live edge with a reachable node $u$. Since $u$ is active, this means that $v$ will also become active.

Reachability defines a monotone and submodular function. Therefore, the expected spread $\sigma_{\text{pEIC}}$ can be written as a linear combination of monotone and submodular functions, and thus it is also monotone and submodular. □

In Figure 5 we can see the set of reachable nodes from the set $A^t = \{v_i^t\}$, and the time-step at which each node is activated. Note that reachability in the graph $G_X$ is different from reachability in the graph $G^t = (V, E^t)$ through live edges. In our example, in the $G^t$ graph, all nodes are reachable from $v_1$ through live edges. However, in the graph $G_X$ node $v_5$ never becomes reachable, since at the time that $v_2$ is activated the edge $(v_2, v_3)$ no longer appears in the graph.

From the proof and the discussion above, it is clear that the best time to activate a node $u$ in the pEIC model is at the beginning of the diffusion process, since this maximizes the chances of $u$ to meet other nodes in the future. We can prove by induction that the pEIC model on a graph sequence with fixed probabilities is timing-insensitive.

However, monotonicity and submodularity properties do not hold if the activation probabilities vary over time.

**Lemma 2.** The function $\sigma_{\text{pEIC}}$ is neither monotone nor submodular for arbitrary graph sequences.

**Proof.** The proof is similar to that for the $t$EIC model. We use the same graph sequence as in Figure 2 except for the fact that in graph $G_2$ we have an additional edge $(v_3, v_4)$ with activation probability $\varepsilon$. All other activation probabilities are 1. As before, if we activate node $v_3$ at time $t = 0$, we have spread $\sigma_{\text{pEIC}}(\{v_1^t\}) = N + 3$. If we add $v_5$ to the initiator set, the diffusion reaches node $v_5$ at time $t = 1$. As a result the first time that $v_3$ connects with $v_4$ is at time $t = 2$, where the activation probability of edge $(v_3, v_4)$ is $\varepsilon$. The expected spread in this case is $\sigma_{\text{pEIC}}(\{v_1^t, v_3^t\}) = \varepsilon(N + 1) + 4$. It follows that $\sigma_{\text{pEIC}}$ is not monotone. Using the same argument as in the previous proof we can show that it is also not submodular. □

The intuition behind this counter-example is similar to that for the $t$EIC model in Section 5.1.1. Since the activation probability of $(v_3, v_4)$ varies over time, it is important to time the activation of node $v_3$ appropriately, so that it gets activated when the edge activation probability is high. Otherwise, similar to before, the diffusion is blocked. The variation in the activation probabilities makes the pEIC model timing-sensitive.

### 5.2 Evolving LT Model

Given a static graph $G = (V, E)$, the LT model assumes that every edge $(u, v)$ in $E$ is associated with a weight $b_{uv}$, such that for any node $v \in V$, the weight of its incoming edges sums to a value less than 1. In diffusion under the LT model, each node has a threshold $\theta_v$ chosen uniformly at random in the interval $[0, 1]$. If $(u, v)$ is an incoming edge to $v$, and $u$ is active, we say that $(u, v)$ is live. Node $v$ is activated when the sum of weights over the live edges exceeds the threshold $\theta_v$.

In the case of an evolving graph, let $G_t = (V, E_t)$ denote the union graph defined in Section 2. Similar to the static case we assume that every edge $(u, v)$ in $E_t$ is associated with a weight $b_{uv}$ independent of the time that the edge appears in the graph sequence. We also assume that for any node $v \in V$, we have that $\sum_{u: (u, v) \in E_t} b_{uv} \leq 1$.

The threshold of a node is defined in the same way as for the static case.

Similar to the EIC model, we will consider two variations of the LT model for the case of evolving graphs that differ in the mechanism used for node activations. We will collectively refer to these models as the Evolving Linear Threshold model, and denote it by ELT.

#### 5.2.1 Transient ELT Model

In this model, the diffusion process is similar to that in the static graph, but the total incoming weight of live edges is computed only over the live edges that are all present in a single graph instance. This model captures again a transient diffusion process, where a node is affected only by the neighbors present in a graph instance, and their influence dies off when not present. We refer to this model as the Transient Evolving Linear Threshold model, and denote it by ELT.

