Overexposure-aware influence maximization

Article *in* ACM Transactions on Internet Technology · June 2020

CITATIONS
0

READS
37

3 authors:

Grigorios Loukides
King's College London
83 PUBLICATIONS **1,127** CITATIONS
SEE PROFILE

Robert Gwadera
Cardiff University
32 PUBLICATIONS **473** CITATIONS
SEE PROFILE

Shing-Wan Chang
Middlesex University, UK
8 PUBLICATIONS **164** CITATIONS
SEE PROFILE

Overexposure-aware influence maximization

GRIGORIOS LOUKIDES, King's College London, United Kingdom
ROBERT GWADERA, Cardiff University, United Kingdom
SHING-WAN CHANG, Middlesex Univesity, United Kingdom

Viral marketing campaigns are often negatively affected by overexposure. Overexposure occurs when users become less likely to favor a promoted product, after receiving information about the product from too large a fraction of their friends. Yet, existing influence diffusion models do not take overexposure into account, effectively overestimating the number of users who favor the product and diffuse information about it. In this work, we propose the first influence diffusion model that captures overexposure. In our model, LAICO (Latency Aware Independent Cascade Model with Overexposure), the activation probability of a node representing a user is multiplied (discounted) by an overexposure score, which is calculated based on the ratio between the estimated and the maximum possible number of attempts performed to activate the node. We also study the influence maximization problem under LAICO. Since the spread function in LAICO is non-submodular, algorithms for submodular maximization are not appropriate to address the problem. Therefore, we develop an approximation algorithm which exploits monotone submodular upper and lower bound functions of spread, and a heuristic which aims to maximize a proxy function of spread iteratively. Our experiments show the effectiveness and efficiency of our algorithms.

CCS Concepts: • **Information systems → Social networks**;

Additional Key Words and Phrases: influence maximization, social networks, influence diffusion

ACM Reference Format:
Grigorios Loukides, Robert Gwadera, and Shing-Wan Chang. 2020. Overexposure-aware influence maximization. 1, 1 (April 2020), 30 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Viral marketing campaigns are being performed over social networks, as a cost-effective and efficient means of advertising a product to a large number of users. In these campaigns, a small set of users diffuse information about the promoted product, with the aim of influencing (*activating*) many other users (i.e., making these users favor the product) through word-of-mouth effects. The influence diffusion process can be modeled by an influence diffusion model.

However, existing models (e.g., [9, 13, 20, 25, 33]) do not consider the impact of *overexposure* on the influence diffusion process. Overexposure occurs when a viral marketing campaign becomes too aggressive (i.e., too large a fraction of a user's friends attempt to activate the user, making the user bored or annoyed and less willing to diffuse information) [3, 18]. For example, consider a user, Alice, whose friends tell her about a new fashion product. The probability that Alice favors the product and tells others about it increases when more friends tell Alice about the product, but up to

Authors' addresses: Grigorios Loukides, King's College London, London, United Kingdom, grigorios.loukides@kcl.ac.uk; Robert Gwadera, Cardiff University, Cardiff, United Kingdom, GwaderaR@cardiff.ac.uk; Shing-Wan Chang, Middlesex Univesity, London, United Kingdom, S.Chang@mdx.ac.uk.

XXXX-XXXX/2020/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

a certain point. When too large a fraction of Alice's friends tell her about the product, Alice believes that the product is no longer fashionable and, consequently, she becomes less likely to favor the product and tell her friends about it [3]. For example, users become less likely to buy a DVD or book after receiving "too" many recommendations from their friends through email, because they perceive such recommendations as spam [23]. Similarly, players of social games become less willing to join a game, after receiving requests from too large a fraction of their friends [18]. Overexposure has also had a substantial negative impact on the growth of large-scale systems, including LinkedIn [1] and the Plaxo contact manager [19].

Taking into account the impact of overexposure makes the influence diffusion models more realistic and spread estimation more accurate. This is, in turn, crucial for addressing influence maximization [24], a fundamental problem in viral marketing. In this problem, a social network is typically modeled as a graph, whose nodes and edges correspond to users and their connections, respectively, and the goal is to find a set of at most k graph nodes (*seeds*) which would lead to the largest expected number of activated nodes in the graph (*spread*), according to an influence diffusion model.

Our work makes the following contributions:

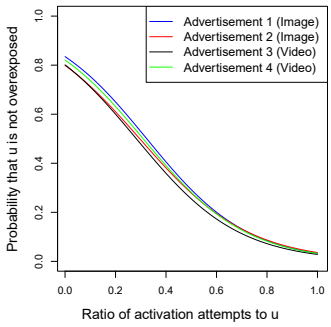
1. LAICO model We incorporate overexposure into the well-known Latency Aware Independent Cascade (LAIC) [25] influence diffusion model. This leads to a modified model, called LAICO (O for Overexposure). The difference between LAIC and LAICO is that in LAICO the activation probability of each node is multiplied (discounted) by the *overexposure score* of the node. We chose the LAIC model as the basis for LAICO because it is time-aware, which is useful in practical applications, and because it generalizes other models (e.g., [9, 20]).

The overexposure score of a node u represents the probability that u is *not* overexposed and acts as a discount factor to the activation probability of u. For example, when u has a small overexposure score, it is more likely to be overexposed. Thus, its activation probability is multiplied with a small number and is heavily reduced. The overexposure score of u is computed by: (I) calculating the ratio between the estimated and maximum possible number of attempts to activate u, and (II) applying a logistic function that gets as input the ratio and outputs the overexposure score of u.

For the calculation of the ratio, we note that in LAICO each in-neighbor of a node u attempts to activate u with some probability. Thus, the number of attempts to activate u cannot be computed exactly and has to be estimated. In fact, Monte Carlo simulation can be used to derive a good estimate. Yet, this strategy is inefficient because it needs a very large number of simulations [12]. Therefore, we propose a more efficient strategy that leads to similar estimates, as shown in our experiments. Our strategy assumes that each in-neighbor of u is not activated by other in-neighbors. This assumption allows modeling the probability that u receives a certain number of activation attempts as a Poisson binomial [40] distribution and using the mean of the distribution (i.e., the expected number of attempts performed to activate u) as an estimate of the number of attempts performed to activate u. Then, the ratio is calculated by normalizing (dividing) the estimate with the maximum possible number of attempts to activate u (i.e., the number of in-neighbors of u). The normalization captures findings in marketing [6] and economics [3], suggesting that users with more friends are accustomed to being exposed to information from more people, or they do not pay attention to all the information they receive.

The logistic function that outputs the overexposure score of u is derived based on a user evaluation study. The study is performed on a sample of users targeted by the viral marketing campaign as follows: The users are shown viral content about the promoted product (e.g., an image or video) and asked about the minimum ratio of activation attempts that would discourage them to diffuse information about the product. Then, based on the user responses, a logistic regression function,

which models the perception of a user's overexposure as a function of the ratio of attempts to activate them, is derived.



(a)

Advertisement	Logistic coefficients	
	β_0	β_1
1	1.61977	−5.00491
2	1.38144	−4.66672
3	1.39301	−4.931301
4	1.52203	−4.91809

(b)

Advertisement	McFadden pseudo-R^2	Nagelkerke pseudo-R^2
1	0.253	0.385
2	0.228	0.350
3	0.244	0.369
4	0.246	0.374

(c)

Fig. 1. (a) Logistic function, for each promoted product in our user evaluation study. (b) Coefficients β_0 and β_1 in the logistic function $\frac{1}{1+e^{-(\beta_0+\beta_1 \cdot x)}}$ for each promoted product in our user evaluation study, where x denotes the minimum ratio of activation attempts. (c) Pseudo-R^2 statistics indicating goodness-of-fit for each logistic function [27].

Example 1. *In Fig. 1a, we plot the logistic functions obtained from the responses of* 476 *social network users, who we asked about four different products. The smallest ratio that users could input was* $\frac{1}{100}$. *In Fig. 1b, we provide, for each function, the logistic regression coefficients, which depend on the product, and in Fig. 1c the values of two pseudo-*R^2 *statistics, which confirm that the function is good fit [27]. Each logistic function is used to compute the probability that a node is not overexposed (i.e., its overexposure score), based on the ratio of activation attempts received by the node when information about a product is diffused. For example, when information about Advertisement 2 is diffused, a node with ratio* $\frac{1}{100}$ *in Fig. 1a is not overexposed with probability about* 0.8. *This probability is computed by giving as input the ratio* $\frac{1}{100}$ *to the red logistic regression curve in Fig. 1a.*

2. Algorithms for the *IML* problem The problem of Influence Maximization in the LAICO model (*IML*) is fundamentally different from that in the LAIC model, because the spread function in LAICO is non-monotone and non-submodular, as we show. Thus, the *Greedy* algorithm [32] does not achieve the approximation ratio $(1 - \frac{1}{e}) \approx \frac{63}{100}$ that it achieves for monotone submodular functions. Therefore, we identify additional properties of the spread function in LAICO and, based on these properties, we design an approximation algorithm, called *ASA*, and an efficient heuristic, called *HSS*, to address the *IML* problem.

ASA is founded on the following property: **(P1)** The spread function in LAICO is lower-bounded by a monotone, submodular function and upper-bounded by another monotone, submodular function. The lower-bound (respectively, upper-bound) function assigns to each node the minimum (respectively, maximum) overexposure score. We show that *ASA* achieves an approximation ratio $\mathcal{M} \cdot (1 - \frac{1}{e})$, where $\mathcal{M} \leq 1$ is a factor that depends on how close are the bound functions to the spread function. This result is due to **P1**, which allows exploiting the *Sandwich Approximation* strategy [28]. We also show that *ASA* constructs a seed-set with spread at least $c \cdot k$ times larger than that of applying *Greedy* [32] with the spread function of the LAICO model, where c is a constant that depends on the logistic regression function used in the overexposure score computation.

HSS is founded on the following two properties: **(P2)** There is a proxy function for the spread function in LAICO, which is constructed by utilizing the logarithm transformation and an upper bound for submodular functions [17]. **(P3)** The proxy function can be approximately maximized (in expectation), to produce a high-quality solution to the *IML* problem based on the effective and efficient *Sup-Sub* procedure [17].

Our experiments show that the LAIC model overestimates spread, because it does not take overexposure into account. Specifically, we show that the spread in LAIC is much higher than that in LAICO, because many nodes have lower activation probabilities in the LAICO model, due to overexposure. The experiments also show that our strategy for estimating the number of attempts to activate a node is as effective as the Monte Carlo simulation based method but at least two orders of magnitude more efficient. Furthermore, we experimentally show that *ASA* finds a solution within $0.88 \cdot (1 - \frac{1}{e}) \approx 55\%$ of the optimal, while *HSS* is less effective but more efficient and scalable with respect to the size of seed-set and the size of the time window.

We note that our methodology for modeling overexposure can be directly used to incorporate overexposure into the classical Independent Cascade (IC) model [20] and the Independent Cascade with Meeting points (IC-M) model [9], which are special cases of the LAIC model. We also note that the modeling of the probability that a node u is not overexposed as a function of the ratio of activation attempts to u through a logistic regression function is of independent interest. For example, such functions can be used to compare different viral marketing campaigns with respect to how much they fatigue users, to predict whether certain nodes would become highly overexposed, and to calculate statistics of practical interest, such as the expected number of overexposed nodes for a campaign.

Paper organization The rest of the paper is organized as follows. Section 2 introduces some preliminary concepts and the LAIC model. Section 3 explains how we estimate the number of attempts to activate a node and how we assign overexposure scores. Section 4 introduces our LAICO model and the *IML* problem we aim to solve. Section 5 and 6 introduces our *ASA* algorithm and the *HSS* heuristic, respectively. In Section 7, we experimentally evaluate our approach. Section 8 discusses related work, and Section 9 concludes the paper.

2 BACKGROUND

2.1 Preliminaries

Let $G(V, E)$ be a directed graph, where V is a set of nodes and E is a set of edges. The set of in-neighbors of a node u is denoted with $n^-(u)$ and has size $|n^-(u)|$, which is referred to as the *in-degree* of u. The set of out-neighbors of u is denoted with $n^+(u)$ and has size $|n^+(u)|$, which is referred to as the *out-degree* of u.

Let U be a universe of elements and 2^U be its power set. A function $f : 2^U \rightarrow \mathbb{R}$ is *monotone* (also called *non-decreasing*), if $f(X) \leq f(Y)$ for all subsets $X \subseteq Y \subseteq U$, and *non-monotone* otherwise. A function $f : 2^U \rightarrow \mathbb{R}$ is *non-negative* if $f(X) \geq 0$, for each subset $X \subseteq U$, and *submodular* if it satisfies the *diminishing returns* property $f(X \cup \{u\}) - f(X) \geq f(Y \cup \{u\}) - f(Y)$, for all $X \subseteq Y \subseteq U$ and any $u \in U \setminus Y$ [22]. If the diminishing returns property holds with equality, then f is modular. A function f is *supermodular* if and only if $-f$ is submodular [22]. A modular function f is both submodular and supermodular.

The *modular upper bound* [17] of a submodular function $g(X) : 2^U \rightarrow \mathbb{R}$ is a modular function

$$\widehat{g_Y}(X) = g(Y) + \sum_{u \in X \setminus Y} (g(\{u\}) - g(\{\})) - \sum_{u \in Y \setminus X} (g(Y) - g(Y \setminus u))$$

where the subset $Y \subseteq U$ is the *parameter* of $\widehat{g_Y}(X)$ and $\{\}$ denotes the empty set.

2.2 Latency Aware Independent Cascade (LAIC) model [25]

A popular influence diffusion model is the Independent Cascade (IC) model [14, 20]. In the IC model, a node is either active or inactive, and a node that becomes active cannot become inactive. The influence propagates at discrete time steps as follows: Each seed is active at time $t = 0$ and attempts to activate each of its out-neighbors at $t = 0$ once. If multiple seeds have the same out-neighbor, they all try to activate it in arbitrary order. At $t = 1$, some out-neighbors of the seeds may become active. These out-neighbors attempt to activate their own inactive out-neighbors at $t = 1$, some of the latter nodes may become active at $t = 2$, and the process proceeds similarly until no new node becomes active.

The LAIC model extends the IC model by accounting for the fact that the nodes may be activated with delays. That is, in the LAIC model a node u' that is activated at time t and activates its out-neighbor u will not necessarily activate u at time $t + 1$, as in the IC model, but at time $t + 1 + i$, where $i \in [0, \delta]$ is a delay at most equal to a maximum delay threshold δ. In the LAIC model, each edge (u', u) is associated with a probability vector $m((u', u)) = [m_0((u', u)), \ldots, m_\delta((u', u))]$, whose element $m_i((u', u))$ denotes the probability that u' activates u with delay i. The probability vectors of edges are set based on the target population of the campaign [25]. Thus, it is easy to see that when each delay i is equal to 0, the LAIC model becomes equivalent to the IC model.

