Limiting the Influence to Vulnerable Users in Social Networks: A Ratio Perspective

Huiping Chen, Grigorios Loukides, Jiashi Fan and Hau Chan

Abstract Influence maximization is a key problem in social networks, seeking to find users who will diffuse information to influence a large number of users. A drawback of the standard influence maximization is that it is unethical to influence users many of whom would be harmed, due to their demographics, health conditions, or socioeconomic characteristics (e.g., predominantly overweight people influenced to buy junk food). Motivated by this drawback and by the fact that some of these vulnerable users will be influenced inadvertently, we introduce the problem of finding a set of users (*seeds*) that limits the influence to vulnerable users while maximizing the influence to the non-vulnerable users. We define a measure that captures the quality of a set of seeds, as an additively smoothed ratio between the expected number of influenced non-vulnerable users and the expected number of influenced vulnerable users. Then, we develop greedy heuristics and an approximation algorithm called *ISS* for our problem, which aim to find a set of seeds that maximizes the measure. We evaluate our methods on synthetic and real-world datasets and demonstrate that *ISS* substantially outperforms a heuristic competitor in terms of both effectiveness and efficiency while being more effective and/or efficient than the greedy heuristics.

1 Introduction

There has been an increased interest from the public and private sectors and organizations in leveraging social networks to spread information of adopting certain behavior (e.g., buying ipads or alcoholic beverages). A typical methodology of an organization is to influence a selected few users (*seeds*), through free gifts, discounts, and information sessions, to adopt the desirable behavior. The hope is that

Huiping Chen, Grigorios Loukides, Jiashi Fan
King's College London, e-mail: firstname.lastname@kcl.ac.uk

Hau Chan
University of Nebraska-Lincoln e-mail: hchan3@unl.edu

these seeds will influence other users in their social circles to adopt the same behavior, and the subsequent influenced users will influence others in their respective social circles. As the information propagates throughout the social network, eventually some number of users will adopt the desirable behavior.

As a result, the organization's goal is to solve the problem of selecting a set of k seeds which maximize the largest expected number of adoptions of *all the users* in the social network (*spread*). This problem is known as the *influence maximization* problem in social networks [10] and has been widely studied in the recent decade [14]. A main drawback of influence maximization is that it is unethical to influence users many of whom could be harmed due to their demographics, health conditions, or socioeconomic profile [7]. The users who could be harmed are referred to as *vulnerable* and are identified based on domain knowledge (e.g., user message content and sentiment analysis) [15, 21]. For example, when an organization aims to promote alcoholic beverages, it should avoid influencing users many of whom have drinking problems. Similarly, when it aims to promote junk food, it should avoid influencing users many of whom are overweight. This is important for performing socially responsible influence maximization [1], which benefits not only the vulnerable users but also the companies, because most users are often willing to pay more for products marketed in a socially responsible way [19]. Motivated by the presence of vulnerable users, we initiate the study of influence maximization in social networks with both vulnerable and non-vulnerable users. In particular, due to the diversity of social networks and that some vulnerable users will be influenced inadvertently, we consider the problem of limiting the influence to vulnerable users while maximizing the influence to the non-vulnerable users in social networks.

Contribution. Our work makes the following specific contributions.

(1) Influence Measure. To deal with influence maximization in our setting, we need a measure to quantify the quality of a set S of seeds (*seed-set*). The measure should ideally consider both vulnerable and non-vulnerable users, limit influencing users many of whom are vulnerable, and allow obtaining a seed-set with guaranteed quality. We examine the following natural measures and show that they are inappropriate to be used for influence maximization in our setting: (a) the difference $\sigma_{\mathcal{N}}(S) - \sigma_{\mathcal{V}}(S)$ and (b) the ratio $\frac{\sigma_{\mathcal{N}}(S)}{\sigma_{\mathcal{V}}(S)}$, where S is a seed-set and $\sigma_{\mathcal{N}}(S)$ and $\sigma_{\mathcal{V}}(S)$ is the expected number of influenced non-vulnerable users and vulnerable users, respectively. Then, we propose an *additive smoothing ratio* (*ASR*) measure $\frac{\sigma_{\mathcal{N}}(S)+c}{\sigma_{\mathcal{V}}(S)+c}$, where $c > 0$ is a specified constant. We show that *ASR* satisfies all the aforementioned properties and examine the impact of c in our influence maximization setting. Thus, our problem becomes finding a seed-set S of size at most k that maximizes *ASR*. This is a challenging problem because *ASR* is not monotone and neither submodular nor supermodular, which implies that it cannot be approximated through algorithms for submodular or supermodular maximization [4, 18, 23].

(2) Baseline Heuristics for Finding an ASR-Maximizing Seed-set. Since ASR is a ratio of submodular functions, we develop a natural greedy heuristic (*GR*) that finds a seed-set of size at most k and large *ASR* iteratively. In each iteration, *GR* selects as seed a non-vulnerable node which influences a large number of additional

non-vulnerable nodes for a small number of additional vulnerable nodes. We then develop GR_{MB}, a variation of *GR* that estimates the spread efficiently.

(3) Approximation Algorithm for Finding an ASR-Maximizing Seed-set. We design *ISS*, an efficient approximation algorithm for finding a seed-set to maximize *ASR*. Since *ASR* is not submodular, *ISS* cannot maximize it directly. Instead, *ISS* constructs three candidate seed-sets (one with *ASR*, another with a submodular lower bound function of *ASR*, and a third with a submodular upper bound function of *ASR*) and selects the best candidate seed-set. This is performed iteratively, with different bound functions that aim to increase the *ASR* of the final seed-set. Our experiments show that *ISS* outperforms a heuristic that is based on the difference $\sigma_{\mathcal{N}}(S) - \sigma_{\mathcal{V}}(S)$ [20] as well as *GR* and GR_{MB}, with respect to *ASR* and the spread of non-vulnerable and/or vulnerable nodes, while it is efficient and also scalable with respect to the seed-set size k.

2 Background

Submodular functions. Let U be a universe of elements and 2^U be its power set. A function $f : 2^U \to \mathbb{R}$ is *monotone*, if $f(X) \leq f(Y)$ for all subsets $X \subseteq Y \subseteq U$, and *non-monotone* otherwise. A function $f : 2^U \to \mathbb{R}$ is *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 $u \in U \setminus Y$ [12]. If the property holds with equality, then f is modular. A function $f : 2^U \to \mathbb{R}$ is *supermodular* if and only if $-f$ is submodular [12]. A modular function $f : 2^U \to \mathbb{R}$ is both submodular and supermodular. For brevity, we may write $f(X|u)$ for the marginal gain $f(X \cup \{u\}) - f(X)$.

Let $f : 2^U \to \mathbb{R}$ be a submodular function. For any $Y \subseteq U$, the *modular upper bound* $\widehat{f_Y}(X)$ of $f(X)$ is a modular function [9]

$$\widehat{f_Y}(X) = f(Y) + \sum_{u \in X \setminus Y} (f(\{u\}) - f(\{\})) - \sum_{u \in Y \setminus X} (f(Y) - f(Y \setminus \{u\})) \quad (1)$$

and the *modular lower bound* $\widecheck{f_{Y,\pi^Y}}(X)$ of $f(X)$ is a modular function [9]

$$\widecheck{f_{Y,\pi^Y}}(X) = \sum_{u \in X} f_{Y,\pi^Y}(u). \quad (2)$$

Y is referred to as the *parameter* of the bound, π^Y is a random permutation of the elements of Y (i.e., one-to-one mapping of Y onto itself), and

$$f_{Y,\pi^Y}(u) = \begin{cases} f(\pi^Y_u) - f(\pi^Y_{u^-}), & \text{if } u \in Y \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $\pi^Y_{u^-}$ is the prefix of π^Y comprised of all elements of π^Y that appear before u in π^Y, and π^Y_u is the prefix of π^Y comprised of all elements of $\pi^Y_{u^-}$ and u.