Formally, for a node $v$, let $N_v^t$ denote the set of incoming neighbors of $v$ at time $t$. Recall that $A^t$ is the set of active nodes at time $t$, and let $N_v^t = N_v^t \cap A^t$ denote the active neighbors of $v$ at time $t$. Now, let $W_v^t$ denote the total weight incoming to node $v$ from live edges at time $t$. That is,

$$W_v^t = \sum_{u \in N_v^t} b_{uv}$$

A node $v$ becomes active at time $t$ if $W_v^t \geq \theta_v$.

We can prove by induction that $\sigma_{\text{ELT}}$ is monotone, and timing-insensitive (i.e., the best time to activate a node is at time $t = 0$). We omit the proofs due to space constraints. Below we prove that $\sigma_{\text{ELT}}$ is not submodular.

**Lemma 3.** The function $\sigma_{\text{ELT}}$ is not submodular.

**Proof.** For the proof we use a simple example shown in Figure 6. There are two snapshots in the graph sequence $G^1$ and $G^2$, shown in the middle and right pane respectively. The left pane shows the union graph $G_t$. We assume that $b_{v_1, v_3} = 1/3$, for all edges $(v_1, u)$. In this graph, it is clear that the only node that can be activated via the diffusion process is node $u$. Let $\Pr[u|z]$ denote the probability that $u$ is active at the end of diffusion, given a set of initiator nodes $I \subseteq \{v_1, v_2, v_3\}$. The expected diffusion spread is $\sigma_{\text{ELT}}(I) = |I| + \Pr[u|I]$. Consider now the case that
\[ I = \{v_1\}. \] Clearly, \( u \) can only be activated in snapshot \( G^t \), and this happens if \( \theta_u \leq 1/3 \). Therefore, \( \Pr[u|\{v_1\}] = \Pr[\theta_u \leq 1/3] = 1/3 \). Consider now the addition of node \( v_2 \) to the initiator set. Since the edges \( \{v_1, u\} \) and \( \{v_2, u\} \) do not appear in the same snapshot, we still need \( \theta_u \leq 1/3 \) in order for \( u \) to be activated. That is, \( \Pr[u|\{v_1, v_2\}] = 1/3 \). Consider now the initiator set \( I = \{v_1, v_2, v_3\} \). In this case, node \( u \) is activated if \( \theta_u \leq 2/3 \), since the total weight of the live edges in \( G^t \) is 2/3. \( \Pr[u|\{v_1, v_2, v_3\}] = 2/3 \). Therefore, \( \sigma_{\text{pELT}}(\{v_1, v_3\}) - \sigma_{\text{pELT}}(\{v_1\}) = 0 \) while \( \sigma_{\text{pELT}}(\{v_1, v_2, v_3\}) - \sigma_{\text{pELT}}(\{v_1, v_2\}) = 1/3 \). That is, the addition of node \( v_3 \) to the set \( \{v_1, v_2\} \) has a greater effect than the addition of \( v_3 \) to \( \{v_1\} \). Hence, \( \sigma_{\text{pELT}}(I) \) is not submodular.

5.2.2 Persistent ELT Model

In this model, we assume that influence persists over time. A node accumulates the influence of the active nodes it has met in the past. When the accumulated influence crosses the node’s threshold it becomes activated. This is a reasonable model to capture the scenario where a user is in a social network, who is interested in an item, collects opinions over time, and when the peer pressure exceeds her threshold, she makes the decision to adopt. We call this model Persistent Evolving Linear Threshold (pELT) model, and denote it by \( pELT \).

Formally, we define \( \text{CNA}^t_v = \cup_{t=1}^{t-1} \text{NA}^t_v \) to be the set of active neighbors of \( v \) at any time up to \( t \), and we use \( W_v^t \) to denote the total weight accumulated by the node \( v \) up to time \( t \). That is,

\[
W_v^t = \sum_{u \in \text{CNA}^t_v} b_{uv} \tag{2}
\]

A node \( v \) becomes active at time \( t \) if \( W_v^t \geq \theta_v \).

We will now show that for the persistent ELT model the spread function \( \sigma_{\text{pELT}} \) is monotone and submodular. The proof works by showing that the diffusion process is equivalent to reachability in the expanded graph \( G_X \) defined in Section 5.1.2. The set of live edges \( E_L^t \) in the case of the pELT model is defined in the same way as in [15]. Given the union graph \( G_U \), for every node \( v \) in \( V \) we randomly select a single edge \( (v, u) \in E_U \) with probability \( b_{uv} \). With probability \( 1 - \sum_{(v, u) \in E_U} b_{uv} \), no edge is selected. This selection is performed for each of the nodes in \( V \) to define the set of live edges \( E_L^t \). The diffusion then happens deterministically through the live edges on the graph \( G_X \). A node connected with a live edge to an active node gets immediately activated.