In the LAIC model, the probability that a seed-set S activates a node u at time j is given by $P_{LAIC}(u, S, j)$. Similarly, the probability that S activates u within a time window $[0, t]$ is given by $\mathcal{P}_{LAIC}(u, S, [0, t]) = \sum_{j \in [0,t]} P_{LAIC}(u, S, j)$. The spread of S in $[0, t]$ is computed as in Eq. 1:

$$\sigma_{LAIC}(S, t) = \sum_{u \in V} \mathcal{P}_{LAIC}(u, S, [0, t]) \tag{1}$$

The spread can be computed by a dynamic programming algorithm [16] which considers all simple paths that start from S and activate a node with probability at least equal to a minimum path probability threshold h. This algorithm is much more efficient than the alternative Monte Carlo simulation based method [25].

In the remaining of the paper, we use the subscript $LAIC$ for the spread and activation probabilities in the LAIC model and no subscript for the spread and activation probabilities in our LAICO model. For example, $P_{LAIC}(u, S, j)$ denotes the probability that a seed-set S activates a node u at a time point j in the LAIC model, and $P(u, S, j)$ denotes the probability that S activates u at a time point j in the LAICO model.

3 MODELING OVEREXPOSURE

3.1 Estimating the number of attempts to activate a node

The number of attempts performed to activate a node u in a time window $[0, t]$ can be estimated based on Monte Carlo simulation. For example, consider the Monte Carlo simulation algorithm in [25], which computes the number of activated nodes for a seed-set S and window $[0, t]$ on an appropriately constructed random subgraph of $G(V, E)$ in time $O(|V| + |E|)$. By executing the algorithm on R different random subgraphs, we can estimate the spread $\sigma_{LAIC}(S, t)$ as the average number of activated nodes over all these subgraphs. The algorithm of [25] can be easily modified to count the number of activated in-neighbors of any node u in a random subgraph, which is equal to the number of attempts performed to activate u in $[0, t]$ by its in-neighbors in the random subgraph. The modified algorithm still needs $O(|V| + |E|)$ time. Then, by executing the modified algorithm on R different random subgraphs, we can estimate the number of attempts performed to activate u in $[0, t]$, as the average number of attempts over all R random subgraphs. However, the modified algorithm is inefficient, since R needs to be large to derive good estimates ($R = 20000$ is suggested in [25]). Thus, in the following, we present a more efficient method that is also fairly

accurate, as shown in our experiments. The method estimates the number of attempts performed to activate a node u in $[0, t]$ by directly using the activation probabilities of in-neighbors of u, which are available when a dynamic programming algorithm [16] is used to compute the spread. The benefit of our method is efficiency, since the dynamic programming algorithm is much faster than Monte Carlo simulation [16].

Our method requires knowing the probability distribution of the number of attempts performed to activate u in a time window $[0, t]$ (i.e., the probability that u receives r activation attempts, for each possible r). Then, it uses the mean of the distribution (i.e., the expected number of attempts in $[0, t]$) as an estimate of the true number of attempts. For simplicity of presentation, we first consider the case in which all activation attempts occur at time point $j = 0$ (i.e., in the window $[0, 0]$). Then, we consider the general case in which the activation attempts occur at any time point j in the window $[0, t]$.

Activation attempts at a time point $j = 0$ Let u_l be an in-neighbor of u that was activated at a time point $j = 0$ with probability $P(u_l, S, j)$. As it will be explained later, in LAICO u_l tries to activate u once, independently of the other in-neighbors that were activated at j. Therefore, each activation attempt by an in-neighbor u_l, $l \in [1, |n^-(u)|]$, is an independent Bernoulli trial with a potentially different success probability $P(u_l, S, j)$, and the number of activation attempts to u at $j = 0$ is a sum of independent Bernoulli trials. Consequently, the probability that u receives a certain number of activation attempts from its in-neighbors at $j = 0$ is given by the Poisson binomial [40] distribution with parameters $P(u_1, S, j), \ldots, P(u_{|n^-(u)|}, S, j)$.

Activation attempts at a time point j in window $[0, t]$ The difference from the case in which $j = 0$ is that the event "an in-neighbor u_l of u was activated at a time point j in $[0, t]$" depends on the event "an in-neighbor u_m of u was activated at a time point $j' < j$", when there are paths from S to u_l that pass through u_m. Similarly, the event "u_l was activated in the window $[0, t]$" may depend on the event "u_m was activated in the window $[0, t]$". Since there may be multiple paths from S that pass through more than one in-neighbors of u, the computation of the probability that u receives r activation attempts in $[0, t]$ is hard. However, the dependencies among these events do not substantially affect the number of attempts to activate u, as shown in our experiments. Thus, we make the following independence assumption: *the events "the in-neighbors of u were activated in $[0, t]$" are mutually independent.*

The assumption allows modeling the activation attempt by the in-neighbor u_l of u as an independent Bernoulli trial with success probability $\mathcal{P}(u_l, S, [0, t])$, and modeling the number of activation attempts to u in $[0, t]$ as a sum of independent Bernoulli trials. Consequently, we can use the Poisson binomial distribution with parameters $\mathcal{P}(u_1, S, [0, t]), \ldots, \mathcal{P}(u_{|n^-(u)|}, S, [0, t])$ to estimate the probability that u receives r activation attempts from its in-neighbors in $[0, t]$. This probability is computed as in Eq. 2:

$$P_{poibin}(Y = r) = \sum_{A \subseteq n^-(u): |A| = r} \prod_{u_l \in A} \mathcal{P}(u_l, S, [0, t]) \prod_{u'_l \in n^-(u) \setminus A} \left(1 - \mathcal{P}(u'_l, S, [0, t])\right) \quad (2)$$

where Y is a random variable measuring the number of attempts to activate the node u, P_{poibin} is the probability mass function (pmf) of the Poisson binomial distribution, A is a subset of r in-neighbors of u, and $n^-(u)$ is the set of in-neighbors of u.

Then, the mean of the distribution P_{poibin} represents the expected number of attempts to activate u in a window $[0, t]$, and it can be used as an estimate of the number of attempts to activate u in $[0, t]$. Our estimate is inspired by the use of the expected number of activated nodes in $[0, t]$ as an estimate of the true number of activated nodes in $[0, t]$ [25]. The expected number of attempts to activate u in $[0, t]$ is computed as $\sum_{u_l \in n^-(u)} \mathcal{P}(u_l, S, [0, t])$.

The computation results in an estimate that is close to that of Monte Carlo simulation, as shown in our experiments. At the same time, the computation is also much more efficient than Monte Carlo simulation. This is because it is based on the probabilities $\mathcal{P}(u_l, S, [0,t])$, which are available from the computation of spread in the LAICO model, instead of Eq. 2, which can be expensive as it considers $O(\binom{|n^-(u)|}{r})$ subsets of $n^-(u)$ of size r. The estimate of the number of attempts to activate u in a window $[0,t]$, when the seed-set is S, is denoted with $N_u^{S,t}$ and computed as illustrated in Example 2.

Example 2. *Fig. 2a shows the activation probabilities of the in-neighbors u_1, u_2, and u_3 of a node u, when the seed-set is S and the window is $[0,t]$. Fig. 2b shows the probability mass function (pmf) of the Poisson binomial with parameters $\{0.6, 0.7, 0.5\}$. The mean of the pmf in Fig. 2b is equal to $0.6 + 0.7 + 0.5 = 1.8$ and corresponds to the estimated number of activation attempts $N_{S,t}^u$.*

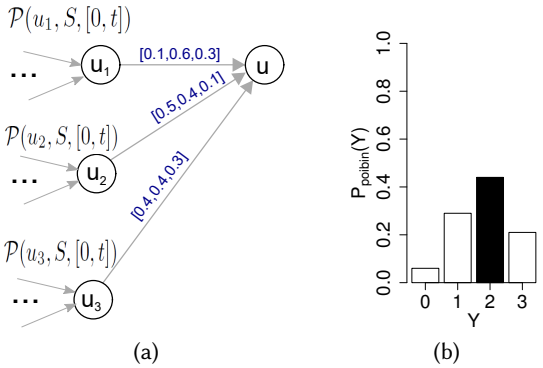


Fig. 2. (a) The in-neighbors of u and their activation probabilities $\mathcal{P}(u_1, S, [0,t]) = 0.6$, $\mathcal{P}(u_2, S, [0,t]) = 0.7$, and $\mathcal{P}(u_3, S, [0,t]) = 0.5$, as well as the probability vector of each edge from an in-neighbor to u (shown as an edge weight). (b) The pmf of the Poisson binomial with parameters $\{0.6, 0.7, 0.5\}$.

3.2 Assigning overexposure scores

The overexposure score of a node u, for a seed-set S and window $[0,t]$, is:

$$\mathbf{O}(u, S, t) = \begin{cases} R\left(\tilde{N}_u^{S,t}\right), & \text{if } N_u^{S,t} > 1 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where $\tilde{N}_u^{S,t} = \frac{N_u^{S,t}}{|n^-(u)|}$ is the ratio of the estimated number of activation attempts received by u to the number of all possible activation attempts received by u (note that the maximum value of Y is $|n^-(u)|$), and $R(\tilde{N}_u^{S,t}) = \frac{1}{1+e^{-\left(\beta_0+\beta_1 \cdot \tilde{N}_u^{S,t}\right)}}$ is the logistic function with coefficients β_0 and β_1. The coefficients are specified by the party performing the viral marketing campaign, based on a user evaluation study, as explained in Introduction. A node u with $N_u^{S,t} \leq 1$ is considered not overexposed (i.e., it has $\mathbf{O}(u, S, t) = 1$), because the node is unlikely to receive more than one activation attempts and overexposure occurs when a user receives the diffused information at least twice. For example, the overexposure score of each seed is 1. Clearly, it is straightforward to use a threshold larger than 1, to specify nodes that are not considered overexposed, if desired.

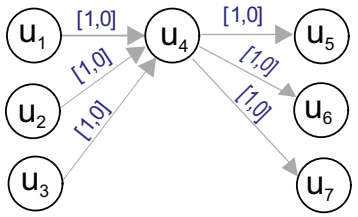
4 LAICO MODEL AND PROBLEM DEFINITION

In our LAICO model, the activation probability $\mathcal{P}_{LAIC}(u, S, [0, t])$ of node u is multiplied (discounted) by the overexposure score $\mathbf{O}(u, S, t)$, where S is a seed-set and $[0, t]$ a window. That is, the activation probability in LAICO is defined as $\mathcal{P}(u, S, [0, t]) = \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t)$. The spread in LAICO is defined as $\sigma(S, t) = \sum_{u \in V} \mathcal{P}(u, S, [0, t])$, and the probability that a node u is activated at a time point $j \in [0, t]$ is defined as $P(u, S, j) = \mathcal{P}(u, S, j) - \mathcal{P}(u, S, j - 1)$.

The *IML* problem Given a graph $G(V, E)$, a window $[0, t]$, and a parameter k, the influence maximization problem in the LAICO model (*IML*) seeks to find a node subset $S \subseteq V$ with size $|S| \leq k$ and maximum spread $\sigma(S, t)$.

The function $\sigma(S, t)$ for a window $[0, t]$ is non-monotone, and it is neither submodular nor supermodular, as illustrated in Example 3.

Example 3. *Consider the graph of Fig. 3a and the window* $[0, 2]$. *Also, consider that the overexposure score of each node u, for seed-set S and window* $[0, 2]$, *was computed using Eq. 3 with the logistic function* $\frac{1}{1+e^{-\left(\beta_0+\beta_1 \cdot \tilde{N}_u^{S,t}\right)}}$ *whose coefficients* $\beta_0 = 1.61977$ *and* $\beta_1 = -5.00491$ *were obtained by a user evaluation study. The spread of different seed-sets in the window is shown in Fig. 3b, together with the ratio* $\tilde{N}_{u_4}^{S,2}$ *and overexposure score* $\mathbf{O}(u_4, S, 2)$ *of* u_4, *when activated by one of the seed-sets S. The function* σ *is non-monotone, since, for* $\{u_1\} \subseteq \{u_1, u_2\}$, *it holds that* $\sigma(\{u_1\}, 2) = 5 > \sigma(\{u_1, u_2\}, 2) = 2.6091$. *The function* σ *is* not *submodular, because for* $\{u_1, u_2\} \subseteq \{u_1, u_2, u_3\} \subseteq \{u_1, \ldots, u_7\}$ *and* $u_4 \in \{u_1, \ldots, u_7\} \setminus \{u_1, u_2, u_3\}$, *it holds that* $\sigma(\{u_1, u_2\} \cup \{u_4\}, 2) - \sigma(\{u_1, u_2\}, 2) = 3.3909 < \sigma(\{u_1, u_2, u_3\} \cup \{u_4\}, 2) - \sigma(\{u_1, u_2, u_3\}, 2) = 3.869$. *In addition,* σ *is* not *supermodular, because for* $\{\,\} \subseteq \{u_1\}$ *and* $u_2 \in \{u_1, \ldots u_7\} \setminus \{u_1\}$, *it holds that* $\sigma(\{u_2\}, 2) - \sigma(\{\,\}, 2) = 5 > \sigma(\{u_1\} \cup \{u_2\}, 2) - \sigma(\{u_1\}, 2) = -2.3909$.



(a)

Seed-set S	$\sigma(S, 2)$	$\tilde{N}_{u_4}^{S,2}$	$\mathbf{O}(u_4, S, 2)$
$\{u_1\}$	5	1/3	1
$\{u_2\}$	5	1/3	1
$\{u_1, u_2\}$	2.6091	2/3	0.1523
$\{u_1, u_2, u_3\}$	3.131	3/3	0.0328
$\{u_1, u_2, u_4\}$	6	2/3	1
$\{u_1, u_2, u_3, u_4\}$	7	3/3	1

(b)

Fig. 3. (a) Example graph. The probability vector for each edge is shown as edge label. (b) The spread of different seed-sets in the window $[0, 2]$, as well as the estimated ratio $\tilde{N}_{u_4}^{S,2}$ and overexposure score $\mathbf{O}(u_4, S, 2)$ of the node u_4, when it is activated by a seed-set S of these seed-sets.

Since $\sigma(S, t)$ is non-submodular, *IML* cannot be approximated by algorithms for influence maximization in the LAIC model [16, 25], nor by adapting approximation algorithms [8, 30] for submodular maximization. Since $\sigma(S, t)$ is non-supermodular, *IML* cannot be approximated with algorithms for supermodular maximization (e.g., [38]).

