Maximizing Influence in Network Dynamics with Social Considerations

Vasiliki Raskopoulou
7115142100018

Examination committee:
Vassilios Zissimopoulos, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.
Archontia Giannopoulou, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.
Stavros Kolliopoulos, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.

Supervisor:
Vassilios Zissimopoulos, Professor, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.

ABSTRACT

This thesis explores the problem of influence maximization in social networks, with a focus on algorithms that take into account the existence of vulnerable users. We begin by modeling social networks and examining foundational properties, such as submodularity of the spread function. Various algorithms, including Naïve Greedy, subsampling and sandwich approximation, are reviewed for their efficiency and applicability. The core of our study examines some classic as well as innovative strategies like Difference Maximization, Ratio Maximization and Additive Smoothing Ratio Maximization, alongside a PageRank-inspired method to balance influence spread and protection of vulnerable users. Our findings aim to provide solutions that optimize influence while adopting a more socially responsible approach.

ΣΥΝΟΨΗ

Αυτή η διπλωματική εργασία διερευνά το πρόβλημα της μεγιστοποίησης επιρροής σε κοινωνικά δίκτυα, με έμφαση στους αλγορίθμους που λαμβάνουν υπόψη την ύπαρξη ευάλωτων χρηστών. Ξεκινάμε με τη μοντελοποίηση των κοινωνικών δικτύων και την εξέταση βασικών ιδιοτήτων. Διάφορες άπληστες τεχνικές, σε συνδυασμό με τη δειγματοληψία και την παρεμβολή, εξετάζονται ως προς την αποτελεσματικότητα και την εφαρμοσιμότητά τους. Ο πυρήνας της μελέτης μας εξετάζει κλασικές καθώς και καινοτόμες στρατηγικές όπως η Μεγιστοποίηση Διαφοράς, η Μεγιστοποίηση Λόγου και η Μεγιστοποίηση του Λόγου Προσθετικής Εξομάλυνσης (Additive Smoothing Ratio), μαζί με μια μέθοδο εμπνευσμένη από τον PageRank για την εξισορρόπηση της διάδοσης επιρροής και της προστασίας των ευάλωτων χρηστών. Τα ευρήματά μας στοχεύουν στο να δώσουν λύσεις που βελτιστοποιούν την επιρροή, ενώ ταυτόχρονα υιοθετούν μια κοινωνικά πιο υπεύθυνη προσέγγιση.

CONTENTS

INTRODUCTION

Social network and social media sites, where people are connected through heterogeneous social relationships, are gaining an increasing popularity to a degree that has rendered them an omnipresent integrated part of daily life. Therefore, social networks play a crucial role as a means for the dissemination of information among their members, by providing the most suitable platforms for the powerful word-of-mouth effect [9, 30]. The constant flow of information, ideas and behaviors along the social relationships lays the ground for marketing campaigns and especially for what we call viral marketing, a highly successful strategy based on the interpersonal influence [29]. Most organizations nowadays use this approach of promoting their products, ideas or innovations, with the premise that by initially selecting a few key members of the network who will recommend the adoption of the desirable behavior to their social circle of friends, a large cascade of influence can be triggered through a series of recommendations from these friends to their respective social circles and so on. In this way, most companies practicing viral marketing target the most influential users, for instance by giving them free samples, gifts or discounts of their products, hoping that eventually many users will buy them. Obviously, the initial set of influencers must also be as small as possible, as to avoid unnecessary costs [20]. Another aspect to take into account is the social responsibility factor, in the sense that the choice of the initial seed set must not only affect as many users as possible, but also avoid affecting users for whom the campaign might be harmful [4, 12].

In order to understand to which extend behaviors are adopted, we need to understand the manner in which the dynamics of influence play out within the network; in other words how likely it is that the users will actually be affected. Provided that we have the corresponding data with estimates of these probabilities from the network, we can model its members as nodes of a graph of relationships and interactions. Hence, given a social network G, a probabilistic model M describing how the nodes in G may influence each other and a small constant k, the influence maximization problem is to find k or less nodes in G that can (directly or indirectly) influence the largest number of nodes, that is, maximize the spread of influence.

In Chapter 1, we lay the groundwork for our study by formulating the problem of influence maximization, describing the model under which it is studied and highlighting key aspects such as the submodularity of our objective function.

Chapter 2 presents a comprehensive overview of some standard algorithms and well-established techniques designed to solve the influence maximization problem.

The main focus of this thesis, Chapter 3, investigates the influence maximization problem under the additional constraint of the existence of vulnerable nodes. Here, we introduce proposed strategies like Difference and Ratio Maximization, alongside the ASR Maximization (Additive Smoothing Ratio) approach. We also explore a PageRank-inspired method, adapting this well-known algorithm to address the unique challenges posed by vulnerable users within the network.

Finally, this thesis ends with three appendices. Appendix A gives us a small glimpse of the premises and methodologies required to evaluate the spread. Appendix B outlines the mathematical properties of submodular functions, which are central to the theoretical underpinnings of our analysis. Appendix C includes the proofs of Chapter 2 as well as some proofs from Chapter 3 that were not as crucial for the understanding of the concepts but still worth demonstrating.

CHAPTER 1 PRELIMINARIES

We begin by setting the foundation for our study of influence maximization. This chapter formulates the problem and describes the model under which it is studied, highlighting key aspects such as the probabilistic nature of influence spread and the submodular property of the spread function. This foundational understanding is crucial for the detailed algorithmic discussions in later chapters.

To start with, a social network is modeled as a digraph $G = (U, E)$, where each edge (u, v) describes the fact that v follows u. G is associated with a stochastic model M of the probabilities of influence between all pairs of nodes, and the spread of influence is described by a set function $\sigma : 2^U \to \mathbb{R}$ parameterized by M (see Appendix A). For a subset $S \subseteq U$, $\sigma(S)$ denotes the number of nodes (or any other metric representing gain) that are influenced by some node in S. Given a cardinality constraint $k < |U|$ determined by budgetary controls, our goal is to find an initial *seed set* S^* of size at most k that maximizes σ. A concise formulation of the INFLUENCE MAXIMIZATION PROBLEM (IMP) is given below [27]:

INFLUENCE MAXIMIZATION PROBLEM (IMP)
Input: Set of nodes $U = \{u_1, \ldots, u_n\}$, spread function $\sigma : 2^U \to \mathbb{R}$, integer $k < |U|$.
Output: $S^* \subseteq U, |S^*| \leq k$ such that

$$\sigma(S^*) = \max_{S \subseteq U} \{\sigma(S) : |S| \leq k\}. \tag{1.1}$$

In order to deal with this problem, we will consider the approach of the Independent Cascade Model.

1.1 Independent Cascade Model

The Independent Cascade Model (ICM) is the simplest dynamic cascade model for diffusion processes. Let $G = (U, E)$ be a social network. Starting with an initial set of active nodes $S \subseteq U$, the spread occurs in discrete steps independent of previous ones. That is, in step 1 the seed set S is activated, while all other nodes remain inactive.

If a node u becomes active in step t, it can activate any of its inactive out-neighbors in step $t+1$ solely. The probability $p_{u,v}$ of the activation of an out-neighbor v is a given parameter from the system's probabilistic model M. Once a node is activated it remains active for the rest of the process, which converges when no further activations are possible.

In order to ensure that ICM process is well-defined, we need to demonstrate that it yields the same distribution over outcomes, independent of the way we arrange the activations, i.e., regardless of the order in which active nodes attempt to activate their neighbors. For this, we consider a step in the process where node u is activated and tries to activate its out-neighbor v with probability $p_{u,v}$, and we parallel this event with the flipping of a biased coin. In this sense, the coin can be flipped at the very beginning of the process and not necessarily at the moment of u's activation, meaning that we can assume from the start and for every pair u, v if node v does actually get activated by u. If this event takes place, we say that edge (u, v) is *live*. Hence, by flipping all the coins in advance, we get an outcome X (a $|U|$-vector with 0's and 1's) and we can know exactly which nodes end up ultimately active [17].

Theorem 1.1 ([17])**.** The IMP is NP-hard under the ICM.