Figure 7 shows the union graph for the example graph sequence in Figure 4 and the selected live edges. Note that, different from the EIC model, each node has exactly one incoming live edge. The expanded graph \( G_X \) is shown in the right part of the figure. The shaded nodes show the diffusion, when \( I = \{v_1^t\} \). We can show that the activated nodes are the ones reachable from node \( v_1^t \) in graph \( G_X \). Note again that reachability in the graph \( G_X \) is different from reachability in the graph \( G_U^t = (V, E_U^t) \) through live edges.

**Theorem 2.** For all instances of the persistent ELT model the spread function \( \sigma_{\text{pELT}} \) is monotone and submodular.

**Proof.** The proof follows closely the one in [15], by showing by induction that the conditional distribution over the activated nodes at time \( t = n \) given a set of initiators \( I \)

\[ W_v^t = \sum_{u \in \text{CNA}^t_v} b_{uv} \]

is the same as the distribution over the reachable nodes at layer \( V^n \) from the set \( I \). We omit the details due to lack of space. Given that the reachability function is monotone and submodular, it follows that the function \( \sigma_{\text{pELT}}(I) \) can be expressed as a linear combination of monotone and submodular functions, and hence it is also monotone and submodular.

5.3 Summary of Model Properties

Table 1 summarizes our theoretical results regarding the properties of the diffusion spread function for the transient and persistent EIC and ELT models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Timing</th>
<th>Monotone</th>
<th>Submodular</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIC</td>
<td>sensitive</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>pEIC (fixed pr)</td>
<td>insensitive</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>pEIC (general)</td>
<td>insensitive</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>tELT</td>
<td>insensitive</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>pELT</td>
<td>insensitive</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

6. EXPERIMENTAL EVALUATION

In this section, our goal is to evaluate experimentally how the diffusion spread is affected by network evolution using real datasets. We first describe the algorithms we use for the influence maximization problem, the datasets, and the experimental setup. We then present the evaluation results.

6.1 Algorithms

The algorithm most commonly used for diffusion maximization in static networks is Greedy. Greedy takes as input a candidate set of nodes \( C \) and a value \( k \), and selects a set \( I \) of \( k \) nodes to be activated. It proceeds iteratively, where at each iteration it computes for each candidate node the marginal increase in the expected spread that results by adding the node to \( I \). This is estimated by performing a large number of Monte-Carlo simulations of the diffusion process and taking the average spread. The node that causes the maximum marginal increase is added to \( I \) and removed
from the candidate set $C$. The process continues until $k$ nodes are selected.

For submodular functions, the greedy algorithm provides a constant factor approximation guarantee. Despite the fact that, for evolving graphs, the spread function is not submodular for all models, the greedy algorithm is still a natural algorithm to consider.

To study the effect of timing of activation to the diffusion spread, we consider two variants of the Greedy algorithm by varying the set $C$ of candidate initiators. The more general case is to have $C = V_t = \{ v^t : v \in V, t = 0, ..., n-1 \}$, that is, to be able to activate nodes at any graph instance. A node is activated once at the selected time instance. We will refer to this greedy algorithm as Greedy-AT (Greedy-Any-Time). The other variant corresponds to the typical setting in diffusion maximization which assumes that all initiators must be activated at a single time-step, namely at the beginning of the diffusion or evolution process. In this case, $C = \{ v^0 : v \in V \}$ is the set of nodes $V$ at time $t = 0$. We will refer to this greedy algorithm as Greedy-OT (Greedy-One-Time). We note that the comparison between Greedy-AT and Greedy-OT is of interest only for the EIC models that are time-sensitive. The ELT models are timing-insensitive, so the optimal activation time is at time $t = 0$.

For selecting each of the $k$ initiators, Greedy runs for each of the $|C|$ candidates $R$ simulations of the diffusion process. Assuming that diffusion has complexity $D$, Greedy-AT runs in $O(kn|V|RD)$ and Greedy-OT in $O(k|V|RD)$ time, where $n$ is the number of graphs in the sequence.