5 *ASA* (APPROXIMATION ALGORITHM FOR *IML* BASED ON *SA*)

This section presents our *ASA* approximation algorithm for finding a seed-set S with size at most k and large spread $\sigma(S, t)$. *ASA* exploits the fact that the spread function $\sigma(S, t)$ satisfies the following property:

P1: $\sigma(S,t)$ is lower-bounded by a monotone, submodular function $\sigma_{\mathcal{L}}$ and upper-bounded by another monotone, submodular function $\sigma_{\mathcal{U}}$.

ASA executes *Greedy* three times, one with $\sigma(S,t)$, another with the function $\sigma_{\mathcal{L}}(S,t)$, and a third with the function $\sigma_{\mathcal{U}}(S,t)$, and then returns the solution with the maximum $\sigma(S,t)$ among the solutions obtained by the three executions of *Greedy*. Thus, *ASA* is an adaptation of the *SA* strategy (see Appendix A).

In the following, we define the functions $\sigma_{\mathcal{L}}(S,t)$ and $\sigma_{\mathcal{U}}(S,t)$ and present their properties:

$$\sigma_{\mathcal{L}}(S,t) = \sum_{u \in V_1} \mathcal{P}_{LAIC}(u,S,[0,t]) + \sum_{u \in V_{<1}} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(1)) \tag{4}$$

$$\sigma_{\mathcal{U}}(S,t) = \sum_{u \in V_1} \mathcal{P}_{LAIC}(u,S,[0,t]) + \sum_{u \in V_{<1}} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(0)) \tag{5}$$

where $V_1 = \{u \in V \mid \mathbf{O}(u,S,t) = 1\}$, $V_{<1} = \{u \in V \mid \mathbf{O}(u,S,t) < 1\}$, and $R(1)$ (resp., $R(0)$) is the minimum (resp., maximum) overexposure score of a node in $V_{<1}$. That is, if the overexposure score $\mathbf{O}(u,S,t)$ of a node u in $\sigma(S,t) = \sum_{u \in V}[\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot \mathbf{O}(u,S,t)]$ is lower than 1, it is replaced by the minimum overexposure score $R(1)$ in $\sigma_{\mathcal{L}}(S,t)$, and it is replaced by the maximum overexposure score $R(0)$ in $\sigma_{\mathcal{U}}(S,t)$. Consequently, $\sigma_{\mathcal{L}}(S,t)$ is a lower bound of $\sigma(S,t)$ and $\sigma_{\mathcal{U}}(S,t)$ is an upper bound of $\sigma(S,t)$. Note that the functions $\sigma_{\mathcal{L}}(S,t)$ and $\sigma_{\mathcal{U}}(S,t)$ are monotone submodular because $\mathcal{P}_{LAIC}(u,S,[0,t])$ is monotone submodular [25] and $R(0)$ as well as $R(1)$ are constants. Thus, Property **P1** holds.

We now discuss the effectiveness of *ASA* in terms of finding a solution with good spread $\sigma(S,t)$. First, we prove that *ASA* offers the approximation guarantee in Eq. 6:

$$\frac{\sigma(S_{ASA},t)}{\sigma(S^*_{IML},t)} \geq \max\left\{\frac{\sigma(S_{\mathcal{U}},t)}{\sigma_{\mathcal{U}}(S_{\mathcal{U}},t)}, \frac{\sigma_{\mathcal{L}}(S^*_{IML},t)}{\sigma(S^*_{IML},t)}\right\} \cdot \left(1 - \frac{1}{\mathrm{e}}\right) \tag{6}$$

where S_{ASA} (resp., $S_{\mathcal{U}}$) is the seed-set returned by *ASA* (resp. *Greedy* applied with $\sigma_{\mathcal{U}}$) and S^*_{IML} is the optimal solution to the *IML* problem. The proof of Eq. 6 easily follows from the use of the *SA* strategy [28] and is omitted. *ASA* is effective (in the worst case captured by Eq. 6), when each term in the max factor of Eq. 6 is close to 1. Note that the second term in the max factor is not computable in polynomial time because it involves the optimal solution S^*_{IML}. Yet, a slightly worse bound, where only the first term $\mathcal{M} = \frac{\sigma(S_{\mathcal{U}},t)}{\sigma_{\mathcal{U}}(S_{\mathcal{U}},t)}$ of the max factor is kept, can be computed. In our experiments, we show that $\mathcal{M}$ is larger than 0.88, implying that *ASA* is very effective.

We also prove, in Theorem 1 below, that *ASA* can be much more effective than applying *Greedy* with the spread function σ, especially when k is large. We refer to this variant of *Greedy* as Gr_{LAICO}. Specifically, Theorem 1 shows that the ratio between the spread of the seed-set constructed by *ASA* and that of the seed-set constructed by Gr_{LAICO} is at least 1, increases linearly with k and depends on the ratio between $R(0)$ and $R(1)$.

Theorem 1 (Ratio between spread of *ASA* and Gr_{LAICO}). *Let S_{ASA} and S_{Gr} be the seed-set constructed by ASA and by Gr_{LAICO}, respectively, when applied to the same graph and with the same k. It holds that*

$$\frac{\sigma(S_{ASA},t)}{\sigma(S_{Gr},t)} \geq \left(\frac{R(0)}{R(1)} - \frac{R(1)}{R(0)}\right) \cdot k + \frac{R(1)}{R(0)},$$

where $R(1)$ and $R(0)$ is the minimum and the maximum value of the logistic regression function R, respectively.

Proof. See Appendix B □

ASA performs $O(|V| \cdot k)$ function evaluations, since it evaluates each of the functions $\sigma_{\mathcal{L}}$, $\sigma_{\mathcal{U}}$, and σ that are executed in the algorithm $O(|V| \cdot k)$ times [28]. Each evaluation takes $O(|V|^2 \cdot t)$ time when computed based on [16] but the bound is pessimistic [16].

6 HSS (HEURISTIC FOR *IML* BASED ON *SUP-SUB*)

This section presents our *HSS* heuristic for finding a seed-set S with size at most k and large spread $\sigma(S,t)$. *HSS* exploits the fact that $\sigma(S,t)$ can be transformed to a function $\widehat{\sigma}(S,t)$ that is a difference of two submodular functions. The latter function is used to construct the proxy function of $\sigma(S,t)$ which is approximately maximized by *HSS*.

We first discuss the function $\widehat{\sigma}(S,t)$ and its properties in Section 6.1, and then discuss the proxy function constructed from $\widehat{\sigma}(S,t)$ in Section 6.2. After that, we discuss the operation of the *HSS* heuristic in Section 6.3.

6.1 The $\widehat{\sigma}(S,t)$ function and its properties

The function $\widehat{\sigma}(S,t)$ is defined in Eq. 7 below:

$$\widehat{\sigma}(S,t) = \sum_{u \in V} \ln\left(\mathcal{P}_{LAIC}(u,S,[0,t])\right) - \sum_{u \in V} \left(-\ln\left(\mathbf{O}(u,S,t)\right)\right) \tag{7}$$

and obtained by applying the logarithmic transformation to each non-zero term in square brackets in the spread function $\sigma(S,t) = \sum_{u \in V} \mathcal{P}(u,S,[0,t]) = \sum_{u \in V} \left[\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot \mathbf{O}(u,S,t)\right]$ [1].

We now show that $\widehat{\sigma}(S,t)$ is a difference of two submodular functions. For this, we provide Theorem 2 whose proof makes use of Lemma 1.

Lemma 1. *The function* $\mathbf{O}(u,S,t)$ *is non-increasing supermodular.*

Proof. See Appendix C. □

Theorem 2 ($\widehat{\sigma}(S,t)$ is a difference of submodular functions). *In function* $\widehat{\sigma}(S,t)$*, the terms* $\sum_{u \in V} \ln\left(\mathcal{P}_{LAIC}(u,S,[0,t])\right)$ *and* $\sum_{u \in V} \left(-\ln\left(\mathbf{O}(u,S,t)\right)\right)$ *are submodular functions.*

Proof. See Appendix D □

6.2 The proxy function of $\sigma(S,t)$

The function $\widehat{\sigma}(S,t)$ is difficult to approximately maximize directly [17], due to the lemma below.

Lemma 2. *The function* $\widehat{\sigma}(S,t)$ *for a window* $[0,t]$ *is neither submodular nor supermodular.*

Proof. See Appendix E. □

However, it can still be used to derive another function $\widehat{\sigma_Y}(S,t)$ that is non-negative, non-monotone submodular and thus can be approximately maximized.

The function $\widehat{\sigma_Y}(S,t)$ is defined in Eq. 8:

$$\widehat{\sigma_Y}(S,t) = \sum_{u \in V} \left(\ln(\mathcal{P}_{LAIC}(u,S,t))\right) - \widehat{\mathbf{O}_Y}(S,t) - |V|\left(\ln(h) + (k+1)\ln(R(1))\right) \tag{8}$$

where

[1] Terms with $\mathcal{P}_{LAIC}(u,S,[0,t]) = 0$ do not affect $\sigma(S,t)$ and are ignored.

$$\widehat{\mathbf{O}_Y}(S,t) = \sum_{u\in V}(-\ln(\mathbf{O}(u,Y,t))) + \sum_{u'\in S\setminus Y}\sum_{u\in V}(-\ln(\mathbf{O}(u,\{u'\},t))) - \sum_{u'\in Y\setminus S}\sum_{u\in V}(-\ln(\mathbf{O}(u,Y,t)) + \ln(\mathbf{O}(u,Y\setminus\{u'\},t))). \quad (9)$$

is the modular upper bound of $\sum_{u\in V}(-\ln(\mathbf{O}(u,S,t)))$ with parameter $Y \subseteq V$, h is the minimum path probability threshold, and $R(1)$ is the maximum value of the logistic function R. Note that to construct $\widehat{\sigma_Y}(S,t)$, we simply replaced the overexposure term $\sum_{u\in V}(-\ln(\mathbf{O}(u,S,t)))$ with its modular upper bound and added the term $|V|\,(\ln(h) + (k+1)\ln(R(1)))$, which is equal to the minimum value of $\sum_{u\in V}(\ln(\mathcal{P}_{LAIC}(u,S,t))) - \widehat{\mathbf{O}_Y}(S,t)$, as proved in Appendix F.

We show in Theorem 3 that $\widehat{\sigma_Y}(S,t)$ is non-negative, non-monotone submodular.

Theorem 3 (Properties of $\widehat{\sigma_Y}(S,t)$). *The function $\widehat{\sigma_Y}(S,t)$ for a subset $Y \subseteq V$ and window $[0,t]$ is:* (I) *non-negative,* (II) *non-monotone, and* (III) *submodular.*

Proof. See Appendix F. □

This implies that $\widehat{\sigma_Y}(S,t)$ can be approximately maximized by *SubSample Greedy* [30] (an overview of *SubSample Greedy* can be found in Appendix G).

6.3 Operation of *HSS*.

HSS uses, as a proxy function of the spread function $\sigma(S,t)$, the function $\widehat{\sigma_Y}(S,t)$ with a suitable seed-set as parameter Y. Furthermore, instead of applying *SubSample Greedy* once with the proxy function, *HSS* performs an iterative procedure similar to the *Sup-Sub* procedure [17] (see Appendix H). Specifically, in each iteration, *HSS* applies *Subsample Greedy* with a proxy function that has a different seed-set as parameter Y. This iterative procedure allows obtaining a solution of larger spread.

As can be seen from the pseudocode, *HSS* works iteratively. In each iteration i, it uses a proxy function with a parameter that is the seed-set S^{i-1} constructed in iteration $i-1$, with $S_0 = \{\}$. That is, in iteration i, *SubSample Greedy* is applied with the proxy function $\widehat{\sigma_{S^{i-1}}}(S^i,t)$ and finds a seed-set S^i. An iteration is performed as long as the relative improvement of the seed-set (with respect to the proxy function) after an iteration is at least equal to a threshold $\phi \geq 0$, specified by the party that performs the viral marketing campaign (step 7). After the loop in step 7 terminates, *HSS* returns the best seed-set found over all iterations in terms of spread (steps 8 and 9). This is needed because a seed-set with higher value in the proxy function may not necessarily have a larger value of spread.

We now provide two observations, $O1$ and $O2$, to justify the effectiveness of *HSS*:

($O1$) *In every iteration, HSS approximately maximizes the proxy function in expectation*: This is because, due to the use of *Subsample Greedy*, the expected value $\mathbb{E}[\widehat{\sigma_{S^{i-1}}}(S^i,t)]$, for the seed-set S^i obtained in iteration $i > 0$, is within $\frac{1}{\mathrm{e}}\cdot(1-\frac{1}{\mathrm{e}})$ of the optimal solution $\arg\max_{S'\subseteq V,|S'|\leq k}\widehat{\sigma_{S^{i-1}}}(S',t)$.

($O2$) *An approximately maximum value in the proxy function implies large spread*: This is because *HSS* approximately maximizes the proxy function $\widehat{\sigma_{S^{i-1}}}(S^i,t)$ (in expectation), which results in large $\sigma(S^i,t)$ according to Lemma 3 below, since the term in square brackets in the inequality of the lemma is independent of S^i and

$$\sum_{u\in V}\sum_{n=2}^{\infty}\frac{(1-\mathcal{P}_{LAIC}(u,S,[0,t])\cdot\mathbf{O}(u,S,t))^n}{n}$$

is small (see Section 7).

Algorithm: HSS (Heuristic for *IML* based on *Sup-Sub*)
Input: Set of nodes V of G, parameter k, window $[0, t]$, minimum allowable relative improvement threshold ϕ
Output: Subset $S^i \subseteq V$ of size $|S^i| \leq k$

1 $i \leftarrow 0$ // Iteration counter
2 $S^0 \leftarrow \varnothing$ // Initialize bound parameter
3 $S^1 \leftarrow V$ // Initialize seed-set in iteration 1
4 **do**
5 $\quad i \leftarrow i + 1$
6 $\quad S^i \leftarrow$ Apply *Subsample Greedy* to approximately solve $\arg\max_{S \subseteq V, |S| \leq k} \{\widehat{\sigma_{S^{i-1}}}(S, t)\}$
7 **while** $\frac{\widehat{\sigma_{S^{i-1}}}(S^i,t) - \widehat{\sigma_{S^{i-2}}}(S^{i-1},t)}{\widehat{\sigma_{S^{i-2}}}(S^{i-1},t)} \geq \phi$
8 $S \leftarrow$ seed-set $S^j \in \{S^0, \ldots, S^i\}$ with maximum $\sigma(S^j, t)$
9 **return** S

Lemma 3. *For any seed-set S^i and window $[0, t]$, it holds that*

$$\sigma(S^i, t) \geq \widehat{\sigma_{S^{i-1}}}(S^i, t) + [|V|(1 + \ln(h) + (k+1) \cdot \ln(R(1)))] + \sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t))^n}{n},$$

where n is an integer and $|V|$ is the number of nodes in the graph.