Proof. Consider an instance $\mathcal{I}_{SC}$ of the SET COVER (SC) problem, consisting of a ground set $U = \{u_1, \ldots, u_n\}$ and a collection of m subsets $S_1, \ldots, S_m$ and let k be the number of subsets we wish to select, whose union equals U (we can assume that $k < n < m$). We now construct a bipartite digraph G with $n + m$ nodes as follows: for every subset S_i, we create a node i, for every element u_j we create a node j and for every pair i, j such that $u_j \in S_i$ we create an edge (i, j) with activation probability $p_{i,j} = 1$. Then, deciding whether we can select k subsets whose union gives U is equivalent to deciding whether there exist k nodes $u_1, \ldots, u_k$ in G such that

$$\sigma(\{u_{i1}, \ldots, u_{ik}\}) \geq k + n.$$

Indeed, $\sigma(\{u_{i1}, \ldots, u_{ik}\})$ denotes the number of nodes reachable from $u_{i1}, \ldots, u_{ik}$ and, by construction of G, these are the elements in the selected subsets $S_{i1}, \ldots, S_{ik}$ which should be at least n (some elements may appear more than once), plus k for the seed set $\{u_{i1}, \ldots, u_{ik}\}$. Therefore, if we can find such a seed set, we can also find a selection of subsets for the SC problem, which implies that IMP is NP-hard. $\square$

1.1.1 Linear Threshold Model

Another equivalent model under which IMP is frequently studied is the Linear Threshold Model (LTM). We briefly present LTM here, but we will not need it until Section 3.6.

According to LTM, a node v is activated by an in-neighbor u with probability $p_{u,v}$ such that $\sum\limits_{u \in n^-(v)} p_{u,v} \leq 1$, allowing the possibility that v remains ultimately inactive. In addition, v is associated with a *threshold* $\mathrm{p}_v \in [0, 1]$, which represents the fraction of v's neighbors that is needed to activate v. Let $S \subseteq U$ be an initial set of active nodes. A node v is activated in step t if $p(v) = \sum\limits_{u \in n^-_{t-1}(v)} p_{u,v} \geq \mathrm{p}_v$, where $n^-_{t-1}(v)$ is the set of v's in-neighbors that are active in step $t - 1$. In addition, if $\mathrm{p} = \max\limits_{v \in S}\{p_v\}$, for some $S \subseteq U$, then S is activated if $\sum\limits_{v \in S} p(v) \geq |S| \cdot \mathrm{p}$.

The LTM is equivalent to ICM, which means that IMP is NP-hard under the LTM as well. For more details see [17, 7].

1.2 The Spread Function

A set function $f : 2^U \to \mathbb{R}$ is said to be *submodular* if and only if f satisfies the *diminishing returns* property, i.e., for every A, B such that $A \subseteq B \subseteq U$ and for every $u_i \notin B$, $f(B \mid u_i) \leq f(A \mid u_i)$, where $f(X \mid u_i) = f(X \cup \{u_i\}) - f(X)$ is the *marginal gain* of node u_i to X (see Appendix B). Intuitively, this means that adding an element to a smaller set yields a larger increase in function value than adding it to a larger set, which is true for our spread function: if a node is added later on to the seed set, it cannot bring a larger gain than if added earlier, since some of its neighbors might be already activated anyway. To demonstrate this formally, we have the following theorem.

Theorem 1.2 ([17]). The spread function $\sigma : 2^U \to \mathbb{R}$ is submodular under the ICM.

Proof. Firstly, we claim that a node v ends up active if and only if there is a *live path* from some node in the seed set S to v, i.e, a path consisting only of live edges. Now we consider the probability space $\mathcal{X}$ of all possible outcomes for the coin flips and we denote by $\sigma_X(S)$ the number of nodes that get activated when S is the initial seed set and X is the outcome of the coin flips. Also, we let $R(v, X)$ be the set of nodes reachable from v via some live path, so that $\sigma_X(S) = \left| \bigcup_{v \in S} R(v, X) \right|$. After having established this terminology and notation, we show that, for every X, σ_X is submodular.

Let $X \in \mathcal{X}$ be an outcome and let A, B be two subsets of U such that $A \subseteq B$. Then

$$\sigma_X(A \mid v) = \sigma_X(A \cup \{v\}) - \sigma_X(A) = \left| R(v, X) \setminus \bigcup_{u \in A} R(u, X) \right|.$$

Since $A \subseteq B$ it holds that

$$\begin{aligned} \sigma_X(A) \leq \sigma_X(B) &\Rightarrow \left| \bigcup_{v \in A} R(v, X) \right| \leq \left| \bigcup_{v \in B} R(v, X) \right| \\ &\Rightarrow \left| R(v, X) \setminus \bigcup_{u \in A} R(u, X) \right| \geq \left| R(v, X) \setminus \bigcup_{u \in B} R(u, X) \right|, \end{aligned}$$

so $\sigma_X(A \mid v) \geq \sigma_X(B \mid v)$, which means that σ_X is submodular. Since

$$\sigma(A) = \sum_{X \in \mathcal{X}} \Pr(X) \cdot \sigma_X(A),$$

σ is submodular as a non-negative linear combination of submodular functions (see B.4). □

As one would expect, σ is also submodular under the LTM [7].

Theorem 1.2 allows us to deal with the IMP in terms of the submodular function maximization problem. More precisely, our goal is to find a solution to the OPTIMAL SUBSET PROBLEM (OSP), i.e., to determine a subset $S^* \subseteq U$ such that $\sigma(S^*) = \max_{S \subseteq U}\{\sigma(S)\}$, where σ is a submodular function. Plus, since σ is also non-decreasing, it is essential that we set a cardinality constraint on S^*, therefore we summarize OSP as follows [23]:

OPTIMAL SUBSET PROBLEM

Input:$U = \{u_1, \ldots, u_n\}$, $\sigma : 2^U \to \mathbb{R}$, integer $k < |U|$.
Output: $S^* \subseteq U, |S^*| \leq k$ such that

$$\sigma(S^*) = \max_{S \subseteq U} \{\sigma(S) : |S| \leq k, \sigma(S) \text{ submodular}\}. \tag{1.2}$$

From now on, we will refer to IMP and OSP interchangeably, according to context.

CHAPTER 2 INFLUENCE MAXIMIZATION ALGORITHMS

This chapter provides a comprehensive overview of the most widely adopted algorithms designed to solve the influence maximization problem. Starting with the Naïve Greedy algorithm, a straightforward yet powerful approach, we then explore more advanced techniques such as subsampling and sandwich approximation.

2.1 Naïve Greedy

The simplest way to solve OSP is to begin with the empty set and select at each step the element that increases the marginal gain the most (see [27, 17, 23]).

Algorithm 1 NAÏVEGREEDY

```
Require: $U = \{u_1, \dots, u_n\}, \sigma : 2^U \to \mathbb{R}$, integer $k < |U|$ .
 1: $S^0 \leftarrow \emptyset, U^0 \leftarrow U, t \leftarrow 1$
 2: while $t \leq k$ do
 3:     Select $u_t \in U^{t-1}$ s.t. $\sigma(S^{t-1} \mid u_t) = \max_{u \in U^{t-1}} \{\sigma(S^{t-1} \mid u)\}$.
 4:     $d_{t-1} \leftarrow \sigma(S^{t-1} \mid u_t)$
 5:     if $d_{t-1} \leq 0$ then
 6:         return $S^G \leftarrow S^{t-1}$
 7:     else $S^t \leftarrow S^{t-1} \cup \{u_t\}, U^t \leftarrow U^{t-1} \setminus \{u_t\}$
 8:     if $t = k$ then
 9:         return $S^G \leftarrow S^k$
10:     else $t \leftarrow t + 1$
```

Note that, if more elements give the exact same increase, one is chosen arbitrarily (line 3) so that the greedy solution is not necessarily unique. Let S^* be an optimal solution to OSP and let S^G be a solution given by NAÏVEGREEDY. Then

$$\sigma(S^G) = \sigma(\emptyset) + \sum_{i=1}^{\ell} d_{i-1}, \quad \ell \leq k,$$

where ℓ denotes the number of iterations of the algorithm ($S^\ell = S^G$).

Now, for some real parameter $\theta \geq 0$, let $C(\theta)$ denote the class of submodular functions satisfying the property $f(A \mid u) \geq -\theta$ for all $A \subset U$ and $u \in U \setminus A$ (see Appendix B). Suppose that $\sigma \in C(\theta)$. From [27] we have the next result:

Proposition 2.1. The greedy solution is optimal, i.e., $\sigma(S^*) \leq \sigma(S^G)$.