Independent Cascade (IC) model. To model influence, we consider the classical IC model [10, 20, 24]. The model views the social network as a weighted directed graph $G(V, E)$, where V and E is the set of nodes and edges of G, respectively. In our setting, V is partitioned into $\mathcal{N}$ and $\mathcal{V}$, comprised of all non-vulnerable and vulnerable nodes, respectively. We assume that $\mathcal{N} \neq \varnothing$, otherwise no seed can be selected, and that $\mathcal{V}$ is selected by the organization performing influence maximiza-

tion based on domain knowledge [15, 21]. The set of in-neighbors (respectively, out-neighbors) of a node u is denoted with $n^-(u)$ (respectively, $n^+(u)$), and its size is referred to as the *in-degree* (respectively, out-degree) of u. In the IC model, each newly activated node u' tries to activate each inactive out-neighbor $u \in n^+(u')$ once with probability $p((u', u))$, which is modeled as a weight of the edge (u', u) in G and is typically set to $\frac{1}{|n^-(u)|}$ [24]. If multiple nodes have the same out-neighbor, they all try to activate it in an arbitrary order independently. The diffusion process starts from a set S of nodes (*seeds*), which are active at time 0. Each seed tries to activate its out-neighbors at time 0, each activated out-neighbor stays active and tries to activate its own inactive out-neighbors at time 1, and the process proceeds similarly and ends when no new node becomes active. A seed-set S activates a node u with probability $P_S(u)$, and the spread of S over V, $\mathcal{N}$, and $\mathcal{V}$ is defined as $\sigma(S) = \sum_{u \in V} P_S(u)$, $\sigma_{\mathcal{N}}(S) = \sum_{u \in \mathcal{N}} P_S(u)$, and $\sigma_{\mathcal{V}}(S) = \sum_{u \in \mathcal{V}} P_S(u)$, respectively. For any seed-set S, $\sigma_{\mathcal{N}}(S)$ and $\sigma_{\mathcal{V}}(S)$ are monotone submodular functions [10]. We may omit the argument and value of $\sigma_{\mathcal{N}}$ and $\sigma_{\mathcal{V}}$ when it is clear from the context (e.g., write a seed-set with zero $\sigma_{\mathcal{N}}$ instead of a seed-set S with $\sigma_{\mathcal{N}}(S) = 0$).

3 Measures and Problem Definition

To study influence maximization in our setting, we need a measure that quantifies the quality of a seed-set and can be incorporated into methods to construct a high quality seed-set. The measure should favor a seed-set S that influences many non-vulnerable but few vulnerable nodes and also satisfy the following properties:

1. It should consider the influence of vulnerable and non-vulnerable nodes. In fact, we observed experimentally that constructing S based on only $\sigma_{\mathcal{N}}(S)$ (resp., $\sigma_{\mathcal{V}}(S)$) results in large $\sigma_{\mathcal{V}}(S)$ (resp., small $\sigma_{\mathcal{N}}(S)$), which is undesirable.
2. It should consider what fraction of all influenced users are vulnerable. This is important to penalize seed-sets that influence a large expected number of users many of whom are vulnerable.
3. It should allow constructing a seed-set with guaranteed quality (e.g., not "too far" from the optimal seed-set in the worst case) [10].

Natural Measures. A first measure is the difference $\sigma_{\mathcal{N}}(S) - \sigma_{\mathcal{V}}(S)$ given a seed-set S (i.e., the measure used in [20], with vulnerable nodes being treated as non-target nodes). This measure does not consider what fraction of all influenced users are vulnerable. Therefore, it may lead to constructing seed-sets with a large expected number of influenced users many of whom are vulnerable. For example, this measure would favor promoting an alcoholic beverage to 140 users out of whom 40 have drinking problems, instead of 59 users with no drinking problems, since $(140 - 40) - 40 > 59 - 0$. In addition, $\sigma_{\mathcal{N}}(S) - \sigma_{\mathcal{V}}(S)$ cannot be approximately maximized [9]. Thus, to construct a seed-set S, one has to settle with heuristics, such as [20], which offer no approximation guarantees.

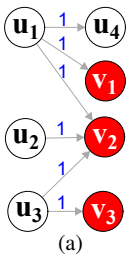
Another natural measure is the ratio $\frac{\sigma_{\mathcal{N}}(S)}{\sigma_{\mathcal{V}}(S)}$. The ratio considers what fraction of all influenced users are vulnerable, because it can be rewritten as $\frac{\sigma(S) - \sigma_{\mathcal{V}}(S)}{\sigma_{\mathcal{V}}(S)} =$

$\frac{\sigma(S)}{\sigma_{\mathcal{V}}(S)} - 1$ and the constant can be removed when it is maximized. However, it is undefined for every seed-set S with $\sigma_{\mathcal{V}}(S) = 0$ (i.e., S that does not influence vulnerable nodes). Thus, it cannot distinguish between any two seed-sets S_1, S_2 such that $\sigma_{\mathcal{V}}(S_1) = \sigma_{\mathcal{V}}(S_2) = 0$ and $\sigma_{\mathcal{N}}(S_1) > \sigma_{\mathcal{N}}(S_2)$ (e.g., it cannot favor promoting an alcoholic beverage to 59 users with no drinking problems vs. 2 users with no drinking problems) and also it is not clear how it can be approximately maximized. For example, the *GreedRatio* framework [3] for maximizing a ratio of two monotone submodular functions would result in a seed-set of unbounded size, which is not useful for influence maximization. The inverse ratio $\frac{\sigma_{\mathcal{V}}(S)}{\sigma_{\mathcal{N}}(S)}$ is defined for $\sigma_{\mathcal{V}}(S) = 0$ but it cannot be used to distinguish between the seed-sets S_1 and S_2 above, and it is equally difficult to minimize (minimizing it is equivalent to maximizing $\frac{\sigma_{\mathcal{N}}(S)}{\sigma_{\mathcal{V}}(S)}$). Thus, it cannot be used to find a seed-set with small or zero $\sigma_{\mathcal{V}}(S)$ and large $\sigma_{\mathcal{N}}(S)$, which helps our goal (to attract many users few of whom are vulnerable).

Our Proposed Measure. To retain the benefits of the ratio $\frac{\sigma_{\mathcal{N}}(S)}{\sigma_{\mathcal{V}}(S)}$, while fixing the issues caused by seed-sets that do not influence any vulnerable nodes, we apply additive smoothing [16] to the ratio. This leads to our additive smoothing ratio (*ASR*) measure, defined as $ASR(S, c) = \frac{\sigma_{\mathcal{N}}(S)+c}{\sigma_{\mathcal{V}}(S)+c}$, where S is a seed-set and $c > 0$ is a constant determined by the organization performing influence maximization. *ASR* is well defined (and larger than zero) when $\sigma_{\mathcal{V}}(S) = 0$. Furthermore, among the seed-sets S_1 and S_2 mentioned above, it favors the seed-set S_1, which influences a larger expected number of non-vulnerable nodes. In *ASR*, the constant c can be seen as a weight whose addition to $\sigma_{\mathcal{N}}(S)$ and to $\sigma_{\mathcal{V}}(S)$ changes their ratio and determines seed selection. The impact of c on seed selection will be discussed in Sections 4 and 6. Given our measure ASR, we define our influence maximization problem below.