Proof. See Appendix I. □

HSS performs an extra iteration only when the value of a seed-set in the proxy function becomes sufficiently larger than the value of the seed-set constructed in the previous iteration (see step 7). The goal is to construct a seed-set with larger value in the proxy function and hence larger spread, due to observation $O2$. The stopping criterion guarantees that *HSS* terminates (since an iteration is not performed when S^i has a lower value in the proxy function than S^{i-1}) [17].

HSS performs $O(I \cdot |V|)$ proxy function evaluations, where I is the number of iterations (in our experiments, I was at most 4), since *Subsample Greedy* performs $O(|V|)$ function evaluations [30]. Each evaluation takes $O(|V|^2 \cdot t)$ time when computed based on [16] but the bound is pessimistic [16].

7 EXPERIMENTAL EVALUATION

In this section, we first demonstrate that our approach can efficiently produce a good estimate of the number of attempts performed to activate a node u and of its overexposure score. Then, we show that, unlike our LAICO model, the LAIC model overestimates spread. After that, we evaluate the effectiveness and efficiency of our *HSS* and *ASA* methods.

7.1 Baselines

Since no existing algorithms can deal with the *IML* problem (see Section 8), we compared our methods against three baselines: Gr_{LAIC}, *Deg*, and Gr_{LAICO}. Gr_{LAIC} is the *Greedy* algorithm with the spread function $\sigma_{LAIC}(S, t)$ which outperforms the methods and baseline heuristics in [25] in terms of maximizing spread, as shown in [25]. *Deg* [20] constructs a seed-set comprised of the k nodes with the largest out-degrees in the graph. Gr_{LAICO} is the *Greedy* algorithm with the non-submodular spread function $\sigma(S, t)$ (see Section 5). The *Lazy Greedy* (a.k.a CELF) [20, 29] optimization was used in Gr_{LAIC}, and in the part of *ASA* where *Greedy* is applied with the monotone

submodular bound functions $\sigma_{\mathcal{L}}$ and $\sigma_{\mathcal{U}}$. Other optimizations are possible for this part too [24]. However, the bottleneck of *ASA* is the part where *Greedy* is applied with the spread function σ. Due to the non-submodularity of this function, this part cannot be optimized with *Lazy Greedy* or other methods based on *Reverse Influence Sets* [39]. The results for *HSS* are averages over 10 runs.

7.2 Datasets and experimental setup

All algorithms were implemented in C++ and applied to the real datasets in Table 1, which were used in [9, 16, 25]. *POL* is available at http://www-personal.umich.edu/~mejn/ and all other datasets at http://snap.stanford.edu/data. The probability vector of each edge (u', u) was set to $\mathcal{P}_{u'} \cdot \frac{1}{|n^-(u)|}$, where $\mathcal{P}_{u'}$ is a Poisson distribution with a random mean $\lambda \in [1, 20]$ as in [16, 25]. The default values for k, window size $|W|$, maximum delay threshold δ, minimum path probability threshold h, and threshold ϕ were set to 5, 10, 10, $5 \cdot 10^{-3}$, and 10^{-2}, respectively. The activation probabilities of nodes were calculated using the (exact) dynamic programming method of [16].

Dataset	$\|V\|$	$\|E\|$	avg in-degree	max in-degree
Pol	1490	19090	11.9	305
WI	7115	103689	13.7	452
PH	34546	421578	24.3	846
EPIN	75879	508837	13.4	3079

Table 1. Characteristics of real datasets.

In addition, we used a synthetic dataset, referred to as *SYN*, to evaluate the accuracy of estimating the number of attempts $N_u^{S,t}$ to activate a node u. The *SYN* dataset was comprised of 10000 graphs. Our objective was to create graphs exhibiting very strong dependencies among in-neighbors, to test the independence assumption made by our approach. That is, to see how good is the estimate $N_u^{S,t}$ when an in-neighbor of u may be activated by other in-neighbors. In the *SYN* dataset, each graph has the following three layers: (I) the *seed* layer, comprised of nodes that are selected as seeds, (II) the *in-neighbors* layer, comprised of the out-neighbors of seeds, and (III) the *destination node* layer, comprised of a node u that is the out-neighbor of the nodes in the in-neighbors layer. Also, each graph has all possible edges: (I) from the seed layer to the in-neighbor layer, (II) between nodes in the in-neighbor layer, and (III) from the in-neighbor layer to the destination layer. The edge probability of an edge (u', u) was set to $(\frac{1}{|n^-(u)|})^\ell$, where ℓ was a randomly selected integer in $[1, L]$ for each edge between in-neighbors of the destination node and $\ell = 1$ for each other edge. The objective of the parameter L is to simulate longer paths between in-neighbors, leading to weaker dependencies among in-neighbors (and smaller activation probabilities for in-neighbors and for u). Unless stated otherwise, the seed layer and the in-neighbors layer is comprised of 10 nodes each, and the parameter L was set to 3.

We quantified the accuracy of estimating $N_u^{S,t}$, for a given graph in the *SYN* dataset, by computing the Relative Error $RE_{est} = \frac{|M_u^{S,t} - N_u^{S,t}|}{M_u^{S,t}}\%$, where $M_u^{S,t}$ is the number of activation attempts to the node u, computed by Monte Carlo simulation. The simulation is based on the method of [25] but instead of the activation probability node u it records the number of activated in-neighbors of u (see Section 3). Each simulation is repeated 20000 times following [25], and the average number of activated in-neighbors number is used as $M_u^{S,t}$. Clearly, a small RE_{est} implies that $N_u^{S,t}$ is similar to the estimate $M_u^{S,t}$ and, and hence the ratio $\tilde{N}_u^{S,t} = \frac{N_u^{S,t}}{|n^-(u)|}$ is similar to $\frac{M_u^{S,t}}{|n^-(u)|}$. We report the average RE_{est} score over each graph in *SYN*, referred to as ARE_{est} (Average Relative Error). Clearly, ARE_{est} quantifies the overall accuracy of our method in terms of estimating the number of activated in-neighbors.

We also quantified the accuracy of estimating the overexposure score $\mathbf{O}(u, S, t)$ of a node u for a seed-set S and window $[0, t]$, by computing the Relative Error $RE_{ov}(u, S, t) = \frac{|\mathbf{O}_M(u,S,t) - \mathbf{O}(u,S,t)|}{\mathbf{O}_M(u,S,t)}$, where $\mathbf{O}_M(u, S, t)$ is the overexposure score computed by Monte Carlo simulation. That is, $\mathbf{O}_M(u, S, t)$ differs from $\mathbf{O}(u, S, t)$ in that the former uses the estimate $M_u^{S,t}$ instead of $N_u^{S,t}$ in the computation of the logistic function $R()$. Clearly, a small $RE_{ov}(u, S, t)$ implies that $\mathbf{O}(u, S, t)$ is similar to the estimate $\mathbf{O}_M(u, S, t)$ obtained by the Monte Carlo simulation method. We evaluated RE_{ov} using the real datasets and the logistic function obtained from our user evaluation study (to be discussed later). Specifically, for a given node u and window $[0, t]$, we selected as seed-set S a random subset of nodes which are at distance t from u (i.e., there is at least one path from a node in S to u that has length t). Clearly, each node u may be activated by a node in S within $[0, t]$. The reason we selected the nodes at distance t is to generate large activation graphs. Thus, for a node u, seed-set S and window t, we get a score $RE_{ov}(u, S, t)$. Then, we repeat the process for different nodes and report the median RE_{ov}. This is because the distribution of RE_{ov} was skewed, due to the variability of the settings we considered, and thus the median provided a more robust estimate than the average. The size $|S|$ of S, the node u, and the window size $|W| = t$ were parameters in our evaluation. Unless stated otherwise, $|S|$ was set to 15, $|W|$ was set to 3, and we selected each node u with $|n^-(u)| \in [5, 100]$. The default parameters were chosen so that the Monte Carlo simulation method could terminate in reasonable time. Note, we did not plug in the Monte Carlo method directly into our algorithms for influence maximization, because it was too computationally inefficient for that use in the tested settings (i.e., about four orders of magnitude slower than our method).

We derived logistic regression functions based on our user study (see Introduction) that was conducted on Amazon Mechanical Turk. Our sample was comprised of 476 users, who were given a questionnaire with two images and two videos from recent, viral marketing campaigns of popular brands (see Appendix J). Table 4 summarizes some important statistics about our sample.

Gender	Percentage
Male	54.4%
Female	45.6%

(a)

Primary Social network	Percentage
Facebook	67.0%
Instagram	13.7%
Twitter	12.0%
Other	7.3%

(b)

Number of friends in Primary Social network	Percentage
[2, 100)	29.4%
[100 − 250)	26.2%
[250 − 500)	20.1%
[500, 10000)	22.5%

(c)

Fig. 4. Summary statistics for our sample: (a) Percentage of male and female respondents. (b) Social network that is used most often. (c) Number of friends in the social network that is used most often.

After analyzing the results of our user study, we obtained the four logistic functions whose coefficients are shown in Fig. 1b. Each logistic function corresponds to a different advertisement. To derive a logistic function for a product advertisement, we asked each user in our sample: "What percentage of friends telling you to watch this advertisement makes you feel discouraged to buy the product? For example if you have 100 friends in Facebook and you choose 20%, it means that you become discouraged to buy the product when 20 out of 100 of your friends post this picture on Facebook.". Then, we constructed a set of 101 records $\{r_1, \ldots, r_{101}\}$, for each user. Each record r_i, $i \in [1, 101]$, is a tuple whose first attribute is equal to $(i - 1)/100$ and second attribute is N, if $(i - 1)/100$ is less than the percentage selected by the user, and Y, otherwise. For example, if a user selected 2%, their records will be $\{(0/100, \mathsf{N}), (1/100, \mathsf{N}), (2/100, \mathsf{Y}), \ldots, (100/100, \mathsf{Y})\}$. Clearly, N indicates that the user does not feel overexposed, whereas Y indicates that they feel overexposed.

Next, we derived the logistic function for the product by using the R command $glm()$ [2] on the resultant dataset. The functions were validated using 10-fold validation and the McFadden and the Nagelkerke pseudo-R^2 statistics [27]. As mentioned in Introduction, the logistic function models the perception of a user's overexposure as a function of the ratio of attempts to activate the user, and then it is used to assign an overexposure score to any user (given a seed-set and window). However, there are other possibilities, which we leave to future work. For example, it is possible to assign an overexposure score to a user while taking into account the user's profile. To do this, one could first partition users into groups (e.g., based on their demographics and/or online behavior [34]), build a different logistic function for each group, and then assign an overexposure score to a user using the logistic function that was obtained for the user's group.

We present results for the logistic function with coefficients $\beta_0 = 1.61977$ and $\beta_1 = -5.00491$, which corresponds to Advertisement 1. The results with other logistic functions were similar (omitted), since the logistic functions are quite similar (see Fig. 1a).

All experiments ran on an Intel i7 at 2.8GHz with 16GB RAM.

7.3 Accuracy and efficiency of estimating the number of activation attempts $N_u^{S,t}$

We examined the impact of parameters $|n^-(u)|$ (number of nodes in the in-neighbors layer of the synthetic graphs), $|W|$ (window size), and L (exponent in the edge probability formula for edges in the in-neighbor layer of the synthetic graphs) to the accuracy and efficiency of obtaining $N_u^{S,t}$. Recall that the default values for $|n^-(u)|$, $|W|$, and L are 10, 10, and 3, respectively.

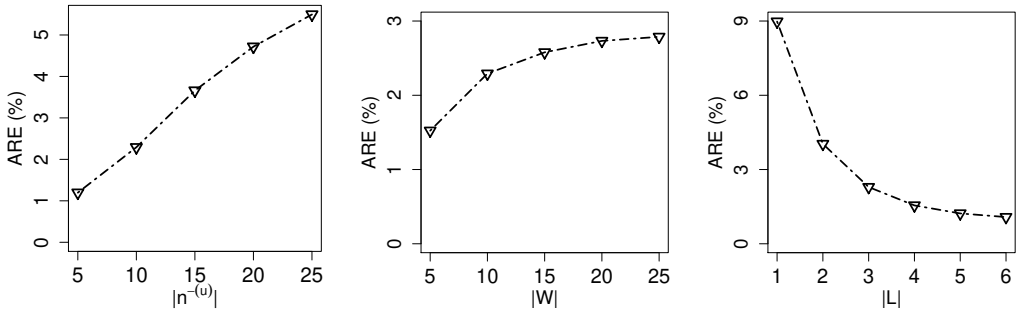


Fig. 5. $RE_{est} = \frac{|M_u^{S,t} - N_u^{S,t}|}{M_u^{S,t}}$ for a seed-set S of size 10 and for varying (a) $|n^-u|$, (b) $|W|$, and (c) L.

Fig. 5a shows the ARE_{est} scores for our method with varying number of in-neighbors $|n^-(u)|$ of u (i.e., number of nodes in the in-neighbors layer of the graphs of the SYN dataset). The ARE_{est} increases with $|n^-(u)|$. This is because there are stronger dependencies between the in-neighbors of v when $|n^-(u)|$ is large. The reason is that, in the graph of the SYN dataset, all edges between in-neighbors are present and thus a larger $|n^-(u)|$ leads to larger activation probabilities of in-neighbors. Nevertheless, the ARE_{est} is very low (average 3.5% and up to 5.5%) even when $|n^-(u)| = 25$ (i.e., when there are 25 in-neighbors of v, each connected to all seeds). Fig. 5b shows the ARE_{est} scores for our method with varying $|W|$. ARE_{est} increases with $|W|$. This is because a larger window size leads to stronger dependencies between in-neighbors, since the activation probabilities of in-neighbors get larger with the window size. Yet, ARE_{est} is again low (average 2.3%

[2] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html

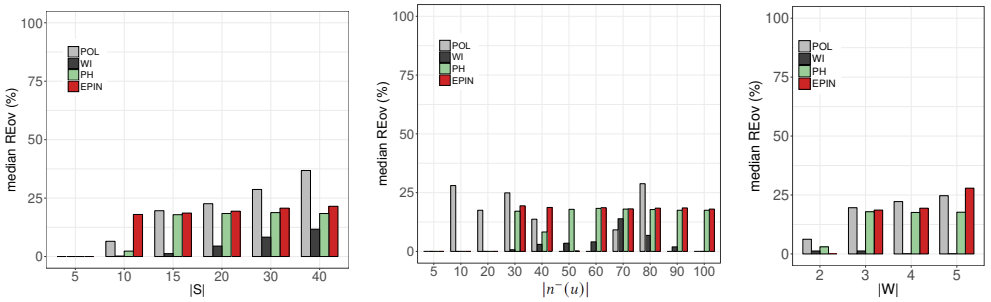
and up to 2.8%). Fig. 5c shows that the ARE_{est} scores for our method decrease as L increases. This is because a larger L leads to weaker dependencies between in-neighbors (and smaller activation probabilities for in-neighbors). Again, ARE_{est} was low (average 3.2% and up to 9.0%) in all tested cases.