Proof. See Appendix C. □

Theorem 2.2 ([18])**.** Let $\{S^t\}_{t\geq 0}$ be the greedily selected sets constructed by the NAÏVEGREEDY. Then

$$\sigma(S^\ell) \geq \left(1 - e^{-\ell/k}\right) \sigma(S^*),$$

where S^* is an optimal solution of size k.

Proof. See Appendix C. □

Corollary 2.3. If the NAÏVEGREEDY algorithm terminates at exactly k steps, then

$$\sigma(S^G) \geq \left(1 - e^{-1}\right) \sigma(S^*).$$

In reality, it is quite unlikely that the algorithm will terminate before k steps, given that the k constant represents a restriction on our budget, so we can claim that the approximation factor of NAÏVEGREEDY is simply $1 - e^{-1}$. Additionally, in the following algorithms we will omit to check whether further propagation is useful (lines 4-6), assuming that it always is but the algorithm does not proceed due to budget constraints.

We now discuss the evaluations of the spread function σ. As we can easily see, NAÏVEGREEDY performs $O(|U| \cdot k)$ evaluations of σ. So far, we have assumed the *value query model* according to which our algorithm has access to σ in a black-box manner, making queries to an oracle which returns the value $\sigma(S)$, for a queried set S (line 3). In practice, these evaluations are actually approximations performed via Monte Carlo methods, dynamic programming or sampling-based algorithms as well as heuristics [17, 8]. In fact, any of these methods introduce some approximation error $\varepsilon > 0$, so that the approximation factor of our greedy algorithm eventually becomes $1 - e^{-1} - \varepsilon$.

At this point we must explicitly state that whenever we refer to the spread function σ as input, it should be understood that we also refer to the parameters needed to approximate σ: the graph G and the probabilistic diffusion model M, i.e., the activation probabilities between G's nodes. These are employed by the aforementioned approximation algorithms that estimate σ. Even though we will not bother with the details of evaluating σ in the following analysis, it is important to bear in mind that these evaluations are costly and therefore keeping their number low is crucial for performance [8, 19, 13, 31, 6]. For more details on simulation methods see Appendix A.

2.2 Subsample Greedy

Among the various settings that further improve NAÏVEGREEDY, the notion of randomized sampling appears to be one of the most efficient in practice [31, 3, 24]. In this section, we have chosen to demonstrate SUBSAMPLEGREEDY, an algorithm that uses the technique of subsampling at each iteration in order to decrease the number of evaluations of the spread function.

To start with, we give the following definition.

Definition 2.4. An element u is *dummy* if for every subset $S \subseteq U$ it holds that

$$\sigma(S \cup \{u\}) = \sigma(S),$$

that is, it offers zero marginal gain to any subset $S \subseteq U$ with respect to σ.

Next, we present the SUBSAMPLEGREEDY algorithm introduced in [25].

Algorithm 2 SUBSAMPLEGREEDY

```
Require: $U = \{u_1, \dots, u_n\}$, $\sigma : 2^U \to \mathbb{R}$, integer $k < |U|$.      ▷ Phase I
 1: $D \leftarrow \{v_1, \dots, v_k\}$, where $v_i$ is a dummy element $\forall i \in [k]$
 2: while $|U|/k \notin \mathbb{N}$ do
 3:     $U \leftarrow U \cup \{u\}$, where $u \notin D$ is a dummy element
                                                                                  ▷ Phase II
 4: $S^0 \leftarrow \emptyset, t \leftarrow 1$
 5: while $t \leq k$ do
 6:     $R^t \leftarrow$ uniform random sample of $U$ with $|U|/k$ elements.
 7:     Select a random element $v_t \in D$, $R^t \leftarrow R^t \cup \{v_t\}$
 8:     Select $u_t \in R^t$ s.t. $\sigma(S^{t-1} \mid u_t) = \max_{u \in R^t} \{\sigma(S^{t-1} \mid u)\}$
 9:     $S^t \leftarrow S^{t-1} \cup \{u_t\}$
10:     $t \leftarrow t + 1$
                                                                                  ▷ Phase III
11: $S^k \leftarrow S^k \setminus \{v \mid v \text{ is dummy}\}$
12: return $S^G \leftarrow S^k$
```

The algorithm operates in three phases:

Phase I k dummy elements are created in a set D. Then, dummy elements not in D are created and added to U, until $|U|$ is a multiple of k.

Phase II A random sample of $|U|/k$ elements of U is created and a dummy element from D is added. From this sample, the algorithm selects the element u that most increases the largest marginal gain w.r.t. σ and adds it to S (initially empty). The process repeats k times, until S has k such elements.

Phase III All dummy elements are removed and the set is returned.

Observe that, since the algorithm randomly samples only $|U|/k$ elements at each iteration, after k iterations the number of evaluations of the spread function is $O(|U|)$, an obvious improvement.

Theorem 2.5. SUBSAMPLEGREEDY returns a solution S^G with

$$\mathbb{E}[\sigma(S^G)] \geq \left(1 - e^{-(1-1/e)}\right) \sigma(S^*),$$

where $\mathbb{E}[\sigma(S^G)]$ is the expected value of the greedy solution and S^* is an optimal solution of size at most k.

Proof. See Appendix C. □

2.3 Sandwich Approximation

Another technique that proves very useful is the Sandwich Approximation (SA). Suppose we have a non-negative non-submodular function $\sigma : 2^U \to \mathbb{R}$ and two non-negative monotone submodular bound functions $\sigma^L, \sigma^U : 2^U \to \mathbb{R}$ s.t.

$$\sigma^L(S) \leq \sigma(S) \leq \sigma^U(S), \quad \forall S \subseteq U.$$

Given a parameter $k < |U|$, the SA strategy [22] deals with OSP by applying some greedy optimization algorithm (such as NaïveGreedy) to all three functions $\sigma, \sigma^L, \sigma^U$. From the corresponding constructed sets S^O, S^L, S^U SA selects the one with the largest value in σ.

The advantage of SA is that it can approximately maximize a non-submodular function by simultaneously optimizing its submodular bounds, which results in an approximation guarantee not generally provided in the non-submodular case. More precisely, by applying NaïveGreedy, or any other greedy alternative with approximation factor $1 - e^{-1}$, we get the subsequent theorem:

Theorem 2.6 ([22])**.** It holds that

$$\sigma(S^G) \geq \max\left\{\frac{\sigma(S^U)}{\sigma^U(S^U)}, \frac{\sigma^L(S^*)}{\sigma(S^*)}\right\}\left(1 - \frac{1}{e}\right)\sigma(S^*),$$

where S^G is the greedy solution, S^U is the greedily selected set maximizing σ^U and S^* is the optimal solution w.r.t σ.

Proof. See Appendix C. □

It is important to notice that the bound functions should be as tight as possible, lest the fractions in **max** become arbitrarily small, leading to a trivial approximation factor.

CHAPTER 3

INFLUENCE MAXIMIZATION UNDER VULNERABILITY

In this chapter, we examine the IMP in a social network with the presence of vulnerable nodes, i.e., nodes that we want to avoid influencing. More precisely, let $U = N \cup V$ be the disjoint union of two sets of *non-vulnerable* (N) and *vulnerable* (V) nodes. We again consider the submodular spread function σ, but now we take its two restrictions $\sigma_N : 2^N \to \mathbb{R}$ and $\sigma_V : 2^N \to \mathbb{R}$ on the aforementioned sets, in the sense that $\sigma_N(S)$ only returns the nodes of N that are activated by S and $\sigma_V(S)$ the corresponding nodes of V. Our goal would be to select a seed set to maximize σ_N while minimizing σ_V. The most direct approaches we can consider in this setting are to maximize either the difference $\sigma_N - \sigma_V$ or the ratio σ_N/σ_V.