Problem Definition. Given a graph G whose nodes are partitioned into $\mathcal{N}$ and $\mathcal{V}$ and parameters k and c, find a seed-set $S \subseteq \mathcal{N}$ of size at most k that maximizes $ASR(S, c)$.

Our problem is NP-hard (by reduction from the standard influence maximization problem [10]) and cannot be approximated using algorithms for submodular [4, 18] or supermodular [23] maximization. This is because *ASR* is non-monotone and is neither submodular nor supermodular, as we show below.



(a)

Seed-set S	$\sigma_{\mathcal{N}}(S)$	$\sigma_{\mathcal{V}}(S)$	$ASR(S,1)$
$S_0 = \{u_2, u_4\}$	2	1	3/2
$S_0 \cup \{u_1\}$	3	2	4/3
$S_1 = \{u_2, u_3, u_4\}$	3	2	4/3
$S_1 \cup \{u_1\}$	4	3	5/4
$S_2 = \{u_3\}$	1	2	2/3
$S_2 \cup \{u_2\}$	2	2	1
$S_3 = \{u_1, u_3\}$	3	3	1
$S_3 \cup \{u_2\}$	4	3	5/4

(b)

Fig. 1: (a) Example graph. $\mathcal{N} = \{u_1, \ldots, u_4\}$, $\mathcal{V} = \{v_1, v_2, v_3\}$, and each edge probability is equal to 1. (b) The spread over non-vulnerable nodes, the spread over vulnerable nodes, and *ASR* for different seeds-sets with $c = 1$.

Example 1. Consider the graph of Fig. 1a, whose set of nodes is partitioned into $\mathcal{N} = \{u_1, \ldots, u_4\}$ and $\mathcal{V} = \{v_1, v_2, v_3\}$, and the *ASR* of the seed-sets in Fig. 1b. $ASR(S, c)$ is: (I) *non-monotone*, because for $S_0 \subseteq S_0 \cup \{u_1\}$, $ASR(S_0, 1) = 3/2 > ASR(S_0 \cup \{u_1\}, 1) = 4/3$; (II) *not* submodular, because for $S_0 \subseteq S_1$ and $u_1 \in \mathcal{N} \setminus S_1$, $ASR(S_0 \cup \{u_1\}, 1) - ASR(S_0, 1) = -1/6 < ASR(S_1 \cup \{u_1\}, 1) - ASR(S_1, 1) = -1/12$, and

(III) *not* supermodular, because for $S_2 \subseteq S_3$ and $u_2 \in \mathcal{N} \setminus S_3$, $ASR(S_2 \cup \{u_2\}, 1) - ASR(S_2, 1) = 1/3 > ASR(S_3 \cup \{u_2\}, 1) - ASR(S_3, 1) = 1/4$. □

4 Baselines: Greedy Heuristics for Maximizing ASR

We explore two greedy baseline methods for constructing a seed-set S with size at most k and large $ASR(S, c)$. The first is, ***GR***, a natural heuristic for limiting the influence to vulnerable nodes. ***GR*** performs k iterations. In each iteration i (steps 3 to 6), it adds into the subset S_i the node u with the maximum ratio between: (I) the sum of the marginal gain in $\sigma_{\mathcal{N}}$, caused by adding u, and the constant c, and (II) the sum of the marginal gain in $\sigma_{\mathcal{V}}$, caused by adding u, and the constant c. Since ***ASR*** is non-monotone, a subset constructed in an iteration before i may have a larger ***ASR*** than S_i. Therefore, in step 7, ***GR*** considers the subsets constructed in all iterations and returns the one with the largest ***ASR***.

Algorithm: *GR* (GReedy heurisitc)
Input: $\mathcal{N} \subseteq V$, $\mathcal{V} \subseteq V$, graph G, parameter k, constant c
Output: Subset $S \subseteq \mathcal{N}$ of size $|S| \leq k$
1 $i \leftarrow 0$ `// Iteration counter`
2 $S_i \leftarrow \{\}$
3 **while** $i < k$ **do**
4 $u \in \underset{v \in \mathcal{N} \setminus \{S_i\}}{\arg\max} \frac{\sigma_{\mathcal{N}}(S_i|v) + c}{\sigma_{\mathcal{V}}(S_i|v) + c}$
5 $S_{i+1} \leftarrow S_i \cup \{u\}$
6 $i \leftarrow i + 1$
7 **return** $S \leftarrow \underset{S' \in \{S_1, \ldots, S_k\}}{\arg\max} ASR(S', c)$

We now discuss how ***GR*** deals with a non-vulnerable node v that influences no vulnerable nodes. Adding v into S_i makes the objective function of ***GR*** equal to $\frac{\sigma_{\mathcal{N}}(S_i|v)+c}{c}$ (see step 4), since S_i does not influence more vulnerable nodes after the addition of v (i.e., $\sigma_{\mathcal{V}}(S_i|v) = 0$). If $\sigma_{\mathcal{N}}(S_i|v)$ is small, it is better to add a different node v' which influences few vulnerable nodes but "through" these vulnerable nodes reaches out to many more non-vulnerable nodes than v. In fact, ***GR*** adds v' instead of v if $\frac{\sigma_{\mathcal{N}}(S_i|v')+c}{\sigma_{\mathcal{V}}(S_i|v')+c} > \frac{\sigma_{\mathcal{N}}(S_i|v)+c}{c}$, and uses the parameter c to control the bias towards nodes such as v', which influence a small number of vulnerable nodes but many more non-vulnerable nodes than u, as shown in Example 2 and experimentally in Section 6.

Example 2. In iteration $i = 0$, the non-vulnerable nodes u_1 to u_4 in Table 1a are considered and the node $u \in \{u_1, \ldots, u_4\}$ with the largest $\frac{\sigma_{\mathcal{N}}(S_0|u)+c}{\sigma_{\mathcal{V}}(S_0|u)+c}$ is added into $S_0 = \{\}$. As shown in Table 1b, c determines the added node. For $c = 0.01$, u_1 that influences no vulnerable and few non-vulnerable nodes is added, for $c = 1$, u_3 that influences one vulnerable and many non-vulnerable nodes is added, and for $c = 10$, u_4 that influences more vulnerable and non-vulnerable nodes than u_2 is added. □

Node	u_1	u_2	u_3	u_4
$\sigma_{\mathcal{V}}$	0	0.01	1	10
$\sigma_{\mathcal{N}}$	3	5	150	300

(a)

c	0.01	0.02	1	10
Added Node	u_1	u_2	u_3	u_4

(b)

Table 1: (a) Non-vulnerable nodes u that are considered for addition into $S_0 = \{\}$, and the expected number of vulnerable and non-vulnerable nodes they influence. (b) The node that is added into S_0 for different values of c.

To improve the efficiency of ***GR***, we propose a variant, ***GR***$_{MB}$ (MB is for MIA Batch-update). Unlike ***GR*** which computes spread exactly by adapting the method of [8] to the IC model, ***GR***$_{MB}$ estimates the spread efficiently using the MIA (Maximum Influence Arborescence) method

[24]. MIA estimates the probability $P_S(u)$ for a node u and seed-set S based on the union of paths from S that have the highest probability to influence u, instead of all paths. Consequently, *GR*$_{MB}$ is two orders of magnitude faster on average than *GR*.

5 The *ISS* Approximation Algorithm for Maximizing *ASR*

This section presents *ISS* (Iterative Subsample with Spread bounds), starting from the bound functions of *ASR* that *ISS* employs.