### 6.2 Datasets and Experimental Setup

We consider datasets from three real evolving networks. For each dataset, nodes correspond to users, and edges to interactions between them. All edges have time-stamps within a time period $T$. We construct a graph sequence $G = \{ G^1, ..., G^n \}$ by breaking up the time period $T$ into $n$ intervals of equal length. The graph $G^t$ captures all user interactions within the $t$-th time interval. If more than one interaction occurs between two users within interval $t$, multiple edges are created between the corresponding nodes in the graph. We provide next a short description of our datasets.

The Hospital-Ward dataset\(^1\) contains the network of contacts between 46 health-care workers and 20 patients of a hospital ward for 4 days in December 2010. We construct a sequence of 16 graphs, where each graph represents a time-frame of 6 hours. The union graph $G_U$ of this dataset is very dense and includes a central node adjacent to about 80% of the nodes. All graphs in the sequence are sparse with fluctuating degrees, following the day and night-time habits of the hospital residents.

The DBLP-Authors dataset corresponds to the co-authorship graph of authors that have published papers in a major data mining, database or theory conferences between 2004 and 2013 downloaded from DBLP\(^2\). We include only authors that have published in at least three distinct years during this period resulting in 1,249 authors. We construct a graph sequence of 10 graphs, where each graph represents collaborations within a single year. All graphs are sparse and highly fragmented.

The Social-Evolution dataset\(^3\) reports meetings between college students in an undergraduate dormitory based on mobile phones usage. The probability of two users meeting at a specific time instant is estimated using bluetooth information and proximity to WiFi access points. Thus each edge $e$ is annotated with a time-stamp $T$ and a probability $q_e$. We select all interactions in the first week of October 2008 and create a sequence of 7 graphs, where each graph corresponds to one day. There are 48 nodes. We view this graph sequence as a typical example of the weekly pattern of interactions of a group of users. All graphs in the sequence are connected and relatively dense except from the first one which contains few nodes and edges.

When simulating the diffusion process for the EIC model, we set $p = 0.01$ for the propagation probabilities of the edges, except for the DBLP-Authors network which is extremely sparse, for which we use $p = 0.1$. Note that due to the variation in the multiplicity of edges, the activation probabilities vary over time, making the persistent EIC model timing-sensitive. For the ELT model, the weight $b_{uv}$ is set equal to the fraction of the edges incident on $v$ in the union graph that are between $u$ and $v$. Finally, $R$ is equal to 10,000.

### 6.3 Results

We address two fundamental issues: (1) How does the timing of the activation of the influencers affect the diffusion spread? (2) Does the evolution of the network affect the estimation of the spread?

\(^1\) http://www.sociopatterns.org
\(^2\) http://dblp.uni-trier.de/xml/
\(^3\) http://realitycommons.media.mit.edu/
Timing of activations: We first look into the importance of timing in activating an initiator. For this experiment, we only consider the EIC model which is time-sensitive. Figure 8 shows the spread of the different algorithms as a function of the number of initiators for the transient EIC model. The first observation is that Greedy-AT significantly outperforms Greedy-OT. To stress the importance of timing we also consider two weaker any-time algorithms: the GreedySort-AT algorithm runs a single iteration of Greedy-AT and returns the $k$ nodes with the highest spread; the Degree-AT algorithm selects as initiators the $k$ nodes with the highest degree at any graph instance. We also consider Random that outputs a random selection of initiators at any graph snapshot.

We observe that all any-time algorithms, even the simple heuristics, outperform the Greedy-OT algorithm. In Social-Evolution and Hospital-Ward, Greedy-OT performs close or worse than random. It is competitive only on the DBLP-Authors dataset. We now compare such estimates with those obtained on the full graph sequence. In particular, we want to study (a) how accurate is the estimation of spread on the union graph compared to that obtained on the graph sequence, and (b) how good are the initiators computed for the static case when used on the dynamic graph.

Evolution-agnostic vs evolution-aware diffusion: Although most real-life networks evolve over time, most existing work on diffusion views the network as a static object and estimates the spread of influence and the set of initiators on a static graph, more specifically on the union graph $G_U$. We now compare such estimates with those obtained on the full graph sequence. In particular, we want to study (a) how accurate is the estimation of spread on the union graph compared to that obtained on the graph sequence, and (b) how good are the initiators computed for the static case when used on the dynamic graph.