In a more general context, let $\sigma : 2^U \to \mathbb{R}_{\geq 0}$ and $\rho : 2^U \to \mathbb{R}_{\geq 0}$ be non-decreasing submodular set functions, such that $\sigma(\emptyset) \geq 0, \rho(\emptyset) > 0$ and $\sigma(u) > 0, \rho(u) > 0, \forall u \in U$. Hence, since σ, ρ are monotone we have that $\sigma(S) > 0, \rho(S) > 0, \forall S, \emptyset \subset S \subseteq U$. Our two problems are described below [28]:

Difference of Submodular Maximization (DS Max)
Input:$U = \{u_1, \dots, u_n\}, \sigma : 2^U \to \mathbb{R}_{\geq 0}, \rho : 2^U \to \mathbb{R}_{\geq 0}$.
Output: $\emptyset \subset S^* \subset U$ such that

$$\sigma(S^*) - \rho(S^*) = \max_{\emptyset \subset S \subseteq U} \{\sigma(S) - \rho(S)\}. \tag{3.1}$$

Ratio of Submodular Maximization (RS Max)
Input:$U = \{u_1, \dots, u_n\}, \sigma : 2^U \to \mathbb{R}_{\geq 0}, \rho : 2^U \to \mathbb{R}_{\geq 0}$.
Output: $\emptyset \subset S^* \subset U$ such that

$$\frac{\sigma(S^*)}{\rho(S^*)} = \max_{\emptyset \subset S \subseteq U} \left\{ \frac{\sigma(S)}{\rho(S)} \right\}. \tag{3.2}$$

In the sections that follow, we present some of the basic algorithms for the above problems.

3.1 Modular Bounds

Before delving into the algorithms, we must define notion of modular bounds [1, 5, 15] which we are going to use in our analysis (see Appendix B).

Let $f : 2^U \to \mathbb{R}_{\geq 0}$ be a submodular function. For any parameter $Y \subseteq U$, we define the *modular upper bounds* as the (modular) functions

$$\widehat{f_{Y,1}}(X) = f(Y) + \sum_{u \in X \setminus Y} (f(u) - f(\emptyset)) - \sum_{u \in Y \setminus X} (f(Y) - f(Y \setminus \{u\})),$$

$$\widehat{f_{Y,2}}(X) = f(Y) + \sum_{u \in X \setminus Y} f(Y \cup \{u\}) - f(Y)) - \sum_{u \in Y \setminus X} (f(U) - f(U \setminus \{u\})).$$

We will be using $\widehat{f_{Y,1}}$ unless explicitly stated otherwise. For simplicity, we will just denote it by $\widehat{f_Y}$. We observe that $\widehat{f_Y}(X)$ is not computed for X, but for Y while adding the maximum possible contributions to $f(X)$ of each element in $X \setminus Y$ and subtracting the minimum possible contributions to $f(Y)$ of every element in $Y \setminus X$. Indeed, since f is submodular, if $u \in X$ then $f(u) - f(\emptyset)$ does represent an overestimation of u's marginal gain to X, while $f(Y) - f(Y \setminus \{u\})$ being the minimum marginal gain $u \in Y$ can contribute to $f(Y)$ indicates an underestimation of the contribution of $Y \setminus X$. Obviously, the equality holds when $Y = X$, so that $X \setminus Y = Y \setminus X = \emptyset$.

Similarly, we define the *modular lower bound* with parameter $Y \subseteq U$ as the (modular) function

$$\widetilde{f_{Y,\sigma^Y}}(X) = \sum_{u \in X} f_{Y,\sigma^Y}(u),$$

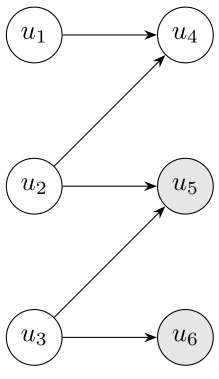
where σ^Y is a random permutation of the elements of Y and

$$f_{Y,\sigma^Y}(u) = \begin{cases} f(\sigma_u^Y) - f(\sigma_{u^-}^Y), & \text{if } u \in Y \\ 0, & \text{otherwise} \end{cases},$$

where $\sigma_{u^-}^Y$ denotes the prefix of all elements in σ^Y that come before u and σ_u^Y is $\sigma_{u^-}^Y$ along with u.

Here we observe that the computation of $\widetilde{f_{Y,\sigma^Y}}(X)$ takes into account the marginal gain of u to its corresponding prefix $\sigma_{u^-}^Y$ in permutation σ, but only for $u \in X \cap Y$, providing a value at most equal to $f(X)$. The equality again holds for $Y = X$ (telescoping sum).

Example 3.1. *We consider the network below, and let $X = \{u_2, u_3\}, Y = \{u_2, u_6\}$. We will compute σ's bounds on X with parameter Y. For the upper bound, we have*



$$\begin{aligned}\hat{\sigma}_Y(X) &= \sigma(Y) + (\sigma(u_3) - \sigma(\emptyset)) - (\sigma(Y) - \sigma(Y \setminus \{u_6\})) \\ &= 4 + (3 - 0) - (4 - 3) = 6.\end{aligned}$$

Also, if we consider the permutation $\pi^Y = (u_6, u_2)$ of Y, then $\sigma_{Y,\pi^Y}(u_2) = \sigma(Y) - \sigma(u_6) = 4 - 1 = 3$, $\sigma_{Y,\pi^Y}(u_3) = 0$ and

$$\widetilde{\sigma_{Y,\pi^Y}}(X) = \sigma_{Y,\pi^Y}(u_2) + \sigma_{Y,\pi^Y}(u_3) = 3.$$

Since $\sigma(X) = 5$, it holds that $\widetilde{\sigma_{Y,\pi^Y}}(X) < \sigma(X) < \hat{\sigma}_Y(X)$.

3.2 Difference of Submodular Maximization

In this section, we seek to approximate DS MAX with two greedy algorithms, one that is direct and the other one that uses our functions' modular bounds, deploying a Minorization-Maximization technique. To begin with, we will denote the difference $\sigma - \rho$ by d:

$$d(S) := \sigma(S) - \rho(S), \forall S \in 2^U.$$

Theorem 3.2. The difference between two monotone submodular functions is not monotone neither is submodular.

Proof. For now we will prove this using a counterexample, but later in Lemma 3.4 we will see an even stronger result. In Example 3.1, consider u_5 and u_6 to be vulnerable and set $d := \sigma_N - \sigma_V$. Then, if we set $S = \{u_1, u_4\}$ and $T = \{u_1, u_3, u_4\}$, we have

$$d(S) = \sigma_N(S) - \sigma_V(S) = 2 - 0 = 2$$

and

$$d(T) = \sigma_N(T) - \sigma_V(T) = 3 - 2 = 1 < d(S)$$

while $S \subset T$, which means that d is not monotone. On the other hand,

$$d(S \mid u_2) = d(S \cup \{u_2\}) - d(S) = \sigma_N(S \cup \{u_2\}) - \sigma_V(S \cup \{u_2\}) - d(S) = 3 - 1 - 2 = 0$$

and

$$d(T \mid u_2) = d(T \cup \{u_2\}) - d(T) = \sigma_N(S \cup \{u_2\}) - \sigma_V(S \cup \{u_2\}) - d(T) = 4 - 2 - 1 = 1,$$

i.e., $S \subset T$ and $d(S \mid u_2) < d(T \mid u_2)$, which means that d is not submodular. □

The standard greedy method to maximize the difference of two submodular functions is given by the following algorithm:

Algorithm 3 GREEDDIFFERENCE

```
Require: $U = N \cup V, \sigma, \rho : 2^N \to \mathbb{R}_{\geq 0}$.
1: Select $S^0 \leftarrow \emptyset, N^0 \leftarrow N, t \leftarrow 1$
2: while $N^t \neq \emptyset$ do
3:     Select $u_t \in N^{t-1}$ s.t. $d(S^{t-1} \mid u_t) = \max_{u \in N^{t-1}} \{d(S^{t-1} \mid u)\}$.
4:     $S^t \leftarrow S^{t-1} \cup \{u_t\}$
5:     $N^t \leftarrow \{u \in N^{t-1} \mid d(S^t \mid u) > 0\}$
6:     $t \leftarrow t + 1$
7: return $S^G \leftarrow S^t$
```

The above naïve approach can be further improved using the functions' modular bounds we mentioned earlier. Among the variations proposed by [26, 15], we present the most comprehensive case of greedily optimizing at each step some instance of the difference of the modular bounds of both σ, ρ instead of the actual difference. Specifically, we will consider the function

$$d_{Y,\pi^Y}(X) = \max_{i \in \{1,2\}} \left\{ \widetilde{\sigma_{Y,\pi^Y}}(X) - \widehat{\rho_{Y,i}}(X) \right\}$$

describing such an instance depending on parameter $Y \subseteq U$. As described in Section 3.1, π^Y denotes a random permutation of Y. Also note that $d_{S^{t-1},\pi^{t-1}}$ minorizes the objective function d. We modify the following algorithm from [15] to correspond to our maximization criteria.