Lower and upper bound function of *ASR*. $ASR(S, c)$ is non-monotone non-submodular for any subset S (see Section 3) and, thus, it is difficult to approximate directly. Our *ISS* algorithm finds a seed-set S with approximately maximum $ASR(S, c)$, using two submodular functions $ASR^{\mathbf{L}}$ and $ASR^{\mathbf{U}}$ that bound *ASR* from below and from above, respectively. These functions are defined as follows:

$$ASR^{\mathbf{L}}(S, c, Y) = \frac{\sigma_{\mathcal{N}}(S) + c}{\widehat{\sigma_{\mathcal{V},Y}}(S) + c} = \frac{\sigma_{\mathcal{N}}(S) + c}{\sigma_{\mathcal{V}}(Y) + \sum\limits_{u \in S \setminus Y} \sigma_{\mathcal{V}}(\{u\}) + \sum\limits_{u \in Y \setminus S} (\sigma_{\mathcal{V}}(Y) - \sigma_{\mathcal{V}}(Y \setminus \{u\})) + c}$$

$$ASR^{\mathbf{U}}(S, c, \pi^Y) = \frac{\sigma_{\mathcal{N}}(S) + c}{\widehat{\sigma_{\mathcal{V},\pi^Y}}(S) + c} = \frac{\sigma_{\mathcal{N}}(S) + c}{\sum\limits_{u \in S} (\sigma_{\mathcal{V},Y,\pi^Y}(u)) + c}$$

where $Y \subseteq \mathcal{N}$ is the *parameter* in each bound function, and

$$\sigma_{\mathcal{V},Y,\pi^Y}(u) = \begin{cases} \sigma_{\mathcal{V}}(\pi^Y_u) - \sigma_{\mathcal{V}}(\pi^Y_{u^-}) & \text{, if } u \in Y \\ 0 & \text{, otherwise} \end{cases}$$

$ASR^{\mathbf{L}}$ is obtained by replacing $\sigma_{\mathcal{V}}(S)$ in *ASR* with its modular upper bound $\widehat{\sigma_{\mathcal{V},Y}}(S)$ (see Eq. 1) and using the fact that $\sigma_{\mathcal{V}}(\{\}) = 0$. $ASR^{\mathbf{U}}$ is obtained by replacing $\sigma_{\mathcal{V}}(S)$ in *ASR* with its modular lower bound $\widehat{\sigma_{\mathcal{V},\pi^Y}}(S)$ (see Eq. 2).

$ASR^{\mathbf{L}}$, as well as $ASR^{\mathbf{U}}$, is submodular with respect to a seed-set S, because: (a) Its numerator is monotone submodular, as a sum of the monotone submodular function $\sigma_{\mathcal{N}}(S)$ and the constant c [12], (b) its denominator is a modular function, as a sum of a modular bound function and the constant c, and (c) the ratio between a submodular function and a modular function is clearly submodular (see Section 2). However, $ASR^{\mathbf{L}}$, as well as $ASR^{\mathbf{U}}$, is non-monotone, as shown in Example 3.

Example 3. (continuing from Example 1) Let $c = 1$ and $S' = \{\}$. Since $ASR^{\mathbf{L}}(S_1, 1, S') = 4/4 > ASR^{\mathbf{L}}(S_1 \cup \{u_1\}, 1, S') = 5/6$, $ASR^{\mathbf{L}}$ is non-monotone. Let $S'' = \{u_2, u_3\}$ and its permutation $\pi^{S''} = (u_3, u_2)$. Since $ASR^{\mathbf{U}}(S_3, 1, \pi^{S''}) = 4/3 > ASR^{\mathbf{U}}(S_3 \cup \{u_2\}, 1, \pi^{S''}) = 5/4$, $ASR^{\mathbf{U}}$ is non-monotone.

***ISS* algorithm.** The algorithm works iteratively, as can be seen from the pseudocode. In each iteration, it creates a seed-set S_{cur} in three phases: (a) dummy element creation, (b) construction of three candidate seed-sets (one using *ASR*, a second using $ASR^{\mathbf{L}}$ and a third using $ASR^{\mathbf{U}}$), and (c) selection of the best candidate seed-set and removal of dummy elements from it. The iterations stop when S_{cur} is not better than the previously created seed-set S_{pr} in terms of *ASR* (steps 21-22). This guarantees that the algorithm terminates [9].

Algorithm: *ISS* (Iterative Subsample with Spread bounds)
Input: Set of non-vulnerable nodes $\mathcal{N} \subseteq V$, set of vulnerable nodes $\mathcal{V} \subseteq V$, graph G, parameter k, constant c
Output: Subset $S \subseteq \mathcal{N}$ of size $|S| \leq k$

1 $S_{pr} \leftarrow \{\}$
2 $S_{cur} \leftarrow \mathcal{N}$
3 **while** *true* **do**
`// Phase I`
4 $\mathcal{D} \leftarrow$ set of k dummy elements $\{u_1, \ldots, u_k\}$ such that, for each element u_i, $i \in [1, k]$, and every set $S \subseteq \mathcal{N}$: $\sigma_{\mathcal{N}}(S \cup \{u_i\}) = \sigma_{\mathcal{N}}(S)$ and $\sigma_{\mathcal{V}}(S \cup \{u_i\}) = \sigma_{\mathcal{V}}(S)$
5 $\mathcal{N}' \leftarrow \mathcal{N}$
6 **while** $\frac{|\mathcal{N}'|}{k}$ *is not an integer* **do**
7 Add into $\mathcal{N}'$ a dummy element $u' \notin \mathcal{D}$ such that $\sigma_{\mathcal{N}}(S \cup \{u'\}) = \sigma_{\mathcal{N}}(S)$ and $\sigma_{\mathcal{V}}(S \cup \{u'\}) = \sigma_{\mathcal{V}}(S)$
`// Phase II`
8 $i \leftarrow 0$; $S_0^{\mathbf{O}} \leftarrow \{\}$; $S_0^{\mathbf{L}} \leftarrow \{\}$; $S_0^{\mathbf{U}} \leftarrow \{\}$
9 **while** $i < k$ **do**
10 $\mathcal{R} \leftarrow$ uniform random sample of $\mathcal{N}'$ with $\frac{|\mathcal{N}'|}{k}$ elements
11 Add into $\mathcal{R}$ a random element from $\mathcal{D}$
12 $u^{\mathbf{O}} \in \arg\max_{u \in \mathcal{R}} (ASR(S_i^{\mathbf{O}} \cup \{u\}, c) - ASR(S_i^{\mathbf{O}}, c))$
13 $S_{i+1}^{\mathbf{O}} \leftarrow S_i^{\mathbf{O}} \cup \{u^{\mathbf{O}}\}$
14 $u^{\mathbf{L}} \in \arg\max_{u \in \mathcal{R}} (ASR^{\mathbf{L}}(S_i^{\mathbf{L}} \cup \{u\}, c, S_{pr}) - ASR^{\mathbf{L}}(S_i^{\mathbf{L}}, c, S_{pr}))$
15 $S_{i+1}^{\mathbf{L}} \leftarrow S_i^{\mathbf{L}} \cup \{u^{\mathbf{L}}\}$
16 $u^{\mathbf{U}} \in \arg\max_{u \in \mathcal{R}} (ASR^{\mathbf{U}}(S_i^{\mathbf{U}} \cup \{u\}, c, \pi^{S_{pr}}) - ASR^{\mathbf{U}}(S_i^{\mathbf{U}}, c, \pi^{S_{pr}}))$
17 $S_{i+1}^{\mathbf{U}} \leftarrow S_i^{\mathbf{U}} \cup \{u^{\mathbf{U}}\}$
18 $i \leftarrow i + 1$
`// Phase III`
19 $S_{cur} \leftarrow \arg\max_{S \in \{S_k^{\mathbf{O}}, S_k^{\mathbf{L}}, S_k^{\mathbf{U}}\}} ASR(S, c)$
20 $S_{cur} \leftarrow$ Remove all dummy elements from S_{cur}
21 **if** $ASR(S_{cur}, c) \leq ASR(S_{pr}, c)$ **then**
22 **break**
23 $S_{pr} \leftarrow S_{cur}$
24 **return** S_{cur}