For this experiment, we run the Greedy algorithm on the union graph $G_U$ and select a set $I$ of $k$ initiators. We use Static-U to denote the spread obtained by using the selected set $I$ on the union graph $G_U$ (recall that the union graph is the graph with edge set the union of all edges in all snapshots). This is an optimistic estimation of the actual influence spread, assuming that all edges are present at all times. We then use the set $I$ on the graph sequence $G$ and compute the spread under the evolving diffusion model. For the EIC model, we use Static-OT to denote the algorithm that activates the nodes in $I$ at time $t = 0$, and Static-AT to denote the algorithm that activates each node in $I$ at the best time instant $t$, so that it (individually) achieves maximum spread. For the ELT model the best activation time is always $t = 0$, so we use Static to denote this algorithm.

Figures 9 and 10 show the results of the above algorithms for the transient EIC and ELT models respectively and Figures 11 and 12 for the persistent EIC and ELT models respectively. Note that in the pEIC case the curves for Static-OT and Static-AT, and Greedy-OT and Greedy-AT are almost identical. As we discussed before, the pEIC model is essentially timing-insensitive in our experiments.
A first observation is that the diffusion spread is severely over-estimated when using the union graph. Static-U achieves spread that is an order of magnitude higher than that of Greedy-AT. This is especially pronounced in the DBLP-Authors dataset which is highly fragmented. For any practical application that wants to make decisions based on the size of spread, using the static graph will yield a very poor estimate of the true diffusion on the evolving graph.

A second observation is that, when using the initiator set obtained for the static graph on the evolving graph, performance is poor, especially for the EIC model. Static-OT is clearly the worst algorithm indicating that the static union graph is a poor indicator of how the diffusion will progress on the evolving graph. Note though that the results can be significantly improved for the EIC model by adding some amount of time information. Static-AT achieves competitive performance, being the second best algorithm for the Social-Evolution dataset. Therefore, although the static graph provides some signal about which nodes are good initiators, it is important to time appropriately the activation of these initiators for the signal to be of any use. These differences are less evident in the ELT model. In this case, the initiators on the static graph perform well on the evolving graph sequence. This is reasonable given the activation mechanism of the model.

Another observation is with regards to the differences between persistent and transient diffusion models. In general, as expected, the actual spread of influence under the persistent models is larger than that of the corresponding transient variant, and slightly closer to the estimation obtained on the static graph.

7. DISCUSSION AND CONCLUSIONS

In this paper, we studied the problem of influence maximization on dynamic networks, where the network evolves while the diffusion process is in progress. We proposed the Evolving Independent Cascade (EIC) and the Evolving Linear Threshold (ELT) diffusion models, and studied them theoretically and experimentally. Our work reveals that there are key differences between diffusion in static and evolving graphs, both in theory and in practice, and that it is wrong to ignore the dynamic nature of the network. Our evolving models that incorporate the importance of timing in diffusion result in a fundamentally different diffusion process.

We note that in our problem definition we assume that the entire graph sequence is known in advance. It would be interesting to study diffusion on dynamic graphs that evolve following specific patterns, for example weekly ones [6]. Furthermore, instead of estimating the diffusion spread using the actual graph sequence, one could provide approximate estimations, e.g., based on studies of how real graphs evolve over time and corresponding graph generation models, such as those in [17]. Another approach would be to design algorithms that have only partial information about the future, e.g., only a subset of the future edges, or a window of the m next graphs. It would be interesting to understand and quantify the tradeoff between the amount of information available and the success of the initiator selection. Finally, it would be interesting to understand and analyze online algorithms for the problem.

Another possible direction for future work is to study in more detail the relationship between diffusion time and evolution time. Dynamic processes on dynamic graphs have been studied in the past (e.g., see [11] for random walks on...
dynamic graphs) and it would be interesting to investigate if such mathematical tools could be applied to the diffusion problem. Finally, we note that the greedy algorithm needs to run a large number of simulations for all candidate nodes to estimate the spread, making it computationally expensive. Recently, sketching algorithms have been proposed for the influence maximization problem in static graphs \[7\]. It would be interesting to consider such algorithms for the case of evolving graphs.

**Acknowledgments**

This work is supported by the Marie Curie Reintegration Grant project titled JMUGCS which has received research funding from the European Union.

8. **REFERENCES**