Regarding efficiency, we note that our method was at least *two orders of magnitude faster* than the Monte Carlo simulation method (see Section 3.1). This is because our method avoids the need for a large number of costly simulations, being able to directly use the activation probabilities that are available during spread computation using the dynamic programming method of [16]. Specifically, our method was over 2281, 3184, and 3122 times faster on average in the experiments of Figs. 5a, 5b, and 5c, respectively.

7.4 Accuracy and Efficiency of estimating the overexposure score $\mathbf{O}(u, S, t)$

In this section, we show that our Poisson-binomial based estimation method can be used to obtain similar overexposure scores to those obtained by the Monte Carlo simulation method in several orders of magnitude less time.

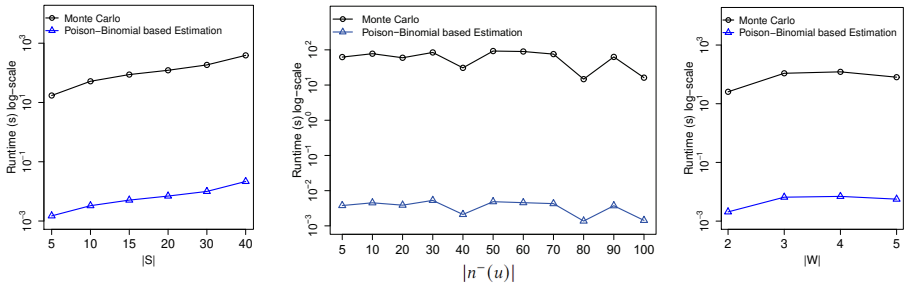
Fig. 6a shows the median RE_{ov} scores for our method with varying number of seeds $|S|$. The average score over the values of $|S|$ was 19.4%, 4.32%, 12.6%, and 16.4% for the *Pol*, *WI*, *PH*, and *EPIN* dataset, respectively. The median RE_{ov} score was 0 for $|S| = 5$ (i.e., our method on average derived the same overexposure score as that of the Monte Carlo simulation method) and increased with $|S|$ by a level that depends on the dataset (very small for the largest dataset *EPIN* and larger for the smallest dataset *Pol*). The difference between the datasets is attributed to their structure (*Pol* is denser than *EPIN*). Fig. 6b shows the median RE_{ov} scores for our method with varying number of in-neighbors $|n^-(u)|$. The average score over the values of $|n^-(u)|$ was 15.3%, 19.5%, 12.0%, and 11.8% for the *Pol*, *WI*, *PH*, and *EPIN* dataset, respectively. Fig. 6c shows the median RE_{ov} scores for our method with varying window size $|W|$. The average score over the values of $|W|$ was 18.2%, 0.63%, 14.1%, and 16.5% for the *Pol*, *WI*, *PH*, and *EPIN* dataset, respectively. These results show that our method performed consistently close to the Monte Carlo simulation method in terms of estimating overexposure scores.



(a) All datasets (b) All datasets (c) All datasets

Fig. 6. $RE_{ov}(u, S, t) = \frac{|\mathbf{O}_M(u,S,t) - \mathbf{O}(u,S,t)|}{\mathbf{O}_M(u,S,t)}$ for varying (a) $|S|$, (b) $|n^=(u)|$, and (c) $|W|$.

Figs. 7a, 7b, and 7c show that our estimation method is four orders of magnitude more efficient than the Monte Carlo simulation method, for varying $|S|$, $|n^-(u)|$, and $|W|$, respectively. They also show that both methods need more time as $|S|$ or $|W|$ increases, since the activation graphs for the nodes get larger (i.e., there are more paths from S to u)



(a) EPIN (b) EPIN (c) EPIN

Fig. 7. Time needed for computing $\mathbf{O}_M(u, S, t)$ by the Monte Carlo simulation method and $\mathbf{O}(u, S, t)$ by our Poisson-binomial based estimation method, averaged over all nodes vs (a) $|S|$, (b) $|n^-(u)|$, and (c) $|W|$, for the EPIN dataset.

Overall, the results of Figs. 6 and 7 suggest that, in fact, our estimation method can be used to obtain similar overexposure scores to those of the Monte Carlo estimation method, while being more efficient by orders of magnitude.

7.5 Spread overestimation in the LAIC model

Fig. 8a shows that the spread in the LAIC model for all algorithms was on average 42% and up to 124% higher than the spread in the LAICO model. This suggests that LAIC overestimates spread, since several nodes have lower activation probabilities in the LAICO model due to overexposure.

7.6 Effectiveness

We demonstrate that *ASA* and *HSS* find solutions with large spread for different parameters. Figs. 8b to 8e show that *ASA* outperforms Gr_{LAICO} for large k values, which is consistent with Theorem 1. They also show that *HSS* performs well, outperforming *Deg* and Gr_{LAIC} by 84% and 73% on average, respectively. *HSS* was worse by 15% on average compared to *ASA*, which finds a solution within $\mathcal{M} \cdot (1 - \frac{1}{e}) \approx 55\%$ of the optimal, where $\mathcal{M} > 0.88$ (see Fig. 9a).

HSS performs well because the proxy function $\widehat{\sigma_{S^{i-1}}}(S, t)$ is approximately equal to $\sigma(S, t) - |V|$, since the Taylor/Maclaurin remainder is small (see Lemma 3). Specifically, the normalized Taylor/MacLaurin remainder

$$\frac{-\sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1-\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot \mathbf{O}(u,S,t))^n}{n}}{\widehat{\sigma_{S^{i-1}}}(S, t)}$$

is less than 0.08, as shown in Fig. 9b. Fig. 9c shows that *ASA* and *HSS* outperform *Deg* and Gr_{LAIC} for different window size values, by at least 74% on average. The spread increases with the window size, because more paths are explored. Fig. 9d shows that *ASA* and *HSS* outperform *Deg* and Gr_{LAIC} for different mean rate λ, by at least 84% on average. The spread decreases with λ because the edge probability vectors have smaller values.

7.7 Efficiency

We demonstrate the runtime of *ASA* and *HSS* with respect to different parameters. *Deg* and Gr_{LAIC} were much faster than *ASA*, while the runtime of Gr_{LAICO} was very similar to that of *ASA*, so the

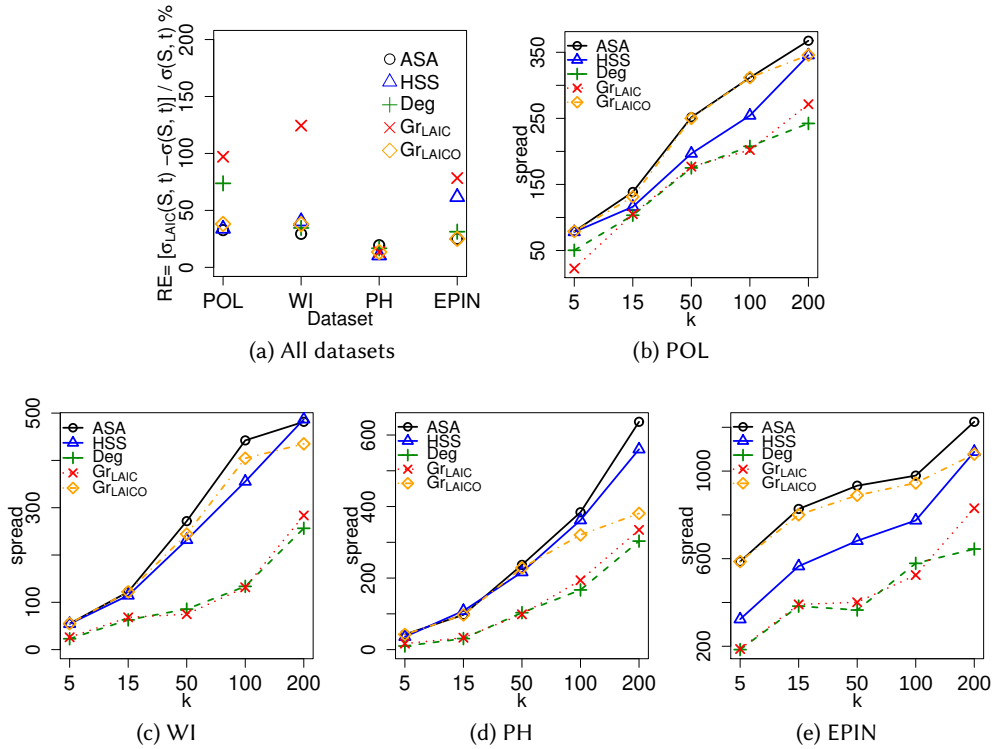


Fig. 8. (a) Relative error $RE = \frac{\sigma_{LAIC}(S,t)-\sigma(S,t)}{\sigma(S,t)}\%$ for the seed-set S found by each algorithm when $k = 50$. (b) to (e) Spread $\sigma(S, t)$ for the seed-set S found by each algorithm vs seed-set size k, for different datasets.

results of *Deg*, Gr_{LAIC}, and Gr_{LAICO} are omitted. Figs. 9e and 10a show that *HSS* is 2 orders of magnitude faster than *ASA* on average and that *HSS* scales sublinearly with k, while *ASA* scales subquadratically with k. This finding is in line with the time-complexity results for *ASA* and *HSS*, which show that *ASA* performs more function evaluations than *HSS* (i.e., $O(|V| \cdot k)$ vs. $O(|V| \cdot I)$, where I was at most 4 in our experiments) when k is larger. Also, by combining the results in Figs. 9e and 10a with those in Figs. 8c and 8d, which show that *HSS* and *ASA* achieve comparable results in terms of spread, one can see that *HSS* is a practical method to address the *IML* problem. Yet, since *HSS* is a heuristic, *ASA* is the preferred choice when guarantees for the quality of the solution to the *IML* problem are needed.

Fig. 10b shows that *HSS* is 5.2 times faster than *ASA* on average and that both methods scale sublinearly with the window size $|W|$. The runtime of both methods increases with $|W|$, because the number of nodes that may be activated generally increases as $|W|$ increases, which in turn implies that the methods need more time to compute their objective functions (σ for *ASA* and σ, $\sigma_{\mathcal{L}}$ and $\sigma_{\mathcal{U}}$ for *ISS*). The difference in the runtime of the algorithms is attributed to the fact that *ASA* evaluates the spread function σ, while *ISS* evaluates the proxy function $\widehat{\sigma_{S^{i-1}}}$, which is much faster to compute. The reason that the proxy function $\widehat{\sigma_{S^{i-1}}}$ is faster to compute, compared to σ, is that it is based on the modular upper bound, which is computed efficiently [17] as it is based on single nodes and a fixed parameter set (see Section 2.1).

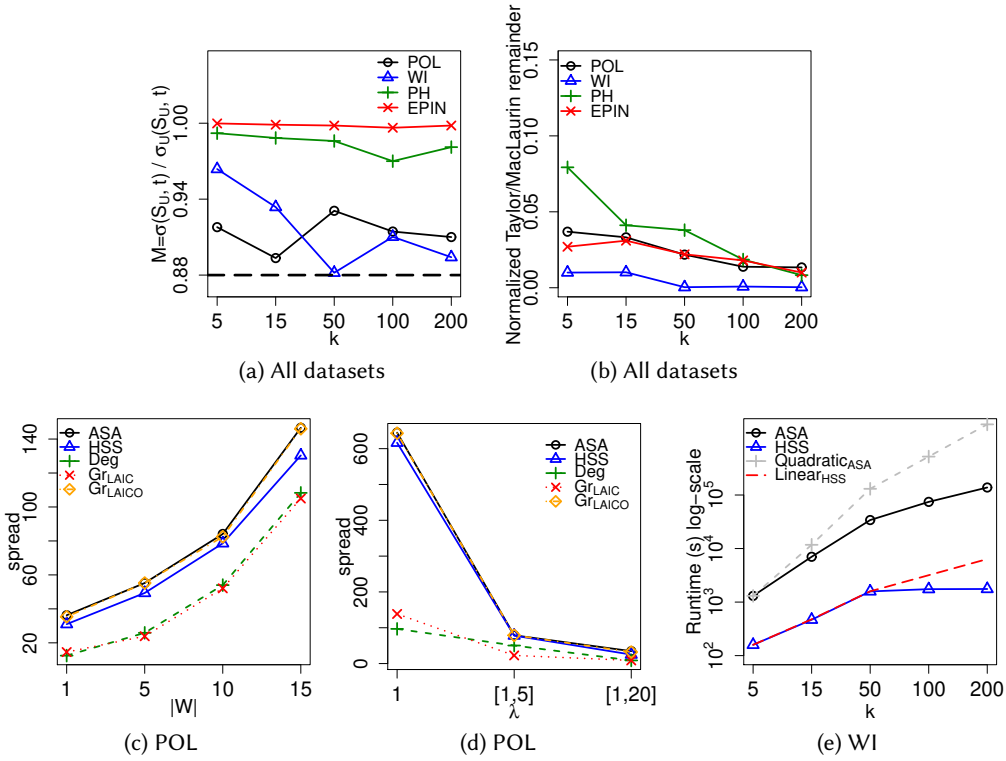


Fig. 9. (a) The term $\mathcal{M} = \sigma(S_{\mathcal{U}},t)/\sigma_{\mathcal{U}}(S_{\mathcal{U}},t)$ in the approximation ratio $\mathcal{M} \cdot (1 - \frac{1}{e})$ of *ASA* vs k. (b) Normalized Taylor/MacLaurin remainder, defined as $-\sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1-\mathcal{P}_{LAIC}(u,S,[0,t]))\cdot O(u,S,t))^n}{n} / \widehat{\sigma_{S^{i-1}}}(S,t)$ (see Lemma 4), vs k for all datasets. Spread $\sigma(S,t)$ vs (c) window size $|W|$, and (d) mean rate λ. (e) Runtime vs k for the *WI* dataset.

Fig. 10c shows that *HSS* is 2.6 times faster than *ASA* on average when λ varies. The runtime of both methods decreases when λ increases, because the edge probabilities become smaller and thus more paths have path probability lower than h and are pruned. Again, *HSS* is faster than *ASA* because it employs the proxy function $\widehat{\sigma_{S^{i-1}}}$ instead of the spread function σ that is employed in *ASA* and is more expensive to compute.