Phase I (steps 4 to 7): A set $\mathcal{D}$ of k *dummy* elements whose addition into any seed-set S does not change $\sigma_{\mathcal{N}}(S)$ and $\sigma_{\mathcal{V}}(S)$, are created. Then, a dummy element $u' \notin \mathcal{D}$ is added into a subset $\mathcal{N}'$ (initially containing all non-vulnerable nodes), until $\frac{|\mathcal{N}'|}{k}$ is an integer.

Phase II (steps 8 to 18): A random sample of $\frac{|\mathcal{N}'|}{k}$ elements from $\mathcal{N}'$ and a dummy element is created. Next, the candidate subset $S_i^{\mathbf{O}}$, $S_i^{\mathbf{U}}$ and $S_i^{\mathbf{L}}$ is extended with a node in the sample causing the largest marginal gain with respect to *ASR*, $ASR^{\mathbf{L}}$, and $ASR^{\mathbf{U}}$, respectively. The parameter of $ASR^{\mathbf{L}}$ is the seed-set S_{pr}, constructed in the previous iteration and that of $ASR^{\mathbf{U}}$ is a random permutation $\pi^{S_{pr}}$ of S_{pr}.

Phase III (steps 19 to 23): The best candidate subset with respect to *ASR* is selected as S_{cur}, and all dummy elements are removed from it. If S_{cur} is not better than S_{pr} in terms of *ASR*, the while loop in step 3 is terminated and S_{cur} is returned. Otherwise, another iteration is performed with the aim of generating a seed-set with larger *ASR*, due to the use of different (and often better [9]) bounds.

Theorem 1. ISS *constructs a seed-set S such that:*

$$\mathbb{E}[ASR(S,c)] \geq \max\left(\frac{\sigma_{\mathcal{V}}(S^*)+c}{\widehat{\sigma_{\mathcal{V},S_{pr}}}(S^*)+c}, \frac{c}{c+k\cdot\max_{u\in\mathcal{N}}\widehat{\sigma_{\mathcal{V},S_{pr}}}(\{u\})}\right)\cdot\frac{1}{\mathrm{e}}\cdot(1-\frac{1}{\mathrm{e}})\cdot ASR(S^*,c)$$

where $S^* = \arg\max_{S\subseteq\mathcal{N},|S|\leq k} ASR(S,c)$, $\widehat{\sigma_{\mathcal{V},S_{pr}}}$ is the modular upper bound used in $ASR^{\mathbf{L}}$ (step 14) in the last iteration of ISS*, and the expectation is over every possible S constructed by ISS.*

Proof. Let $S_k^{\mathbf{L},j}$ (respectively, S_{pr_j}) denote the subset $S_k^{\mathbf{L}}$ (respectively, S_{pr}) in step 19 of an iteration j of *ISS* (j-th execution of the while loop in step 3). Since $ASR^{\mathbf{L}}$ bounds ASR from below, we have $ASR(S_k^{\mathbf{L},j}, c) \geq ASR^{\mathbf{L}}(S_k^{\mathbf{L},j}, c, S_{pr_j})$. Also, from the monotonicity of expectation, this inequality can be written as

$$\mathbb{E}[ASR(S_k^{\mathbf{L},j}, c)] \geq \mathbb{E}[ASR^{\mathbf{L}}(S_k^{\mathbf{L},j}, c, S_{pr_j})], \tag{4}$$

where each expectation is over every $S_k^{\mathbf{L},j}$. We now observe that

$$\mathbb{E}[ASR^{\mathbf{L}}(S_k^{\mathbf{L},j}, c, S_{pr_j})] \geq \frac{1}{\mathrm{e}} \cdot (1 - \frac{1}{\mathrm{e}}) \cdot ASR^{\mathbf{L}}(S^{*,j}, c, S_{pr_j}), \tag{5}$$

where $S^{*,j} = \arg\max_{S \subseteq \mathcal{N}, |S| \leq k} ASR^{\mathbf{L}}(S, c, S_{pr_j})$. Eq. 5 holds because $S_k^{\mathbf{L},j}$ is constructed based on the *Sub-sample Greedy* algorithm [17] with $ASR^{\mathbf{L}}$ in each iteration (execution of the while loop in step 3) of *ISS*. Thus, we obtain:

$$\begin{aligned}
\mathbb{E}[ASR(S_k^{\mathbf{L},j}, c)] \geq & \frac{1}{\mathrm{e}} \cdot (1 - \frac{1}{\mathrm{e}}) \cdot ASR^{\mathbf{L}}(S^{*,j}, c, S_{pr_j}) \\
\geq & \frac{1}{\mathrm{e}} \cdot (1 - \frac{1}{\mathrm{e}}) \cdot \frac{\sigma_{\mathcal{N}}(S^{*,j}) + c}{\widehat{\sigma_{\mathcal{V},S_{pr_j}}}(S^{*,j}) + c} \cdot \left[\frac{\sigma_{\mathcal{N}}(S^{*,j}) + c}{\sigma_{\mathcal{V}}(S^{*,j}) + c} \cdot \frac{\sigma_{\mathcal{V}}(S^{*,j}) + c}{\sigma_{\mathcal{N}}(S^{*,j}) + c} \right] \\
\geq & \frac{\sigma_{\mathcal{V}}(S^{*,j}) + c}{\widehat{\sigma_{\mathcal{V},S_{pr_j}}}(S^{*,j}) + c} \cdot \frac{1}{\mathrm{e}} \cdot (1 - \frac{1}{\mathrm{e}}) \cdot ASR(S^{*,j}, c).
\end{aligned}$$

The first inequality holds from Eqs. 4 and 5, the second from the definition of $ASR^{\mathbf{L}}$ and because we multiply by 1 (in square brackets), and the third by the definition of *ASR*. Since the third inequality holds for every iteration j of *ISS*:

$$\mathbb{E}[ASR(S^{\mathbf{L}}, c)] \geq \frac{\sigma_{\mathcal{V}}(S^*) + c}{\widehat{\sigma_{\mathcal{V},S_{pr}}}(S^*) + c} \cdot \frac{1}{\mathrm{e}} \cdot (1 - \frac{1}{\mathrm{e}}) \cdot ASR(S^*, c), \tag{6}$$

where $S^{\mathbf{L}}$ is the subset $S_k^{\mathbf{L},j}$ constructed in step 19 of the last iteration of *ISS*.