Algorithm 4 MMaxDiff

```
Require: $U = N \cup V, \sigma, \rho : 2^N \to \mathbb{R}_{\geq 0}$.
1: Select $S^0 \leftarrow \emptyset, t \leftarrow 1$
2: while $S^t \neq S^{t-1}$ do
3:     Select a permutation $\pi^{t-1} := \pi^{S^{t-1}}$ of $S^{t-1}$.
4:     Select $S^t \subseteq N$ s.t. $d_{S^{t-1},\pi^{t-1}}(S^t) = \max_{S \subseteq N} \{d_{S^{t-1},\pi^{t-1}}(S)\}$.
5:     $t \leftarrow t + 1$
6: return $S^G \leftarrow S^t$
```

This algorithm chooses a permutation of the current set and performs maximization on the difference of the bounds with parameter the current set. The choice of permutation is essential for the algorithm's quality, and we may employ some heuristic such as choosing $\pi^{t-1} \in \arg\max_{\pi} \max_{X} \{d_{S^{t-1},\pi}(X)\}$ or choosing an ordering of S^{t-1} according to greatest gains of σ.

Theorem 3.3. If the objective value does not decrease on checking $O(n)$ different permutations with different elements at adjacent positions, then the algorithm has converged to a local maximum of d.

Proof. Firstly, we show that MMaxDiff indeed monotonically increases the value of difference at every iteration:

$$\begin{aligned}
\sigma(S^{t+1}) - \rho(S^{t+1}) &\geq \max_{i \in \{1,2\}} \left\{ \widetilde{\sigma_{S^t,\pi_t^{S^t}}}(S^{t+1}) - \widehat{\rho_{S^t,i}}(S^{t+1}) \right\} \\
&\geq \max_{i \in \{1,2\}} \left\{ \widetilde{\sigma_{S^t,\pi_t^{S^t}}}(S^{t}) - \widehat{\rho_{S^t,i}}(S^{t}) \right\} \\
&= \sigma(S^t) - \rho(S^t).
\end{aligned}$$

Furthermore, suppose that in some iteration, say $t+1$, where we either add or remove an element from S^t, the objective value of d_{S^t,π^t} does not increase. Then S^t is a local optimum for all modular bounds and therefore

$$\widetilde{\sigma_{S^t,\pi^t}}(S^t) - \rho_{S^t,1}(S^t) \geq \widetilde{\sigma_{S^t,\pi^t}}(S^t \setminus \{v\}) - \rho_{S^t,1}(S^t \setminus \{v\}), \forall v \in S^t$$

and

$$\widetilde{\sigma_{S^t,\pi^t}}(S^t) - \rho_{S^t,2}(S^t) \geq \widetilde{\sigma_{S^t,\pi^t}}(S^t \cup \{v\}) - \rho_{S^t,1}(S^t \cup \{v\}), \forall v \notin S^t,$$

which means that d_{S^t,π^t} is also at a local optimum.

Since $\widetilde{\sigma_{S^t,\pi^t}}(S^t \setminus \{v\}) = \sigma(S^t \setminus \{v\})$ and $\rho_{S^t,1}(S^t \setminus \{v\}) = \rho(S^t \setminus \{v\})$ we have

$$\sigma(S^t) - \rho(S^t) \geq \sigma(S^t \setminus \{v\}) - \rho(S^t \setminus \{v\})$$

and similarly, since $\widetilde{\sigma_{S^t,\pi^t}}(S^t \cup \{v\}) = \sigma(S^t)$ and $\rho_{S^t,2}(S^t \cup \{v\}) = \rho(S^t \cup \{v\})$ we also have

$$\sigma(S^t) - \rho(S^t) \geq \sigma(S^t \cup \{v\}) - \rho(S^t \cup \{v\}).$$

So in every case $d(S^t)$ is a local optimum. Notice that this observation allows us to consider only $O(n)$ permutations of S^{t+1}, each with a different element at position $|S^t| - 1$ or $|S^t| + 1$. □

At this point, it is noteworthy to comment on the nature of the difference of submodular functions.

Lemma 3.4 ([15]). Let $f : 2^U \to \mathbb{R}$ be a set function. Then there exist submodular functions $\sigma, \rho : 2^U \to \mathbb{R}$ such that $f(S) = \sigma(S) - \rho(S), \forall S \subseteq U$.

Proof. See Appendix C. □

In essence, even though the optimization of difference is conceptually straightforward, in the case of DS maximization we observe that the problem in question is very general and encompasses the strictly larger class of set function optimization problems. Moreover, as discussed above, an important factor in our case would be to maximize DS under cardinality constraints. It is becoming clear that the problem is NP-hard and also hard to approximate.

3.3 Ratio of Submodular Maximization

Here, we focus on the more interesting problem of RS MAX, by presenting the standard GREEDRATIO algorithm as well as a Minorization-Maximization technique. Similarly to submodular difference, we have the next theorem.

Theorem 3.5. The ratio between two monotone submodular functions is not monotone neither is submodular.

Proof. We will again use the counterexample of 3.1, where u_5 and u_6 are vulnerable. If we set $S = \{u_2\}$ and $T = \{u_2, u_3\}$, we have

$$\frac{\sigma_N(S)}{\sigma_V(S)} = \frac{2}{1} = 2$$

and

$$\frac{\sigma_N(T)}{\sigma_V(T)} = \frac{3}{2} < \frac{\sigma_N(S)}{\sigma_V(S)}$$

while $S \subset T$, which means that the ratio is not monotone. On the other hand, let $S = \{u_4\}$ and $T = \{u_2, u_4\}$. Then

$$\frac{\sigma_N(S \mid u_3)}{\sigma_V(S \mid u_3)} = \frac{\sigma_N(S \cup \{u_3\}) - \sigma_N(S)}{\sigma_V(S \cup \{u_3\}) - \sigma_V(S)} = \frac{2-1}{2-0} = \frac{1}{2}$$

and

$$\frac{\sigma_N(T \mid u_3)}{\sigma_V(T \mid u_3)} = \frac{\sigma_N(T \cup \{u_3\}) - \sigma_N(T)}{\sigma_V(T \cup \{u_3\}) - \sigma_V(T)} = \frac{3-2}{2-1} = 1$$

i.e., $S \subset T$ and $\frac{\sigma_N(S|u_3)}{\sigma_V(S|u_3)} < \frac{\sigma_N(T|u_3)}{\sigma_V(T|u_3)}$, which means that the ratio is not submodular. □

The first greedy approach to RS Max is given by the following algorithm [1]:

Algorithm 5 GreedRatio

```
Require: $U = N \cup V, \sigma, \rho : 2^N \to \mathbb{R}_{\geq 0}$.
1: Select $S^0 \leftarrow \emptyset, N^0 \leftarrow N, t \leftarrow 1$
2: while $N^t \neq \emptyset$ do
3:     Select $u_t \in N^{t-1}$ s.t. $\frac{\sigma(S^{t-1}|u_t)}{\rho(S^{t-1}|u_t)} = \max_{u \in N^{t-1}} \left\{ \frac{\sigma(S^{t-1}|u)}{\rho(S^{t-1}|u)} \right\}$.
4:     $S^t \leftarrow S^{t-1} \cup \{u_t\}$
5:     $N^t \leftarrow \{u \in N^{t-1} \mid \rho(S^t \mid u) > 0\}$
6:     $t \leftarrow t + 1$
7: return $S^G \leftarrow S^t$
```

Theorem 3.6. GreedRatio is guaranteed to yield a solution S^G such that

$$\frac{\sigma(S^G)}{\rho(S^G)} \geq \left(1 - e^{\kappa_\rho - 1}\right) \frac{\sigma(S^*)}{\rho(S^*)},$$

where S^* is the optimal solution and $\kappa_\rho = 1 - \min_{u \in U} \frac{\rho(U) - \rho(U \setminus \{u\})}{\rho(u)}$ is the *submodular curvature* of ρ (see Appendix B).