To sum up, *HSS* is more efficient than *ASA* for three reasons: (I) It selects seeds from a sample of the graph nodes, which implies that it typically performs a smaller number of function evaluations. (II) Its proxy function is computed much faster than the spread function used in *ASA*, and (III) it needs at most 4 iterations to terminate.

7.8 Impact of h

Figs. 10d and 10e show that the spread for *ASA* and *HSS* decreased by less than 1.8%, when $h \leq 5 \cdot 10^{-3}$ and substantially for larger h values, while the runtime of both methods decreased as h increased. Similar trends were observed for the other datasets. Thus, setting $h = 5 \cdot 10^{-3}$ offers a good effectiveness/efficiency trade-off.

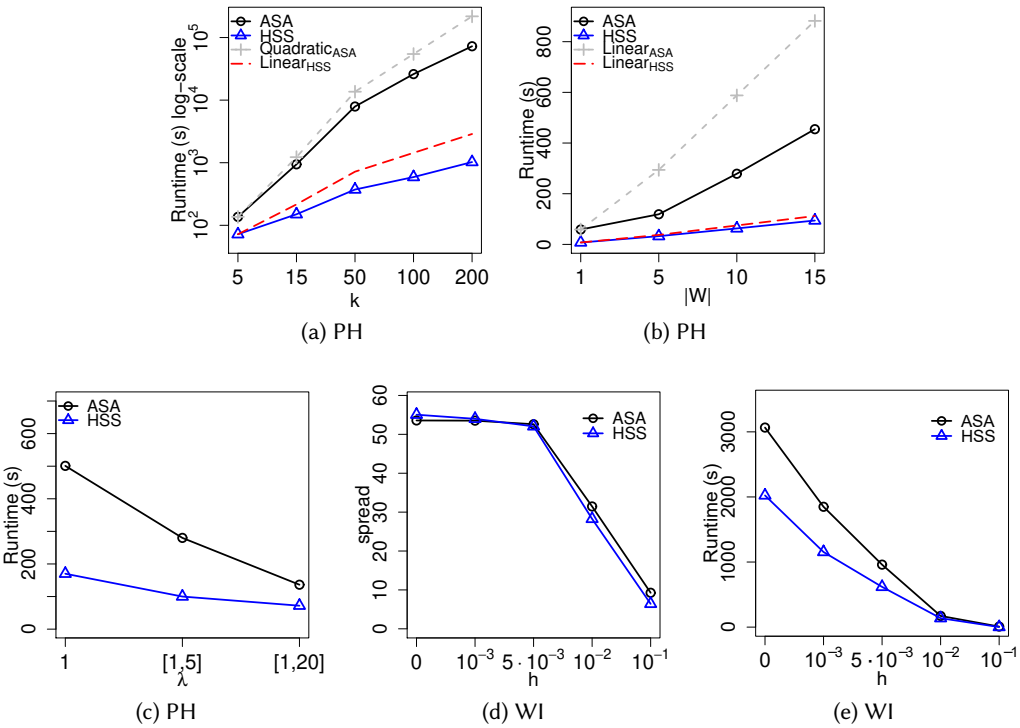


Fig. 10. Runtime vs (a) k, (b) window size $|W|$, and (c) mean rate λ, for the PH dataset. Impact of the minimum path probability threshold h on (d) spread and (e) runtime.

8 RELATED WORK

Influence diffusion plays a key role in analytics [42, 43]. In the following, we review existing works focusing on the three issues around influence diffusion that are also considered by our work: activation probability modeling (Section 8.1), influence maximization (Section 8.2), and negative aspects of information diffusion (Section 8.3).

8.1 Activation probability modeling

Several works [13, 23, 31, 36] considered the issue of modeling the activation probability of users.

The works of [23] and [31] used past interaction data from a recommendation website and Twitter, respectively, to study a fundamental question in viral marketing: "What determines the probability that a user is activated by their friends?". They found that this probability is determined by factors including the advertised product, information transmitted through channels that are external to a social network (e.g., TV and newspapers), and number of received activation attempts (e.g., product recommendations).

The works of [13, 31, 36] studied how the activation probability of a user is affected by the number of activation attempts. In [31, 36], this number is used to model the probability that a Twitter user will diffuse information they receive. In [13], this number is used to learn an exponential function that penalizes the activation probability of a node in the IC model.

Our work differs from the works of [13, 23, 31, 36] along three important dimensions. First, it does not use past user interaction data but an evaluation study of a sample of users, targeted by the

viral marketing campaign. Second, our work derives a logistic function that penalizes the activation probability of a node in the LAIC model. Third, our work uses the ratio between the estimated and the maximum possible number of attempts performed to activate a user, which captures empirical findings related to overexposure in marketing [6] and economics [3].

8.2 Influence maximization

Influence maximization is a key problem in viral marketing that has been studied extensively [24]. Typically, the problem is to select at most k users who will influence the largest number of other users, if they start a diffusion process which proceeds as specified by an influence diffusion model.

Influence diffusion models can be classified into *time-unaware* models [14, 15, 20], in which information is diffused until no new node becomes active, and *time-aware* models [9, 25, 33], in which information is diffused within a time window. Examples of time-unaware models are the Independent Cascade (IM) model [14, 20] and the Linear Threshold (LT) model [15]. Examples of time-aware models are the Independent Cascade with Meeting points (IC-M) model [9] and the Latency Aware Independent Cascade (LAIC) model [25]. LAICO is a time-aware model that is based in LAIC. However, it differs from existing time-aware models in that it considers overexposure and in that its spread function is not submodular, which prevents the use of methods based on *Greedy* for influence maximization.

Several algorithms for influence maximization have been proposed [24]. These include heuristics [10, 11, 13, 25] and approximation algorithms [20, 28, 29, 39]. For example, path-based heuristics were proposed in [10, 11], an efficient algorithm that uses Reverse Influence Sets [7] to approximate spread was proposed in [39], and the Sandwich Approximation strategy, which forms the basis of our *ASA* algorithm, was proposed in [28]. The Sandwich Approximation strategy was also employed in [44], which considered the collective impact of subsets of the in-neighbors of a node on its activation probability. Unlike ours, existing algorithms for influence maximization do not consider overexposure (i.e., the negative impact on the activation probability of a user, as a function of the ratio of the activation attempts the user receives).

8.3 Negative aspects of information diffusion

While no existing research considers overexposure, there is much research on other negative aspects of information diffusion. First, we discuss works on influence maximization when some users are negatively inclined towards the advertised product and do not diffuse information about it. These works are the most similar to ours, since they consider influence maximization. Then, we discuss works beyond influence maximization. The goal of these works is to block rumors who may negatively impact users when propagated, or to prevent the influence of users who could be harmed by the diffused information (*vulnerable users*).

Influence maximization with negatively inclined users. Two recent works [2, 45] consider an influence maximization setting, in which some users are negatively inclined towards the advertised product and will not diffuse information about it when they are activated.

The goal of [2] is to select at most k users who will maximize the *payoff* of an advertised product, if they start an information diffusion process. The payoff is a sum of terms, each contributed by a user as follows: activated users who are positively (respectively, negatively) inclined towards the product contribute a positive (respectively, negative) term, whereas inactive users contribute a zero term. The term of each user and whether the user will be positively or negatively inclined is decided before the diffusion process starts. The set of activated users is calculated as follows: Initially, the selected k users, who are positively inclined, inform their out-neighbors. Each out-neighbor is either positively inclined and informs its own out-neighbors about it, or it is negatively inclined and does not inform it own out-neighbors about it. The process ends when no newly activated user

is positively inclined towards the product. The work of [2] shows interesting NP-hardness results for the problem but provides no algorithm for it.

The work of [45] is similar to [2] in that it aims to select k users who will maximize the payoff, if they start an information diffusion process, and in that whether a user is positively or negatively inclined is decided before the diffusion process starts. However, [45] extends the IC model by assuming that negatively inclined users do not attempt to influence others. Furthermore, [45] proposes an approximation algorithm for the case where the number of negatively inclined users is lower than k.

We now summarize the main differences between the works of [2, 45] and our work:

- *Meaning of "overexposure"*: In [2] and [45], the term "overexposure" is used to describe the negative impact to a business from diffusing information to negatively inclined users (e.g., higher-income users who will not purchase or advertise a cheap product [2]). On the contrary, we use the term "overexposure" as in the marketing literature [3, 19], to describe the negative impact to users as a result of receiving information about the same product from too large a fraction of their friends.
- *Diffusion model*: In [2] and [45], the users are categorized into positive and negative inclined, before the diffusion process starts. Also, negatively inclined users do not advertise it further. On the contrary, we do not adopt such a user categorization, and each influenced user in our model will attempt to influence all its inactive out-neighbors.
- *Optimization objective*: In [2] and [45], the objective is to maximize the payoff of the product, whereas in our work the objective is to maximize the spread. Also, the calculation of payoff and spread is fundamentally different.
- *Algorithms*: In [2], no algorithm to deal with the problem of selecting at most k users in a way that maximizes the payoff is provided [3]. On the contrary, we provide an approximation algorithm and a heuristic for our problem. The work of [45] provides an algorithm, under the extended IC model considered in [45], for when there are fewer than k negatively inclined users. This algorithm is not applicable to our setting, since in our setting the information is diffused according to the LAICO model and there are no negatively inclined users.

Rumor propagation. Rumor propagation can be minimized by blocking communication between users [21], which corresponds to edge deletion, or by blocking users' accounts [41], which corresponds to node deletion. The goal in [21] is to select a subset of k edges whose removal minimizes the spread of a rumor. The authors of [21] proposed two problem formulations differing in the employed measure of rumor spread and greedy approximation algorithms, for each formulation. The goal in [41] is to minimize the spread of a rumor, by blocking users' accounts, while ensuring that the negative impact on user experience is still acceptable (i.e., other users still receive a useful service). The impact on user experience is captured by a utility function which takes into account the blocking time. The work of [41] proposed two algorithms to select user accounts to block (nodes to delete); a greedy algorithm which selects all nodes at once, and a dynamic algorithm which selects nodes to delete in rounds. These works are orthogonal to ours, as they do not consider influence maximization or overexposure.

Limiting the activation probability of vulnerable users. The goal in [26] is to limit the activation probability of users who may be harmed by the diffused information, while preserving the structure of the graph representing the social network. To achieve this, [26] proposes two methods; a greedy approximation algorithm and a heuristic. Both methods aim to select a subset of edges

[3]Of note, [2] provides an algorithm for the case in which any number of users that accept the product can be selected as seeds. This algorithm solves a fundamentally different problem than our problem, under a different diffusion model. Thus, it is not an alternative to our algorithms.

to delete from the graph (which corresponds to blocking communication between pairs of users), while preserving the PageRank of the graph. The work of [26] is orthogonal to ours, as it does not consider influence maximization or overexposure.

9 CONCLUSION

Overexposure has negatively affected viral marketing campaigns. However, it is not taken into account by existing influence diffusion models, which overestimate spread, and it can also have a negative impact on the quality-of-service of systems [4]. This work proposes the LAICO influence diffusion model that captures overexposure, based on the ratio between the estimated and the maximum possible number of attempts to activate a user, as well as an approximation algorithm and a heuristic for influence maximization under LAICO. Our experiments demonstrate that the approximation algorithm is very effective but inefficient, for large k and window size $|W|$, while the heuristic trades-off effectiveness for efficiency and has a sublinear runtime in k and in $|W|$.

REFERENCES

[1] https://blog.linkedin.com/2015/07/27/less-email-from-linkedin?sf11404748=1.
[2] R. Abebe, L. A. Adamic, and J. Kleinberg. Mitigating overexposue in viral marketing. In *AAAI*, 2018.
[3] F. Alkemade and C. Castaldi. Strategies for the diffusion of innovations on social networks. *Comp. Econ.*, 25(1), 2005.
[4] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. *Comput. Netw.*, 33(1–6):1–16, 2000.
[5] J. A. Bilmes and W. Bai. Deep submodular functions. *CoRR*, abs/1701.08939, 2017.
[6] K. Bontcheva, G. Gorrell, and B. Wessels. Social media and information overload. *CoRR*, abs/1306.0813, 2013.
[7] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, 2014.
[8] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. Submodular maximization with cardinality constraints. In *SODA*, 2014.
[9] W. Chen, W. Lu, and N. Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *AAAI*, 2012.
[10] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, page 1029–1038, 2010.
[11] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, page 199–208, 2009.
[12] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. *SIAM J. Comput.*, 29(5):1484–1496, 2000.
[13] S. Feng, X. Chen, G. Cong, Y. Zeng, Y. M. Chee, and Y. Xiang. Influence maximization with novelty decay in social networks. In *AAAI*, 2014.
[14] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.
[15] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.
[16] R. Gwadera and G. Loukides. Cost-effective viral marketing in the latency aware independent cascade model. In *PAKDD*, pages 251–265, 2017.
[17] R. Iyer and J. Bilmes. Algorithms for approximate minimization of the difference between submodular functions, with applications. In *UAI*, 2012.
[18] J. Jeong and S. Moon. Invite your friends and get rewards: Dynamics of incentivized friend invitation in kakaotalk mobile games. In *ACM COSN*, 2014.
[19] K. Kalyanam, S. McIntyre, and T. J. Masonis. Adaptive experimentation in interactive marketing: The case of viral marketing at plaxo. *J. of Inter. Marketing*, 21(3):72–85, 2007.
[20] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
[21] M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Trans. Knowl. Discov. Data*, 3(2), 2009.
[22] A. Krause and D. Golovin. Submodular function maximization. In *Tractability*. 2013.
[23] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5–es, May 2007.
[24] Y. Li, J. Fan, Y. Wang, and K. Tan. Influence maximization on social graphs: A survey. *TKDE*, 30(10):1852–1872, 2018.
[25] B. Liu, G. Cong, D. Xu, and Y. Zeng. Time constrained influence maximization in social networks. In *ICDM*, 2012.
[26] G. Loukides and R. Gwadera. Preventing the diffusion of information to vulnerable users while preserving pagerank. *Int J of Data Science and Analytics*, 5(1):19–39, 2018.