Let $S_k^{\mathbf{U},j}$ (respectively, S_{pr_j}) denote the subset $S_k^{\mathbf{U}}$ (respectively, S_{pr}) in step 19 of an iteration j of *ISS* (j-th execution of the while loop in step 3). We first prove:

$$\widehat{\sigma_{\mathcal{V},S_{pr_j}}}(S_k^{\mathbf{U},j}) \leq \sum_{u \in S_k^{\mathbf{U},j}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\}) \leq k \cdot \max_{u \in S_k^{\mathbf{U},j}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\}) \leq k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\}) \tag{7}$$

The first inequality holds because $\widehat{\sigma_{\mathcal{V},S_{pr_j}}}$ is submodular, the second because $|S_k^{\mathbf{U},j}| \leq k$ and $\widehat{\sigma_{\mathcal{V},S_{pr_j}}}$ is non-negative, and the third because $S_k^{\mathbf{U},j} \subseteq \mathcal{N}$. From Eq. 7 and $\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) \leq \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(S_k^{\mathbf{U},j})$ (as $\widehat{\sigma_{\mathcal{V},S_{pr_j}}}$ bounds $\sigma_{\mathcal{V}}$ from above), we get: $\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) + c \leq c + k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})$, which implies:

$$\frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c}{\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) + c} \geq \frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c}{c + k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})} \geq \frac{c}{c + k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})}. \tag{8}$$

The second inequality is because $\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) \geq 0$ by definition. $ASR(S_k^{\mathbf{U},j}, c) = \frac{\sigma_{\mathcal{N}}(S_k^{\mathbf{U},j}) + c}{\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) + c} \cdot \frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c} = \frac{\sigma_{\mathcal{N}}(S_k^{\mathbf{U},j}) + c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c} \cdot \frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c}{\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) + c}$, which implies:

$$\begin{aligned}
\mathbb{E}[ASR(S_k^{\mathbf{U},j}, c)] = & \mathbb{E}[\frac{\sigma_{\mathcal{N}}(S_k^{\mathbf{U},j}) + c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c} \cdot \frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c}{\sigma_{\mathcal{V}}(S_k^{\mathbf{U},j}) + c}] \\
\geq & \mathbb{E}[\frac{\sigma_{\mathcal{N}}(S_k^{\mathbf{U},j}) + c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j}) + c} \cdot \frac{c}{c + k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})}]
\end{aligned}$$

from Eq. 8, where the expectation is over each $S_k^{\mathbf{U},j}$. Also, $\lambda_j = \frac{c}{c + k \cdot \max_{u \in \mathcal{N}} \widehat{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})}$ is constant in iteration j, so

$$\mathbb{E}[ASR(S_k^{\mathbf{U},j},c)] \geq \lambda_j \cdot \mathbb{E}[\frac{\sigma_{\mathcal{N}}(S_k^{\mathbf{U},j})+c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S_k^{\mathbf{U},j})+c}] = \lambda_j \cdot E[ASR^{\mathbf{U}}(S_k^{\mathbf{U},j},c,\pi^{S_{pr_j}})]. \quad (9)$$

We now observe that

$$\mathbb{E}[ASR^{\mathbf{U}}(S_k^{\mathbf{U},j},c,\pi^{S_{pr_j}})] \geq \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{\mathrm{e}}) \cdot ASR^{\mathbf{U}}(S^{*,j},c,\pi^{S_{pr_j}}), \quad (10)$$

where $S^{*,j} = \arg\max_{S\subseteq\mathcal{N},|S|\leq k} ASR^{\mathbf{U}}(S,c,\pi^{S_{pr_j}})$. Eq. 10 holds because $S_k^{\mathbf{U},j}$ is constructed based on the *Sub-sample Greedy* algorithm [17] with $ASR^{\mathbf{U}}$ in each iteration (execution of the while loop in step 3) of *ISS*. Thus, we obtain:

$$\begin{aligned}
\mathbb{E}[ASR(S_k^{\mathbf{U},j},c)] \geq & \lambda_j \cdot \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{\mathrm{e}}) \cdot ASR^{\mathbf{U}}(S^{*,j},c,\pi^{S_{pr_j}}) \\
\geq & \lambda_j \cdot \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{\mathrm{e}}) \cdot \frac{\sigma_{\mathcal{N}}(S^{*,j})+c}{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S^{*,j})+c} \cdot [\frac{\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}(S^{*,j})+c}{\sigma_{\mathcal{V}}(S^{*,j})+c}] \\
\geq & \frac{c}{c+k\cdot\max_{u\in\mathcal{N}}\widetilde{\sigma_{\mathcal{V},S_{pr_j}}}(\{u\})} \cdot \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{\mathrm{e}}) \cdot ASR(S^{*,j},c)
\end{aligned}$$

The first inequality holds from Eqs. 9 and 10, the second because the term in square brackets is at most 1 (since $\widetilde{\sigma_{\mathcal{V},\pi^{S_{pr_j}}}}$ bounds $\sigma_{\mathcal{V}}$ from below), and the third from the definition of λ_j and *ASR*. Since the third inequality holds for each iteration j of *ISS*, we obtain:

$$\mathbb{E}[ASR(S^{\mathbf{U}},c)] \geq \frac{c}{c+k\cdot\max_{u\in\mathcal{N}}\widetilde{\sigma_{\mathcal{V},S_{pr}}}(\{u\})} \cdot \frac{1}{\mathrm{e}} \cdot (1-\frac{1}{\mathrm{e}}) \cdot ASR(S^*,c) \quad (11)$$

where $S^{\mathbf{U}}$ is the subset $S_k^{\mathbf{U},j}$ constructed in step 19 of the last iteration of *ISS*. The proof follows from Eqs. 6 and 11 and step 19 in *ISS*. □

Although the guarantee in Theorem 1 holds irrespectively of the number of iterations of *ISS*, we observed, in our experiments, that more iterations result in seed-sets with larger *ASR*. This is because the parameter S_{pr} and $\pi^{S_{pr}}$ of the bound functions $ASR^{\mathbf{L}}$ and $ASR^{\mathbf{U}}$, respectively, is updated in every iteration which often improves the bounds [9]. We also observed that *ISS* needed at most 4 iterations to terminate. *ISS* is faster than *GR* and scales better with respect to k, because it selects seeds from a sample of $\mathcal{N}$ of size approximately $\frac{|\mathcal{N}|}{k}$, instead of the entire set $\mathcal{N}$, and because it performs a small number of iterations.

6 Experimental Evaluation

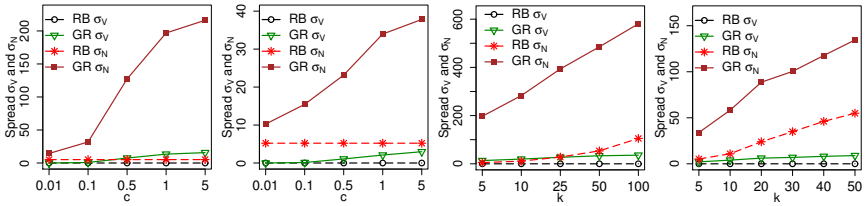
In this section, we evaluate *GR*, GR_{MB}, and *ISS*, in terms of effectiveness and efficiency, by comparing them against *TIM* [20], a heuristic for finding a seed-set S with size at most k and large $\sigma_{\mathcal{N}}(S)-\sigma_{\mathcal{V}}(S)$, and two baselines that employ *Greedy* [18]: *RB*, which applies *Greedy* [18] to the subset of non-vulnerable nodes that do not influence vulnerable nodes, and *RB′*, which applies *Greedy* with the objective function $\sigma_{\mathcal{N}}$. *RB* creates a seed-set S with $\sigma_{\mathcal{V}}(S)=0$ and was used to see whether S can have large $\sigma_{\mathcal{N}}(S)$. *RB′* creates a seed-set S with large $\sigma_{\mathcal{N}}(S)$ and was used to see whether S can have small $\sigma_{\mathcal{V}}(S)$. *RB′* found seed-sets that influenced many more vulnerable nodes than those of all other methods, thus, we omit its results.