Before proving Theorem 3.6, we will present an extended version of a useful lemma from [16].

Lemma 3.7 ([16]). Let $f : 2^U \to \mathbb{R}_{\geq 0}$ be submodular, and suppose that the *supermodular curvature* of f, $\kappa^f = 1 - \min_{u \in U} \frac{f(u)}{f(U) - f(U \setminus \{u\})}$ is defined (see Appendix B). It holds that

$$\sum_{u \in X} f(u) \leq \frac{|X|}{1 + (|X| - 1)(1 - \kappa_f)} f(X)$$

and

$$\sum_{u \in X} f(X \setminus \{u\} \mid u) \geq \frac{|X|}{1 + (|X| - 1)(1 - \kappa^f)} f(X).$$

Proof. For the first claim, we observe that

$$(1 - \kappa_f(X)) \sum_{u \in X} f(u) = \sum_{u \in X} f(X \setminus \{u\} \mid u) \tag{3.3}$$

$$f(X) - f(x) \geq \sum_{u \in X \setminus \{x\}} f(X \setminus \{u\} \mid u), \forall x \in X, \tag{3.4}$$

where 3.3 is the definition of $\kappa_f(X)$ and 3.4 derives from submodularity. If we sum

3.4 over $x \in X$, we obtain

$$\begin{aligned}
|X|f(X) - \sum_{x\in X} f(x) &\geq \sum_{x\in X} \sum_{u\in X\setminus\{x\}} f(X \setminus \{u\} \mid u) \\
&= \sum_{x\in X}\sum_{u\in X} f(X\setminus\{u\} \mid u) - \sum_{x\in X} f(X\setminus\{x\} \mid x) \\
&= (|X|-1)\sum_{u\in X} f(X\setminus\{u\} \mid u) \\
&= (|X|-1)(1-\kappa_f(X))\sum_{u\in X} f(u) && \text{(from 3.3)} \\
&\geq (|X|-1)(1-\kappa_f)\sum_{u\in X} f(u) && (\kappa_f(X)\leq \kappa_f, \forall X)
\end{aligned}$$

Hence,

$$\sum_{u\in X} f(u) \leq \frac{|X|}{1+(|X|-1)(1-\kappa_f)} f(X).$$

Similarly, we have

$$\begin{aligned}
(1-\kappa^f(X))\sum_{u\in X} f(X\setminus\{u\} \mid u) &= \sum_{u\in X} f(u) \\
f(X) - f(X\setminus\{x\} \mid x) &\leq \sum_{u\in X\setminus\{x\}} f(u), \forall x\in X.
\end{aligned}$$

Again by summing the second inequality over x we get

$$\begin{aligned}
|X|f(X) - \sum_{x\in X} f(X\setminus\{x\} \mid x) &\leq \sum_{x\in X}\sum_{u\in X\setminus\{x\}} f(u) \\
&= \sum_{x\in X}\sum_{u\in X} f(u) - \sum_{x\in X} f(x) \\
&= (|X|-1)\sum_{u\in X} f(u) \\
&= (|X|-1)(1-\kappa^f(X))\sum_{u\in X} f(X\setminus\{u\} \mid u) \\
&\leq (|X|-1)(1-\kappa^f)\sum_{u\in X} f(X\setminus\{u\} \mid u)
\end{aligned}$$

and therefore

$$\sum_{u\in X} f(X\setminus\{u\} \mid u) \geq \frac{|X|}{1+(|X|-1)(1-\kappa^f)} f(X).$$

□

Now, we alter the proof of [1] for our maximization purposes.

Proof of Theorem 3.6. Let $\{S^t\}_{t\geq 0}$ be the greedily selected sets constructed by the GreedRatio and $u_1, \ldots, u_\ell$ be the greedily selected elements. Now let $k \leq \ell$ be the largest index such that $\rho(S^k) \leq \rho(S^*)$. For $1 \leq t \leq k$, according to the algorithm we have that

$$\frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \geq \frac{\sigma(S^{t-1} \mid v)}{\rho(S^{t-1} \mid v)}, \forall v \notin S^{t-1} \quad \Rightarrow$$

$$\sigma(S^{t-1} \mid v) \leq \frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \rho(S^{t-1} \mid v), \forall v \notin S^{t-1}.$$

Therefore, since σ is submodular, it holds that

$$\begin{aligned}\sigma(S^*) &\leq \sigma(S^{t-1}) + \sum_{v \in S^* \setminus S^{t-1}} \sigma(S^{t-1} \mid v) \\ &\leq \sigma(S^{t-1}) + \sum_{v \in S^* \setminus S^{t-1}} \frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \rho(S^{t-1} \mid v).\end{aligned}$$

Also from the above lemma and B.7,

$$(1 - \kappa_\rho) \sum_{v \in S^*} \rho(v) \leq (1 - \kappa_\rho(S^*)) \sum_{v \in S^*} \rho(v) = \sum_{v \in S^*} \rho(S^* \setminus \{v\} \mid v) \leq \rho(S^*).$$

Hence,

$$\begin{aligned}\sigma(S^*) - \sigma(S^{t-1}) &\leq \frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \sum_{v \in S^* \setminus S^{t-1}} \rho(S^{t-1} \mid v) \leq \frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \sum_{v \in S^*} \rho(v) \\ &\leq \frac{\sigma(S^{t-1} \mid u_t)}{\rho(S^{t-1} \mid u_t)} \frac{1}{1 - \kappa_\rho} \rho(S^*).\end{aligned}$$

In addition, we observe that

$$\sigma(S^{t-1} \mid u_t) = \sigma(S^t) - \sigma(S^{t-1}) = \sigma(S^*) - \sigma(S^{t-1}) - (\sigma(S^*) - \sigma(S^t)),$$

so the previous inequality yields

$$\frac{\sigma(S^*) - \sigma(S^t)}{\sigma(S^*) - \sigma(S^{t-1})} \leq \left[1 - \frac{(1 - \kappa_\rho)\rho(S^{t-1} \mid u_t)}{\rho(S^*)}\right].$$

Multiplying all k inequalities we get

$$\begin{aligned}\sigma(S^*) - \sigma(S^k) &\leq \prod_{t=1}^{k} \left[1 - \frac{(1 - \kappa_\rho)\rho(S^{t-1} \mid u_t)}{\rho(S^*)}\right] \sigma(S^*) \\ &\leq \prod_{t=1}^{k} e^{-\frac{(1-\kappa_\rho)\rho(S^{t-1} \mid u_t)}{\rho(S^*)}} \sigma(S^*) \\ &\leq e^{-\frac{(1-\kappa_\rho)\rho(S^k)}{\rho(S^*}} \Rightarrow \\ \sigma(S^k) &\geq \left[1 - e^{-\frac{(1-\kappa_\rho)\rho(S^k)}{\rho(S^*)}}\right] \sigma(S^*) \Rightarrow \\ \frac{\sigma(S^k)}{\rho(S^k)} &\geq \left[1 - e^{-\frac{(1-\kappa_\rho)\rho(S^k)}{\rho(S^*)}}\right] \frac{\sigma(S^*)}{\rho(S^*)} \frac{\rho(S^*)}{\rho(S^k)}.\end{aligned}$$

But $\frac{\rho(S^k)}{\rho(S^*)} \leq 1$ and $f(x) = \frac{1-e^{-(1-\kappa)x}}{x}$ is monotonically decreasing for $0 < \kappa < 1$, i.e.,

$$f\left(\frac{\rho(S^k)}{\rho(S^*)}\right) \geq f(1) \Rightarrow$$

$$\left[1 - e^{-\frac{(1-\kappa_\rho)\rho(S^k)}{\rho(S^*)}}\right] \frac{\sigma(S^*)}{\rho(S^*)} \frac{\rho(S^*)}{\rho(S^k)} \geq \left[1 - e^{-(1-\kappa_\rho)}\right] \frac{\sigma(S^*)}{\rho(S^*)} \Rightarrow$$

$$\frac{\sigma(S^k)}{\rho(S^k)} \geq \left[1 - e^{-(1-\kappa_\rho)}\right] \frac{\sigma(S^*)}{\rho(S^*)},$$

which concludes the proof. □

It is clear that, if we set $\sigma := \sigma_N$ and $\rho := \sigma_V$, GreedRatio approximates a seed set with a factor of $1 - e^{\kappa_{\sigma_V} - 1}$. Observe that, in the ideal case where σ_V is modular, i.e., the number of vulnerable nodes activated by a seed node is constant, then $\kappa_{\sigma_V} = 0$ and the approximation factor equals $1 - e^{-1}$, as in NaïveGreedy. On the other hand, in the worst case where σ_V is fully curved, then we get a trivial, zero factor.