[27] J. J. Louviere, D. A. Hensher, and J. D. Swait. *Stated Choice Methods Analysis and Applications*. 2000.
[28] W. Lu, W. Chen, and L. V. S. Lakshmanan. From competition to complementarity: Comparative influence diffusion and maximization. *PVLDB*, 9(2), 2015.
[29] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, 1978.
[30] M. Mitrovic, M. Bun, A. Krause, and A. Karbasi. Differentially private submodular maximization: Data summarization in disguise. In *ICML*, 2017.
[31] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD*, pages 33–41, 2012.
[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.
[33] N. Ohsaka, Y. Yamaguchi, N. Kakimura, and K. Kawarabayashi. Maximizing time-decaying influence in social networks. In *ECML/PKDD*, 2016.
[34] Ayse Bengi Ozcelik and Kaan Varnali. Effectiveness of online behavioral targeting: A psychological perspective. *Electronic Commerce Research and Applications*, 33:100819, 2019.
[35] C. C. Pugh. *Real mathematical analysis*. 2015.
[36] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics. In *WWW*, 2011.
[37] D. Simchi-Levi, X. Chen, and J. Bramel. *Convexity and Supermodularity*. Springer New York, 2005.
[38] Z. Svitkina and L. Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM J. Comput.*, 40(6):1715–1737, 2011.
[39] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *SIGMOD*, 2015.
[40] A. Volkova. A refinement of the central limit theorem for sums of independent random indicators. *Theory of Probability & Its Applications*, 40(4):791–794, 1996.
[41] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang. Drimux: Dynamic rumor influence minimization with user experience in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2168–2181, 2017.
[42] L. Wu, P. Sun, Y. Fu Yanjie, R. Hong, X. Wang, and M. Wang. A neural influence diffusion model for social recommendation. In *SIGIR*, page 235–244, 2019.
[43] Y. Zhou and L. Liu. Social influence based clustering of heterogeneous information networks. In *KDD*, page 338–346, 2013.
[44] J. Zhu, J. Zhu, S. Ghosh, W. Wu, and J. Yuan. Social influence maximization in hypergraph in social networks. *IEEE Transactions on Network Science and Engineering*, 6(4), 2019.
[45] Y. Zhu, P. Yin, D. Li, and B. Lin. Strengthening the positive effect of viral marketing. In *ICDCS*, pages 1941–1950, 2019.

A RELATION BETWEEN *ASA* AND THE SANDWICH APPROXIMATION (*SA*) STRATEGY

The *SA* strategy approximates the following problem: Given a non-negative **non-submodular** function $f : 2^U \rightarrow \mathbb{R}_{\geq 0}$, non-negative monotone submodular functions $l_f : 2^U \rightarrow \mathbb{R}_{\geq 0}$ and $u_f : 2^U \rightarrow \mathbb{R}_{\geq 0}$ such that $l_f(S) \leq f(S) \leq u_f(S)$ for each subset $S \subseteq U$, and a parameter k, find a subset S of size $|S| \leq k$ with maximum $f(S)$. The *SA* strategy applies *Greedy* three times: with f, to produce a subset S_f; with l_f to produce a subset S_{l_f}, and with u_f to produce a subset S_{u_f}. Then, *SA* returns the subset in $\{S_f, S_{l_f}, S_{u_f}\}$ with the largest value in f.

Thus, *ASA* is an adaptation of the *SA* strategy, which uses the non-negative non-submodular spread function $\sigma(S, t)$ as f, and the non-negative, monotone submodular functions $\sigma_{\mathcal{L}}(S, t)$ and $\sigma_{\mathcal{U}}(S, t)$ as the lower and upper bound functions l_f and u_f, respectively.

B PROOF OF THEOREM 1

We first show that Gr_{LAICO} offers the following approximation guarantee:

$$\frac{\sigma(S_{Gr}, t)}{\sigma(S^*_{IML}, t)} \geq \frac{1 - \frac{1}{\mathrm{e}}}{1 + k \cdot \left(\left(\frac{R(0)}{R(1)}\right)^2 - 1\right)}, \tag{10}$$

where S^*_{IML} is the optimal solution of the *IML* problem, $\sigma(S_{Gr}, t)$ and $\sigma(S^*_{IML}, t)$ is the spread of S_{Gr} and S^*_{IML} in the window W, respectively, e is the base of the natural logarithm, and $R(1)$ and $R(0)$ is the minimum and maximum value of the logistic regression function R, respectively.

Let S_i be the subset of nodes found by *Greedy* in iteration $i \in [1,k]$, and S^*_{IML} be the optimal solution of the *IML* problem. Let also $\sigma_A(S_i,t) = \sum_{u\in V_1} \mathcal{P}(u,S^i,[0,t])$, where $V_1 = \{u \in V \mid \mathbf{O}(u,S,t) = 1\}$, and $\sigma_{B'}(S_i,t) = \sum_{u\in V_{<1}} \mathcal{P}(u,S^i,[0,t])$ be a function measuring the spread of nodes with overexposure scores lower than 1 in the LAIC model, where $V_{<1} = \{u \in V \mid \mathbf{O}(u,S,t) < 1\}$.

We observe that:

$$\begin{aligned}\sigma_A(S^*_{IML},t) &\leq \sigma_A(S_i,t) + \sum_{u\in S^*_{IML}} [\sigma_A(S_i\cup u,t) - \sigma_A(S_i,t)]\\ &\leq \sigma_A(S_i,t) + k\cdot[\sigma_A(S_{i+1},t) - \sigma_A(S_i,t)].\end{aligned}$$

The first inequality follows from the submodularity of σ_A [25] and the second inequality follows from the fact that S^*_{IML} has size at most k. By reordering the terms of the second inequality, we obtain:

$$\sigma_A(S_{i+1},t) \geq \frac{1}{k}\cdot[\sigma_A(S^*_{IML},t) + (k-1)\cdot\sigma_A(S_i,t)]. \tag{11}$$

Similarly, by the submodularity of σ_B, the fact that $R(0) \geq R(1)$, and the fact that S^*_{IML} has size at most k, we obtain:

$$\begin{aligned}\sigma_{B'}(S^*_{IML},t) &\leq \sigma_{B'}(S_i,t) + \sum_{u\in S^*_{IML}} [\sigma_{B'}(S_i\cup u,t) - \sigma_{B'}(S_i,t)]\\ &\leq \sigma_{B'}(S_i,t) + \sum_{u\in S^*_{IML}} \left[\frac{R(0)}{R(1)}\cdot\sigma_{B'}(S_i\cup u,t) - \sigma_{B'}(S_i,t)\right]\\ &\leq \sigma_{B'}(S_i,t) + k\cdot\left[\frac{R(0)}{R(1)}\cdot\sigma_{B'}(S_{i+1},t) - \sigma_{B'}(S_i,t)\right]\end{aligned}$$

By reordering the terms of the third inequality and expressing it using σ_B, based on

$$R(1)\cdot\sigma_{B'}(S,t) \leq \sigma_B(S,t) \leq R(0)\cdot\sigma_{B'}(S,t),$$

we obtain:

$$\sigma_B(S_{i+1},t) \geq \frac{1}{k}\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot[\sigma_B(S^*_{IML},t) + (k-1)\cdot\sigma_B(S_i,t)]. \tag{12}$$

Since $\sigma(S,t) = \sigma_A(S,t) + \sigma_B(S,t)$, for each $S \subseteq V$ and $R(1) \leq R(0)$, we obtain the following by adding Eqs. 11 and 12 together:

$$\sigma(S_{i+1},t) \geq \frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot\sigma(S_i,t) + \frac{1}{k}\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot\sigma(S^*_{IML},t). \tag{13}$$

Eq. 13 is of the form $u_{i+1} = \alpha\cdot u_i + \beta$, where $\alpha = \frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2$, $\beta = \frac{1}{k}\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot\sigma(S^*_{IML},t)$ and $u_0 = 0$. Thus, recursively, we obtain $u_i \geq \beta\cdot\frac{1-\alpha^i}{1-\alpha}$, which implies

$$\sigma(S_i,t) \geq \frac{1}{k}\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot\sigma(S^*_{IML},t)\cdot\frac{\left[1-\left(\frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2\right)^i\right]}{1-\frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2}. \tag{14}$$

Since *Greedy* performs k iterations and Eq. 14 holds for each $i \in [1,k]$, we obtain:

$$\sigma(S_k,t) \geq \frac{1}{k}\cdot\left(\frac{R(1)}{R(0)}\right)^2\cdot\frac{\left[1-\left(\frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2\right)^k\right]}{1-\frac{1}{k}\cdot(k-1)\cdot\left(\frac{R(1)}{R(0)}\right)^2}\cdot\sigma(S^*_{IML},t)$$

which can be written as:

$$\sigma(S_k,t) \geq \frac{1-\left(1-\frac{1}{k}\right)^k\cdot\left(\frac{R(1)}{R(0)}\right)^{2\cdot k}}{1+k\cdot\left(\left(\frac{R(0)}{R(1)}\right)^2-1\right)}\cdot\sigma(S^*_{IML},t). \tag{15}$$

It can be shown that $(1-\frac{1}{m})^m \geq \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{m})$ holds, for $m \geq 1$, by induction. Thus, Eq. 15 yields:

$$\frac{\sigma(S_k,t)}{\sigma(S^*_{IML},t)} \geq \frac{1-\frac{1}{\mathrm{e}} \cdot \left(1-\frac{1}{k}\right) \cdot \left(\frac{R(1)}{R(0)}\right)^{2 \cdot k}}{1+k \cdot \left(\left(\frac{R(0)}{R(1)}\right)^2-1\right)} \geq \frac{1-\frac{1}{\mathrm{e}}}{1+k \cdot \left(\left(\frac{R(0)}{R(1)}\right)^2-1\right)}.$$

We now use the following inequalities, which hold by definition:

$$\sigma(S_{\mathcal{U}},t) \geq \sum_{u \in V} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(1))$$
$$\sigma_{\mathcal{U}}(S_{\mathcal{U}},t) \leq \sum_{u \in V} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(0))$$
$$\sigma_{\mathcal{L}}(S^*_{IML},t) \geq \sum_{u \in V} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(1))$$
$$\sigma(S^*_{IML},t) \leq \sum_{u \in V} (\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(0))$$

to rewrite Eq. 6 as:

$$\frac{\sigma(S_{ASA},t)}{\sigma(S^*_{IML},t)} \geq \max\{\frac{\sum_{u \in V}(\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(1))}{\sum_{u \in V}(\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(0))}, \quad \frac{\sum_{u \in V}(\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(1))}{\sum_{u \in V}(\mathcal{P}_{LAIC}(u,S,[0,t]) \cdot R(0))}\} \cdot \left(1-\frac{1}{\mathrm{e}}\right),$$

which implies

$$\frac{\sigma(S_{ASA},t)}{\sigma(S^*_{IML},t)} \geq \frac{R(1)}{R(0)} \cdot \left(1-\frac{1}{\mathrm{e}}\right) \tag{16}$$

The proof follows by dividing Eqs. 16 and 10. □

C PROOF OF LEMMA 1

Let V be the set of nodes of the graph G, $S \subseteq S' \subseteq V$ be subsets of V, and v be a node in $V \setminus S'$. Let also $\mathcal{N}_u^{S,t} \subseteq n^-(u)$, $\mathcal{N}_u^{S',t} \subseteq n^-(u)$, and $\mathcal{N}_u^{\{v\},t} \subseteq n^-(u)$ be the set of in-neighbors of u that may activate u for a seed-set S, S', and $\{v\}$, respectively.

We first show that $\mathbf{O}(u,S,t)$ is non-increasing, by proving that Eq. 17 holds in each possible case Ia to IIIa below.

$$\mathbf{O}(u,S,t) \geq \mathbf{O}(u,S',t) \tag{17}$$

Specifically, since $N_u^{S,t} \leq N_u^{S',t}$ by definition, where $N_u^{S,t}$ (respectively, $N_u^{S',t}$) is the estimated number of activation attempts to u when the seed-set is S (respectively, S'), the cases are as follows:
Case Ia: $1 < N_u^{S,t} \leq N_u^{S',t}$. In this case, Eq. 17 holds, because $\mathbf{O}(u,S,t)$ is given by the logistic regression whose first derivative is negative for $\beta_1 < 0$ as in our case.
Case IIa: $N_u^{S,t} \leq 1 < N_u^{S',t}$. In this case, Eq. 17 holds, because $\mathbf{O}(u,S,t) = 1$ and $\mathbf{O}(u,S',t) < 1$ (since it is given by the logistic function, whose values cannot exceed 1).
Case IIIa: $N_u^{S,t} = N_u^{S',t} \leq 1$. In this case, Eq. 17 holds, because $\mathbf{O}(u,S,t) = \mathbf{O}(u,S',t) = 1$.

We now show that $\mathbf{O}(u,S,t)$ is supermodular, by proving that Eq. 18 holds in each possible case Ib to IIIb below.

$$\mathbf{O}(u, S \cup v, t) - \mathbf{O}(u, S, t) \leq \mathbf{O}(u, S' \cup v, t) - \mathbf{O}(u, S', t) \tag{18}$$

Case Ib: All nodes in $\mathcal{N}_u^{\{v\},t}$ are contained in $\mathcal{N}_u^{S,t}$. Thus,

$$\mathbf{O}(u, S \cup v, t) - \mathbf{O}(u, S, t) = 0 \leq \mathbf{O}(u, S' \cup v, t) - \mathbf{O}(u, S', t) = 0$$

since adding v into S (respectively, S') results in $N_u^{S\cup v,t} = N_u^{S,t}$ (respectively, $N_u^{S'\cup v,t} = N_u^{S',t}$).
Case IIb: All nodes in $\mathcal{N}_u^{\{v\},t}$ are contained in $\mathcal{N}_u^{S',t}$ and at least one node in $\mathcal{N}_u^{\{v\},t}$ is not contained in $\mathcal{N}_u^{S,t}$. Thus,

$$\mathbf{O}(u, S \cup v, t) - \mathbf{O}(u, S, t) \leq \mathbf{O}(u, S' \cup v, t) - \mathbf{O}(u, S', t) = 0$$

since adding v into S results in $N_u^{S\cup v,t} \geq N_u^{S,t}$ and $\mathbf{O}(u, S, t)$ is non-increasing, while adding v into S' results in $N_u^{S'\cup v,t} = N_u^{S',t}$.
Case IIIb: At least one node in $\mathcal{N}_u^{\{v\},t}$ is not contained in $\mathcal{N}_u^{S',t}$. Thus,

$$\mathbf{O}(u, S \cup v, t) - \mathbf{O}(u, S, t) \leq \mathbf{O}(u, S' \cup v, t) - \mathbf{O}(u, S', t)$$

since: (i) adding v into S causes the addition into $\mathcal{N}_u^{S,t}$ of all nodes that are not contained in $\mathcal{N}_u^{S',t}$, and all nodes that are contained in $\mathcal{N}_u^{S',t} \setminus \mathcal{N}_u^{S,t}$, which implies that $N_u^{S\cup v,t} - N_u^{S,t} \geq N_u^{S'\cup v,t} - N_u^{S',t}$, and (ii) the function $\mathbf{O}(u, X \cup \{v\}, t) - \mathbf{O}(u, X, t)$ is non-increasing for each node v and subset $X \subseteq V$ (the proof is similar to the proof that $\mathbf{O}(u, S, t)$ is non-increasing and is omitted). □

D PROOF OF THEOREM 2

The proof is comprised of two steps. In step (I), we show that $\sum_{u\in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t]))$ is a submodular function. In step (II), we show that $\sum_{u\in V} (-\ln(\mathbf{O}(u, S, t)))$ is a submodular function.