Dataset	# of nodes ($\lvert V\rvert$)	# of edges ($\lvert E\rvert$)	avg in-degree	max in-degree	# of vuln. nodes ($\lvert\mathcal{V}\rvert$)	θ
WI	7115	103689	13.7	452	100	0.01
TW	235	2479	10.5	52	25	0.01
POL	1490	19090	11.9	305	100	0.003
AB	840	10008	11.9	137	10	0.01

Table 2: Characteristics of datasets.

All algorithms were implemented in C++ and applied to the *Wiki-vote* (*WI*), Twitter (*TW*), and

PolBlogs (*POL*) datasets (see Table 2). *POL* is available at http://www-personal.umich.edu/ mejn/ and all other datasets at http://snap.stanford.edu/data. We also used synthetic datasets, generated by the Albert-Barabasi model, as in [15], with a varying number of edges in $[500, 10000]$. We refer to the dataset with 10000 edges as *AB*. We set $p(u', u) = \frac{1}{|n^-(u)|}$ for each edge (u', u) as in [5, 15]. We also set the maximum probability threshold for a path to $\theta = 0.01$, so that all methods achieve a good accuracy/efficiency trade-off by discarding paths that have low probability to influence a node, as in [6]. The default value for k was 5 and for c was 1. The vulnerable nodes were selected randomly. To improve the efficiency of *ISS*, we used the CELF optimization [10] for the submodular bound functions (steps 14 and 16). Also, the results for *ISS* were averaged over 10 runs. All experiments ran on an Intel Xeon CPU E5-2640 @2.66GHz with 64GB RAM. Due to space limitations, we omit some results that were qualitatively similar to the reported ones (e.g., results for varying $|\mathcal{V}|$ in WI).



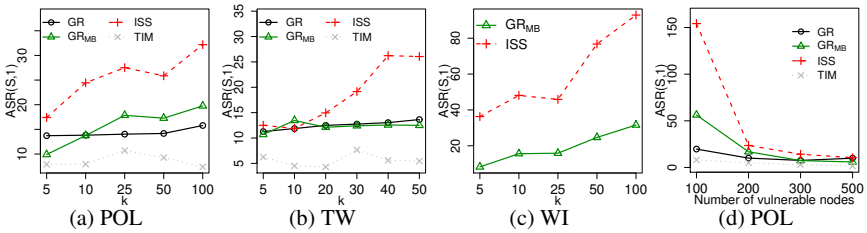
(a) POL (b) TW (c) POL (d) TW

Fig. 2: Spread of vulnerable and non-vulnerable nodes: (a) POL vs c, (b) TW vs c, (c) POL vs k, and (d) TW vs k.

Comparison to *RB*. *GR* constructs seed-sets that influence at least 5.5 and up to 38 times more non-vulnerable nodes than those constructed by *RB*, for different values of c (see Figs. 2a and 2b) and k (see Figs. 2c and 2d). The reason is that, for all c and k values, vulnerable nodes were distributed across the graph. So, the seed-sets constructed by *RB* that did not influence vulnerable nodes did not influence many non-vulnerable nodes, while those constructed by *GR* influenced a small number of vulnerable nodes but could reach to and influence many more non-vulnerable nodes. Moreover, *TIM*, GR_{MB}, and *ISS* outperformed *RB* (the results for them are omitted). Thus, in all subsequent experiments, we omit results for *RB*, since it does not construct practically useful solutions and set $c = 1$ because this allows constructing seed-sets with good $\sigma_{\mathcal{N}}/\sigma_{\mathcal{V}}$ trade-off.

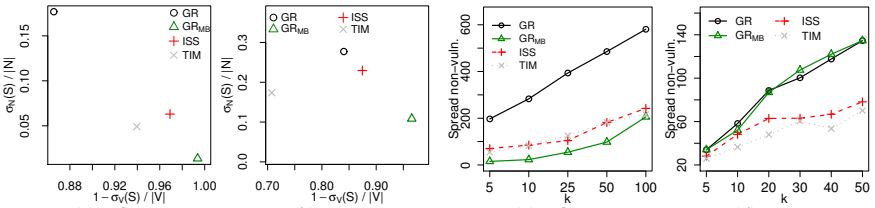
***ASR* with $c = 1$.** All our algorithms substantially outperform *TIM* in terms of *ASR* for varying k (see Figs. 3a, 3b, and 3c) and varying number of vulnerable nodes $|\mathcal{V}|$ (see Fig. 3d). *ISS* outperformed all other methods, being 3, 1.7, and 2 times better than *TIM*, *GR*, and GR_{MB} on average (over all datasets and k values), respectively. *ISS* was also 8.9, 3.3, and 1.9 times better than *TIM*, *GR*, and GR_{MB} on average (over all $|\mathcal{V}|$ values in Fig. 3d), respectively. We omit the results for *GR* and *TIM* for the largest dataset *WI* from all subsequent experiments, since *GR* and *TIM* did not finish within 3 days.

Spread of Vulnerable and Non-vulnerable Nodes. We demonstrate that all our algorithms substantially outperform *TIM* in terms of $\sigma_{\mathcal{N}}$ and/or $\sigma_{\mathcal{V}}$. First, we report Figs. 4a and 4b, where each point (x, y) corresponds to the values



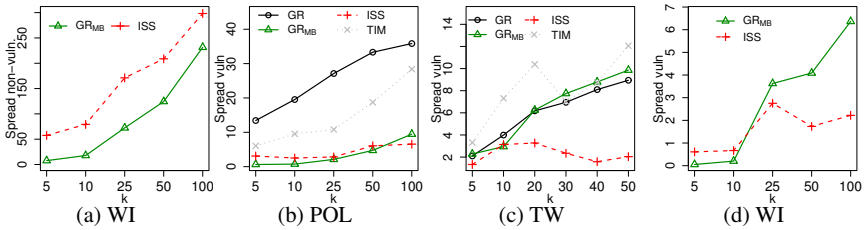
(a) POL (b) TW (c) WI (d) POL

Fig. 3: *ASR* with $c = 1$ vs k for (a) POL, (b) TW, and (c) WI. (d) *ASR* with $c = 1$ vs $|\mathcal{V}|$.



(a) POL (b) TW (c) POL (d) TW

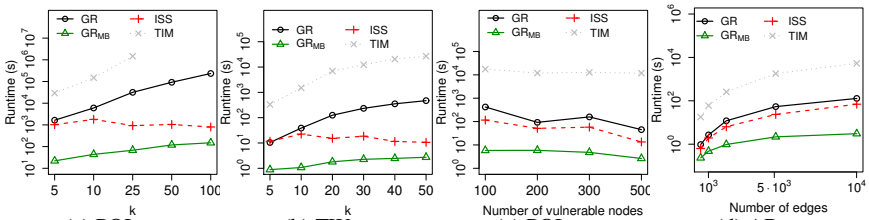
Fig. 4: Map for (a) POL and $k = 5$, (b) TW and $k = 10$. Spread of non-vulnerable nodes vs k for (c) POL, and (d) TW.



(a) WI (b) POL (c) TW (d) WI

Fig. 5: Spread of non-vulnerable nodes vs k for (a) TW, and (b) WI. Spread of vulnerable nodes vs k for (c) POL. Spread of vulnerable nodes for (d) TW, and (e) WI.