Another framework proposed by [1] is a Minorization-Maximization scheme described below.

Algorithm 6 MMaxRatio

```
Require: $U = N \cup V, \sigma, \rho : 2^N \to \mathbb{R}_{\geq 0}$.
1: $S^0$ ←arbitrary subset of $N$, $t \leftarrow 0$
2: repeat
3:     Select a permutation $\pi^t := \pi^{S^t}$ of $S^t$.
4:     Select $S_1 \subseteq N$ s.t. $\frac{\breve{\sigma}_t(S_1)}{\rho(S_1)} = \max_{S \subseteq N}\{\frac{\breve{\sigma}_t(S)}{\rho(S)}\}$.          ▷ Algorithm B
5:     Select $S_2 \subseteq N$ s.t. $\frac{\sigma(S_2)}{\widehat{\rho}_t(S_2)} = \max_{S \subseteq N}\{\frac{\sigma(S)}{\widehat{\rho}_t(S)}\}$.          ▷ GreedRatio
6:     $S^{t+1} \leftarrow \arg\max_{S \in \{S_1, S_2\}}\{\frac{\sigma(S)}{\rho(S)}\}$.
7:     $t \leftarrow t + 1$
8: until $S^t = S^{t-1}$
9: return $S^G \leftarrow S^t$
```

At every iteration, MMaxRatio constructs $\breve{\sigma}$ and $\widehat{\rho}$ using as parameter the current seed set, and minimizes the ratio in two stages:

- Firstly, it uses the ratio of (modular) $\breve{\sigma}$ and (submodular) ρ, which, as we will discuss in the next section, can be approximated by Algorithm B. For now, simply observe that the function $\lambda\rho - \breve{\sigma}, \lambda > 0$ is submodular and therefore can be minimized by a polynomial algorithm [14].

- Secondly, it uses the ratio of (submodular) σ and (modular) $\widehat{\rho}$, which can be approximated by GreedRatio. Since $\widehat{\rho}$ is modular, we have that $\kappa_{\widehat{\rho}} = 0$ and therefore GreedRatio obtains a solution of approximation factor $1 - e^{-1}$.

Then, the algorithm selects the optimal solution between the two and continues until convergence is reached. Note that MMaxRatio maximizes ratios that minorize the objective ratio, i.e., that represent lower bounds of the objective ratio.

Next, suppose that there exists at least one element $u \in U$ with non-zero marginal gain to the ground set U, i.e., $\sigma(U) - \sigma(U \setminus \{u\}) > 0$. Analogously to the result of [1], we have the next theorem.

Theorem 3.8. MMaxRatio outputs a greedy solution with a worst-case approximation factor of

$$O\left(\max\left\{\frac{n}{1+(n-1)(1-\kappa^\sigma)}, \frac{1+(n-1)(1-\kappa_\rho)}{n}\right\}\right),$$

where $n = |U|$, κ^σ is the supermodular curvature of σ and κ_ρ is the submodular curvature of ρ.

Proof. Lemma 3.7 states that the simple lower bound of σ and the simple upper bound of ρ assume approximation factors of

$$\alpha(X) = \frac{|X|}{1+(|X|-1)(1-\kappa^\sigma)} \quad \text{and} \quad \beta(X) = \frac{|X|}{1+(|X|-1)(1-\kappa_\rho)},$$

respectively. Thus,

$$\widetilde{\sigma_{Y,\pi^Y}}(X) \geq \sum_{u\in X} \sigma(X\setminus\{u\} \mid u) \geq \alpha(X)\sigma(X) \Rightarrow \frac{\widetilde{\sigma_{Y,\pi^Y}}(X)}{\rho(X)} \geq \alpha(X)\frac{\sigma(X)}{\rho(X)}$$

$$\widehat{\rho_Y}(X) \leq \sum_{u\in X} \rho(u) \leq \beta(X)\rho(X) \Rightarrow \frac{\sigma(X)}{\widehat{\rho_Y}(X)} \geq \frac{1}{\beta(X)}\frac{\sigma(X)}{\rho(X)}.$$

We set $\alpha := \alpha(U) = \frac{n}{1+(n-1)(1-\kappa^\sigma)}$ and $\beta := \beta(U) = \frac{n}{1+(n-1)(1-\kappa_\rho)}$. Since $\alpha(X)$ is a monotone-decreasing function and $\beta(X)$ is monotone-increasing it holds that $\alpha(X) \geq \alpha$ and $\frac{1}{\beta(X)} \geq \frac{1}{\beta}$ and the approximation factor is in $O\left(\max\left\{\alpha, \frac{1}{\beta}\right\}\right)$. □

3.4 Comparison and weak equivalence

DS Max and RS Max are two related problems, as demonstrated in [1]. We will attempt to showcase this relation here. If we consider the slightly different λ-DS Maximization, same as before but with the objective function $\sigma - \lambda\rho$, where $\lambda \geq 0$, then the following lemma holds.

Lemma 3.9. Let $\epsilon > 0$ and A be an exact algorithm for DS Maximization. Then, there exists a $1/(1+\epsilon)$-approximation algorithm B for RS Maximization .

Proof. B can be constructed using A as follows.

Algorithm B

```
Require: $U, \sigma, \rho, 0 \leq \epsilon < 1$, A.
1: $\lambda_{\min} \leftarrow 0$, $\lambda_{\max} \leftarrow \frac{\max_{X\subseteq U}\sigma(X)}{\min_{X\subseteq U}\rho(X)}$
2: while $\lambda_{\max} > (1+\epsilon)\lambda_{\min}$ do
3:     $\lambda_{\text{mid}} \leftarrow \frac{\lambda_{\max}+\lambda_{\min}}{2}$
4:     $S \leftarrow$A$(\sigma, \lambda_{\text{mid}}\rho)$
5:     if $\frac{\sigma(S)}{\rho(S)} \leq \lambda_{\text{mid}}$ then
6:         $\lambda_{\max} \leftarrow \lambda_{\text{mid}}$
7:     else
8:         $\lambda_{\min} \leftarrow \lambda_{\text{mid}}$
9: return $S^G \leftarrow S$
```

In line 1, $\max\limits_{X\subseteq U}\sigma(X)$ and $\min\limits_{X\subseteq U}\rho(X)$ can be approximated by some other algorithm, such as NAÏVEGREEDY. In line 4, A($\sigma, \lambda_{\text{mid}}\rho$) denotes that algorithm A is being called to maximize the objective $\sigma - \lambda_{\text{mid}}\rho$. As we can see, B employs a binary search scheme to restrict the interval $\left[0, \frac{\max_{X\subseteq U}\sigma(X)}{\min_{X\subseteq U}\rho(X)}\right]$. Since

$$\max_{S\subseteq U}\{\sigma(S) - \lambda_{\max}\rho(S)\} \le 0 \Rightarrow \lambda_{\max} \ge \frac{\sigma(S)}{\rho(S)}, \forall S \subseteq U.$$

$$\begin{aligned}\frac{\sigma(S^G)}{\rho(S^G)} &\ge \lambda_{\min} \ge \frac{1}{1+\epsilon}\lambda_{\max}\\ &\ge \frac{1}{1+\epsilon}\max_{S\subseteq U}\frac{\sigma(S)}{\rho(S)}\end{aligned}$$

□

Thus, we have shown that DS can be used to approximate RS, but as one would suspect from the previous section, the same does not hold the other way around. Indeed, according to [1], there exists a DS function instance that cannot be expressed as an RS function. This verifies what we have already discussed about the inherent difficulty in the approximation of DS Maximization, as well as the fact that the two problems are not equivalent.