Step (I) The function $\ln(\mathcal{P}_{LAIC}(u, S, [0, t]))$ is the composition of the natural logarithm function, which is monotone concave, and of the function $\mathcal{P}_{LAIC}(u, S, [0, t])$, which is monotone submodular, for any seed-set S, according to [25]. Thus, $\ln(\mathcal{P}_{LAIC}(u, S, [0, t]))$ is monotone submodular for any seed-set S, as a composition of a monotone concave function and a monotone submodular function [5]. Consequently, the sum $\sum_{u\in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t]))$ is a non-negative linear combination of submodular functions (i.e., a weighted sum of submodular functions, where the weights are non-negative constants) and hence it is a submodular function [22].

Step (II) We will first consider a helper function $\frac{1}{-\mathbf{O}(u,S,t)}$. The function is a composition of the function $\frac{1}{x}$, which is non-increasing convex, and of the function $-\mathbf{O}(u, S, t)$, which is monotone submodular (since $\mathbf{O}(u, S, t)$ is non-increasing supermodular according to Lemma 1). Thus, the helper function $\frac{1}{-\mathbf{O}(u,S,t)}$ is supermodular, as a composition of a non-increasing convex function and a monotone submodular function [37]. It can also be easily shown that the helper function is non-increasing. Therefore, the helper function $\frac{1}{-\mathbf{O}(u,S,t)}$ is non-increasing supermodular, and the function $\frac{1}{\mathbf{O}(u,S,t)}$ monotone submodular. We now consider the composition of the monotone concave function $\ln(x)$ and of the monotone submodular function $\frac{1}{\mathbf{O}(u,S,t)}$. The composition is the function $\ln(\frac{1}{\mathbf{O}(u,S,t)}) = -\ln(\mathbf{O}(u, S, t))$, which is monotone submodular, as a composition of a monotone concave function and a monotone submodular function [5]. Consequently, the sum

$$\sum_{u\in V} (-\ln(\mathbf{O}(u, S, t)))$$

is a non-negative linear combination of submodular functions and hence it is a submodular function. □.

E PROOF OF LEMMA 2

We prove the lemma by a counterexample, which is similar to Example 3except that the function $\widehat{\sigma}$ is used instead of the spread function σ.

Consider the graph of Fig. 3a, the window $[0, 2]$, and that the overexposure score of each node u, for a seed-set S and window $[0, 2]$, was calculated using Eq. 3 with the logistic function $\frac{1}{1+e^{-\left(\beta_0+\beta_1 \cdot \tilde{N}_u^{S,t}\right)}}$ whose coefficients $\beta_0 = 1.61977$ and $\beta_1 = -5.00491$ were obtained by a user evaluation study. The function $\widehat{\sigma}(S, t)$ is *not* submodular, because for $\{u_1\} \subseteq \{u_1, u_3\}$ and $u_2 \in \{u_1, \dots u_7\} \setminus \{u_1, u_3\}$, it holds that $\widehat{\sigma}(\{u_1\} \cup \{u_2\}, 2) - \widehat{\sigma}(\{u_1\}, 2) = -1.88204 < \widehat{\sigma}(\{u_1, u_3\} \cup \{u_2\}, 2) - \widehat{\sigma}(\{u_1, u_3\}, 2) = -3.418459 - (-1.88204) = -1.596412$. In addition, $\widehat{\sigma}$ is *not* supermodular, because for $\{u_1\} \subseteq \{u_1, u_2, u_3\}$ and $u_4 \in \{u_1, \dots, u_7\} \setminus \{u_1, u_2, u_3\}$, it holds that $\widehat{\sigma}(\{u_1\} \cup \{u_4\}, 2) - \widehat{\sigma}(\{u_1\}, 2) = 0 > \widehat{\sigma}(\{u_1, u_2, u_3\} \cup \{u_4\}, 2) - \widehat{\sigma}(\{u_1, u_2, u_3\}, 2) = -3.418459$. □

F PROOF OF THEOREM 3

We prove each of the properties I, II, and III below.

Property I. Recall that

$$\widehat{\sigma_Y}(S, t) = \sum_{u \in V} \left(\ln(\mathcal{P}_{LAIC}(u, S, [0, t]))\right) - \widehat{\mathbf{O}_Y}(S, t) - |V| \left(\ln(h) + (k + 1) \ln(R(1))\right).$$

Thus, it suffices to show that $|V| \left(\ln(h) + (k + 1) \ln(R(1))\right)$ is the minimum value of

$$\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t])) - \widehat{\mathbf{O}_Y}(S, t).$$

For ease of reference we rewrite $\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t])) - \widehat{\mathbf{O}_Y}(S, t)$ using Eq. 9 as follows:

$$\begin{aligned}\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t])) - \widehat{\mathbf{O}_Y}(S, t) = &\sum_{u \in V} \left(\ln(\mathcal{P}_{LAIC}(u, S, [0, t])) + \ln(\mathbf{O}(u, Y, t))\right) \\ &+ \sum_{u' \in S \setminus Y} \sum_{u \in V} \left(\ln(\mathbf{O}(u, \{u'\}, t))\right) \\ &+ \sum_{u' \in Y \setminus S} \sum_{u \in V} \left(\ln(\mathbf{O}(u, Y \setminus \{u'\}, t)) - \ln(\mathbf{O}(u, Y, t))\right) \qquad (19)\end{aligned}$$

Observe that the minimum value of $\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t])) - \widehat{\mathbf{O}_Y}(S, t)$ is obtained when each of the sums in the right-hand side of Eq. 19 is minimum.

The minimum value of the first sum is

$$\sum_{u \in V} \left(\ln(h) + \cdot \ln\left(R(1)\right)\right) = |V| \cdot \left(\ln(h) + \ln\left(R(1)\right)\right)$$

since $\ln\left(\mathcal{P}_{LAIC}(u, S, [0, t])\right) \geq \ln(h)$ and $\ln\left(\mathbf{O}(u, Y, t)\right) \geq \ln\left(R(1)\right)$. The minimum value of the second sum is $k \cdot |V| \cdot \ln(R(1))$, since $|S \setminus Y| \leq k$ by the way *Sup-Sub* works and $\ln\left(\mathbf{O}(u, Y, t)\right) \geq \ln(R(1))$. The minimum value of the third sum is 0. This is because the function $-\ln\left(\mathbf{O}(u, S, t)\right)$ is monotone with respect to S (see the proof of Theorem 2), which implies that $\ln\left(\mathbf{O}(u, S, t)\right)$ is non-increasing with respect to S and that $\ln\left(\mathbf{O}(u, Y \setminus \{u'\}, t)\right) - \ln\left(\mathbf{O}(u, Y, t)\right) \geq 0$, since $Y \setminus \{u'\} \subseteq Y$.

Therefore, the minimum value of the function in Eq. 19 is:

$$|V| \cdot \left(\ln(h) + \cdot \ln(R(1))\right) + k \cdot |V| \cdot \ln(R(1)) = |V| \cdot \left(\ln(h) + (k + 1) \cdot \ln(R(1))\right).$$

Property II. We prove this property by a counterexample. Consider the graph of Fig. 3a in the paper, the window $[0, 2]$, and that the overexposure score of each node u, for a seed-set S and window $[0, 2]$, was calculated using Eq. 3 with the logistic function $\frac{1}{1+e^{-\left(\beta_0+\beta_1 \cdot \tilde{N}_u^{S,t}\right)}}$ whose coefficients $\beta_0 = 1.61977$ and $\beta_1 = -5.00491$ were obtained by a user evaluation study. Also, consider $h = 10^{-3}$ and $k = 2$. The function $\widehat{\sigma_Y}(S, t)$ is non-monotone, because for $\{u_1\} \subseteq \{u_1, u_2\}$ and $Y = \varnothing$, it holds that $\widehat{\sigma_Y}(\{u_1\}, 2) = 120.1418 > \widehat{\sigma_Y}(\{u_1, u_2\}, 2)) = 117.5391$.

Property III. It suffices to show that the term $-\widehat{\mathbf{O}_{S^{i-1}}}(S, t)$ of the function $\widehat{\sigma}(S, t)$ is submodular, since the term $\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t]))$ is submodular (see Theorem 2) and $-|V| \left(\ln(h) + (k + 1) \ln(R(1))\right)$ is a non-negative constant (and hence modular and submodular). By showing this, we have that $\widehat{\sigma}(S, t)$ is submodular as a weighted sum of submodular functions with non-negative weights [22]. In fact, $\widehat{\mathbf{O}_Y}(S, t)$ is modular and hence supermodular (see Section 2.1). This implies that $-\widehat{\mathbf{O}_Y}(S, t)$ is supermodular. □

G OVERVIEW OF *SUBSAMPLE GREEDY*

Given a non-negative **submodular** function $f : 2^U \rightarrow \mathbb{R}_{\geq 0}$, and a parameter k, *Subsample Greedy* finds a subset $S \subseteq U$ of size $|S| \leq k$ with $\mathbb{E}[f(S)] \geq \frac{1}{e} \cdot (1 - \frac{1}{e}) \cdot \arg\max_{S' \subseteq U : |S'| \leq k} f(S')$, where $\mathbb{E}[f(S)]$ denotes the expected value of $f(S)$. The expected value is computed over every possible S constructed by *Subsample Greedy*. *Subsample Greedy* performs k iterations. In each iteration, it constructs a uniform random sample of U, adds into the sample a dummy element e (i.e., an element with marginal gain $f(X \cup \{e\}) - f(X) = 0$, for each $X \subseteq U$), and adds into the subset S the element with the maximum marginal gain in the sample. The sample has size $\frac{|U|}{k}$ which must be an integer. If it is not an integer, the minimum number of dummy elements are added into U. After k iterations, any dummy elements are removed from the subset S, and the subset is returned. *Subsample Greedy* performs $O(|U|)$ evaluations of f. Thus, it is more efficient than competitors [8] which perform $O(|U| \cdot k)$ evaluations.

H OVERVIEW OF *SUP-SUB*

Sup-Sub is an effective heuristic for the following *inapproximable* problem: Given submodular functions $f : 2^U \rightarrow \mathbb{R}$ and $g : 2^U \rightarrow \mathbb{R}$, find a subset $S \subseteq U$ of size $|S| \leq k$ with maximum $f(S) - g(S)$, where the objective function $f - g$ is not necessarily submodular. In [17], the problem was presented in its minimization form, obtained by swapping f with g. In each iteration i, *Sup-Sub*: (I) constructs a submodular proxy function $f(S) - \widehat{g_{S^{i-1}}}(S)$ of the function $f(S) - g(S)$, where $\widehat{g_{S^{i-1}}}(S)$ is the modular upper bound of $g(S)$ with parameter the subset S^{i-1} constructed in iteration $i - 1$, and (II) finds a subset $S^i \subseteq U$ of size $|S^i| \leq k$ and large or approximately maximum value in the proxy function using an input algorithm, selected based on the properties of the proxy function. *Sup-Sub* stops and returns S^{i-1}, when S^i has a smaller value in the proxy function than that of S^{i-1}. This stopping criterion guarantees that *Sup-Sub* terminates.

I PROOF OF LEMMA 3

Before providing the proof of Lemma 3, we prove an auxiliary lemma below.

Lemma 4. *For a seed-set S and window $[0, t]$, it holds that*

$$\widehat{\sigma}(S, t) = \sigma(S, t) - |V| - \sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t))^n}{n}, \tag{20}$$

where $|V|$ is the number of nodes in the graph and n is an integer.

Proof. The Taylor series of the function $\ln(1 - x)$, $x \in [-1, 1)$, centered at 0 (also known as MacLaurin series) is $-\sum_{n=1}^{\infty} \frac{x^n}{n}$ [35]. Thus, $\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n}$. Therefore, by substituting $1-x$ with $\mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t) \in (0, 1]$ (recall that nodes with $\mathcal{P}_{LAIC}(u, S, [0, t]) = 0$ are ignored in the computation of spread), summing over each node $v \in V$, we obtain:

$$\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t)) = -\sum_{u \in V} (1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t)) - \sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t))^n}{n}.$$

The proof follows from equality

$$\sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t)) = \widehat{\sigma}(S, t),$$

which is easily obtained from Eq. 7, and equality

$$-\sum_{u \in V} (1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t)) = \sigma(S, t) - |V|,$$

which holds from the definition of spread in the LAICO model. □

We are now ready to provide the proof of Lemma 3.

Proof. For any seed-set S^i and window $[0, t]$, it holds that

$$\sigma(S^i, t) = \widehat{\sigma}(S^i, t) + |V| + \sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t))^n}{n} \tag{21}$$

according to the auxiliary Lemma 4.

Furthermore, the following equations hold:

$$\widehat{\sigma}(S^i, t) = \sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S^i, [0, t])) - \sum_{u \in V} (-\ln(\mathbf{O}(u, S^i, t))) \Rightarrow \tag{22}$$

$$\widehat{\sigma}(S^i, t) \geq \sum_{u \in V} \ln(\mathcal{P}_{LAIC}(u, S^i, [0, t])) - \widehat{\mathbf{O}_{S^{i-1}}}(S^i, t). \tag{23}$$

Eq. 22 holds by definition and Eq. 23 holds because, from the construction of proxy function, $\widehat{\mathbf{O}_{S^{i-1}}}(S, t)$ is a modular upper bound of $\sum_{u \in V} (-\ln(\mathbf{O}(u, S, t)))$. In addition, the following holds from Eqs. 7 and 8:

$$\widehat{\sigma}(S^i, t) \geq \widehat{\sigma_{S^{i-1}}}(S, t) + |V|(\ln(h) + (k + 1) \cdot \ln(R(1))). \tag{24}$$

Thus, from Eqs. 21 and 24, we obtain:

$$\sigma(S^i, t) \geq \widehat{\sigma_{S^{i-1}}}(S, t) + [|V|(1 + \ln(h) + (k + 1) \cdot \ln(R(1)))] + \sum_{u \in V} \sum_{n=2}^{\infty} \frac{(1 - \mathcal{P}_{LAIC}(u, S, [0, t]) \cdot \mathbf{O}(u, S, t))^n}{n}.$$

□

J QUESTIONNAIRE

The questionnaire (in pdf format) that we used in our user evaluation study can be found at: https://www.dropbox.com/s/upz0f21upxdz6ng/questionnaire.pdf?dl=0. Ethical approval for the study was obtained by the King's College Research Ethics Committee. The questionnaire was managed through QuestionPro.