$(1 - \frac{\sigma_{\mathcal{V}}(S)}{|\mathcal{V}|}, \frac{\sigma_{\mathcal{N}}(S)}{|\mathcal{N}|})$, referred to as *protection* and *utility* of a seed-set S. *ISS* outperformed *TIM* with respect to both protection and utility, achieving overall better protection than *GR* and better utility than GR_{MB}. We also report $\sigma_{\mathcal{N}}$ and $\sigma_{\mathcal{V}}$ in Figs. 4c to 5d. *GR* and *TIM* constructed seed-sets that influence too many vulnerable nodes. GR_{MB} performed inconsistently (e.g., its seed-sets influenced few vulnerable nodes in Fig. 5b and too many vulnerable nodes in Fig. 5c). *ISS* influenced few vulnerable nodes and a moderate number of non-vulnerable nodes, achieving a good $\sigma_{\mathcal{N}}/\sigma_{\mathcal{V}}$ trade-off.



(a) POL (b) TW (c) POL (d) AB

Fig. 6: Runtime vs k for (a) POL, and (b) TW. Runtime vs (c) number of vulnerable nodes for POL, and (d) number of edges for AB.

Efficiency. All our methods are much faster than *TIM* for varying k (see Figs. 6a and 6b). *TIM* required 10 hours when $k = 50$ in the case of TW which only has 235 nodes, and 17 days when $k = 25$ in the case of POL. *GR* was faster but did not terminate within 3 days in the case of WI, and GR_{MB} was the fastest due to its efficient spread estimation function [24]. *ISS* was significantly faster than *GR* and *TIM* and the most scalable method with respect to k. Fig. 6c shows the runtime for varying $|\mathcal{V}|$. All our algorithms become faster with $|\mathcal{V}|$, since fewer nodes can be selected as seeds and are at least three orders of magnitude faster than *TIM* on average. Fig. 6d shows the runtime for varying number of edges. Our algorithms were faster than *TIM* by up to three orders of magnitude.

7 Related work

No existing work addresses influence maximization when there are vulnerable nodes. The most related works are [20] and [15]. [20] aims to maximize the difference between the expected number of influenced users who belong to a target group and the expected number of all other influenced users. Our work differs from [20] along three dimensions. First, [20] can select target nodes as seeds, but we cannot do the same for vulnerable nodes, as this would harm them. Second, our *ASR* measure has desired properties unlike the measure $\sigma_{\mathcal{N}}(S) - \sigma_{\mathcal{V}}(S)$ in [20] (see Section 3). Third, our methods are substantially more effective and efficient than the heuristic in [20]. [15] is applied after influence maximization (i.e., considers a given seed-set) and seeks to delete edges in order to limit the activation probability of vulnerable nodes in the Linear Threshold model [10]. Thus, it is orthogonal to our work.

There are many works on targeted viral marketing (e.g., [11, 13, 20, 22, 25]). For example, [13] considered influence maximization when each target node has a constant profit, and [22] considered the impact of the location and login time of target nodes. Unlike ours, the works in [11,13,20,22,25] do not consider vulnerable nodes.

There are also works on influence maximization considering nodes with negative impact on the influence diffusion process [2, 5]. [5] studied influence maximization under a model where each node can diffuse information of opposite content to the information that is being spread from the seed-set. [2] studied influence maximization, when some nodes reject the diffused information. Different from these works, no node negatively impacts the influence diffusion process in our approach.

8 Conclusion

In this paper, we study influence maximization when there are vulnerable nodes. We first propose a measure for limiting the influence to vulnerable nodes, which is obtained by applying additive smoothing to the ratio between the expected number of influenced non-vulnerable nodes and the expected number of influenced vulnerable nodes. Based on the measure, we define a new influence maximization problem that seeks to find a seed-set of size at most k that maximizes the measure. We propose two greedy baseline heuristics, and the *ISS* approximation algorithm to solve our influence maximization problem. We evaluate our methods on synthetic and real-

world datasets and show that *ISS* outperforms the method of [20] and our baselines in terms of effectiveness and efficiency.

References

1. http://www.conecomm.com/research-blog/2017-csr-study
2. Abebe, R., Adamic, L., Kleinberg, J.: Mitigating overexposure in viral marketing (2018)
3. Bai, W., Iyer, R., Wei, K., Bilmes, J.: Algorithms for optimizing the ratio of submodular functions. In: ICML. pp. 2751–2759 (2016)
4. Buchbinder, N., Feldman, M., Naor, J., Schwartz, R.: Submodular maximization with cardinality constraints. In: SODA. pp. 1433–1452 (2014)
5. Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincón, D., Sun, X., Wang, Y., Wei, W., Yuan, Y.: Influence maximization in social networks when negative opinions may emerge and propagate. In: SDM. pp. 379–390 (2011)
6. Goyal, A., Lu, W., Lakshmanan, L.V.S.: Simpath: An efficient algorithm for influence maximization under the linear threshold model. In: ICDM. pp. 211–220 (2011)
7. Gupta, S.: A conceptual framework that identifies antecedents and consequences of building socially responsible international brands. Thunderbird Int Business Rev **58**(3), 225–237
8. Gwadera, R., Loukides, G.: Cost-effective viral marketing in the latency aware independent cascade model. In: PAKDD. pp. 251–265 (2017)
9. Iyer, R., Bilmes, J.: Algorithms for approximate minimization of the difference between submodular functions, with applications. In: UAI. pp. 407–417 (2012)
10. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD. pp. 137–146 (2003)
11. Khan, A., Zehnder, B., Kossmann, D.: Revenue maximization by viral marketing: A social network host's perspective. In: ICDE. pp. 37–48 (2016)
12. Krause, A., Golovin, D.: Submodular function maximization. In: Tractability (2013)
13. Li, F., Li, C., Shan, M.: Labeled influence maximization in social networks for target marketing. In: PASSAT/SocialCom 2011. pp. 560–563 (2011)
14. Li, Y., Fan, J., Wang, Y., Tan, K.: Influence maximization on social graphs: A survey. TKDE **30**(10), 1852–1872 (2018)
15. Loukides, G., Gwadera, R.: Preventing the diffusion of information to vulnerable users while preserving pagerank. I. J. Data Science and Analytics **5**(1), 19–39 (2018)
16. Manning, C., Raghavan, P., Schtze, M.: Introduction to Information Retrieval (2008)
17. Mitrovic, M., Bun, M., Krause, A., Karbasi, A.: Differentially private submodular maximization: Data summarization in disguise. In: ICML. pp. 2478–2487 (2017)
18. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions. Mathematical Programming **14**(1), 265–294 (1978)
19. Nielsen: Sustainable selections: How socially responsible companies are turning a profit, https://bit.ly/2DW99pE
20. Pasumarthi, R., Narayanam, R., Ravindran, B.: Near optimal strategies for targeted marketing in social networks. In: AAMAS. pp. 1679–1680 (2015)
21. Shaw, G., Karami, A.: Computational content analysis of negative tweets for obesity, diet, diabetes, and exercise. Proc. of the Association for Information Science and Technology **54**(1), 357–365
22. Song, C., Hsu, W., Lee, M.L.: Targeted influence maximization in social networks. In: CIKM. pp. 1683–1692 (2016)
23. Svitkina, Z., Fleischer, L.: Submodular approximation: Sampling-based algorithms and lower bounds. SIAM J. Comput. **40**(6), 1715–1737 (2011)
24. Wang, C., Chen, W., Wang, Y.: Scalable influence maximization for independent cascade model in large-scale social networks. DMKD **25**(3), 545–576 (2012)
25. Wen, Y.T., Peng, W., Shuai, H.: Maximizing social influence on target users. In: PAKDD. pp. 701–712 (2018)