Next, we show that there can be a "weak" equivalence from one problem to the other. Firstly, we assume that some approximation algorithm exists for either problem. Let $\alpha \in [0, 1]$. Given two set functions $\sigma : 2^U \to \mathbb{R}, \sigma(S) \ge 0, \forall S \subseteq U$ and $\rho : 2^U \to \mathbb{R}, \rho(S) > 0, \forall S \subseteq U$, we consider the following approximation problems:

α-APPROXIMATION OF DIFFERENCE

Input: $U = \{u_1, \ldots, u_n\}$, functions σ, ρ.

Output: $S^* \subseteq U$ such that

$$\sigma(S^*) - \rho(S^*) = \max_{S\subseteq U}\{\alpha\sigma(S) - \rho(S)\}. \tag{3.5}$$

and

α-APPROXIMATION OF RATIO

Input: $U = \{u_1, \ldots, u_n\}$, functions σ, ρ.

Output: $S^* \subseteq U$ such that

$$\frac{\sigma(S^*)}{\rho(S^*)} = \max_{S\subseteq U}\left\{\alpha\frac{\sigma(S)}{\rho(S)}\right\}. \tag{3.6}$$

Now, we present a fully polynomial-time approximations scheme (FPTAS) for reducing each problem to the other, proposed by [28].

Theorem 3.10. α-APPROXIMATION OF DIFFERENCE and α-APPROXIMATION OF RATIO are equivalent.

Proof. Firstly, we show that α-APPROXIMATION OF DIFFERENCE can be reduced to α-APPROXIMATION OF RATIO. Suppose that we have an algorithm A for α-APPROXIMATION OF RATIO. Also, assume that we have run some optimization algorithm, such as NAÏVE-GREEDY, and found $S_\sigma = \arg\max_{X \subseteq U} \sigma X$ and $S_\rho = \arg\min_{X \subseteq U} \rho X$. Then we can use the algorithm below to solve α-APPROXIMATION OF DIFFERENCE:

```
Require: $U, \sigma, \rho, \epsilon > 0$, A.
1: $\lambda_{\text{min}} \leftarrow \sigma(S_\rho) - \rho(S_\rho)$, $\lambda_{\text{max}} \leftarrow \alpha\sigma(S_\sigma) - \rho(S_\rho)$, $S \leftarrow S_\rho$.
2: while $\lambda_{\text{max}} - \lambda_{\text{min}} > \epsilon$ do
3:     $\lambda_{\text{mid}} \leftarrow \frac{\lambda_{\text{max}} + \lambda_{\text{min}}}{2}$
4:     $T \leftarrow$A$(\sigma, \lambda_{\text{mid}} + \rho)$
5:     if $\sigma(T) - \rho(T) \leq \lambda_{\text{mid}}$ then
6:         $\lambda_{\text{max}} \leftarrow \lambda_{\text{mid}}$
7:     else
8:         $\lambda_{\text{min}} \leftarrow \lambda_{\text{mid}}$
9:         $S \leftarrow T$
10: return $S$
```

Indeed, at every iteration we have $\lambda_{\text{min}} \leq \sigma(S) - \rho(S)$. Additionally,

$$\sigma(T) - \rho(T) \leq \lambda_{\text{max}} \Rightarrow \frac{\sigma(T)}{\lambda_{\text{max}} + \rho(T)} \leq 1$$

and, by definition of A for all $S' \in 2^U$,

$$\alpha\frac{\sigma(S')}{\lambda_{\text{max}} + \rho(S')} \leq \frac{\sigma(T)}{\lambda_{\text{max}} + \rho(T)} \leq 1 \Rightarrow \lambda_{\text{max}} \geq \alpha\sigma(S') - \rho(S').$$

Thus, for all $S' \in 2^U$,

$$\alpha\sigma(S') - \rho(S') \leq \lambda_{\text{max}} \leq \lambda_{\text{min}} + \epsilon \leq \sigma(S) - \rho(S) + \epsilon.$$

Conversely, let A be an algorithm for α-APPROXIMATION OF DIFFERENCE and consider the following algorithm:

```
Require: $U, \sigma, \rho, \epsilon > 0$, A.
1: $\lambda_{\text{min}} \leftarrow \sigma(S_\rho)/\rho(S_\rho)$, $\lambda_{\text{max}} \leftarrow \alpha\sigma(S_\sigma)/\rho(S_\rho)$, $S \leftarrow S_\rho$.
2: while $\lambda_{\text{max}} - \lambda_{\text{min}} > \epsilon$ do
3:     $\lambda_{\text{mid}} \leftarrow \frac{\lambda_{\text{max}} + \lambda_{\text{min}}}{2}$
4:     $T \leftarrow$A$(\sigma, \lambda_{\text{mid}}\rho)$
5:     if $\sigma(T)/\rho(T) \leq \lambda_{\text{mid}}$ then
6:         $\lambda_{\text{max}} \leftarrow \lambda_{\text{mid}}$
7:     else
8:         $\lambda_{\text{min}} \leftarrow \lambda_{\text{mid}}$
9:         $S \leftarrow T$
10: return $S$
```

By similar arguments, for all $S' \in 2^U$ we get that

$$\alpha\frac{\sigma(S')}{\rho(S')} \leq \lambda_{\text{max}} \leq \lambda_{\text{min}} + \epsilon \leq \frac{\sigma(S)}{\rho(S) + \epsilon},$$

which concludes our proof. □

3.5 Additive Smoothing Ratio Maximization

Although the ratio σ_N/σ_V considers what fraction of all affected nodes are vulnerable (note that $\sigma_N/\sigma_V = \sigma/\sigma_V - 1$), it is undefined for every seed set S such that $\sigma_V(S) = 0$, thus it cannot distinguish between two seed sets with S_1, S_2 with $\sigma_V(S_1) = \sigma_V(S_2) = 0$ and $\sigma_N(S_1) > \sigma_N(S_2)$, let alone that RS MAX is not concerned with providing a seed set of bound size.

Such issues that make our algorithms fail may arise even in the smallest of networks. For instance in Example 3.1, if we mark u_5 and u_6 as vulnerable, GREEDRATIO outputs the seed set $S^G = \{u_2, u_3\}$ which gives the value $\frac{\sigma_N(S^G)}{\sigma_V(S^G)} = \frac{3}{2}$. In the meantime there is another optimal solution $\frac{\sigma_N(\{u_1,u_2\})}{\sigma_V(\{u_1,u_2\})} = 3$ that the algorithm does not consider due to the fact that $\sigma_V(u_1) = 0$.

In order to avoid these inconveniences while keeping all the benefits of the ratio σ_N/σ_V, Chen et al. [5] suggested applying additive smoothing to the ratio, resulting in the *additive smoothing ratio* (ASR),

$$ASR(S, c) = \frac{\sigma_N(S) + c}{\sigma_V(S) + c},$$

where $c > 0$ is a selected constant. We redefine RS maximization as ASR Maximization (ASR MAX):

ASR MAXIMIZATION (ASR MAX)

Input:$G = (U, E)$, partition N, V of U, $\sigma_N, \sigma_V : 2^N \to \mathbb{R}$, parameters $k \in \mathbb{N}$ and $c > 0$.

Output: $\emptyset \subset S^* \subset N$, $|S^*| \leq k$ such that

$$\text{ASR}(S^*, c) = \max_{\emptyset \subset S \subseteq N} \{\text{ASR}(S, c) \mid |S| \leq k\}. \tag{3.7}$$

Theorem 3.11. The ASR MAX problem is NP-hard.

Proof. As we have already shown, IMP is NP-hard under ICM. We will show that the same holds for ASR MAX by restriction. Suppose that we have an instance $\mathcal{I}_{IM}$ of IMP. Then we can create a corresponding instance $\mathcal{I}_{ASR-MAX}$ with no vulnerable nodes ($N = U, V = \emptyset$) for ASR MAX in polynomial time. Since all nodes are non-vulnerable, a solution for $\mathcal{I}_{\text{ASR\_MAX}}$ also gives the maximum spread $\sigma = \sigma_N$ and is of size at most k, i.e., it also constitutes a solution for $\mathcal{I}_{\text{IM}}$. The inverse can be proved by a similar argument. □

Theorem 3.12. ASR is non-monotone and neither is submodular.

Proof. We will prove this claim using a counterexample. Consider the network from Example 3.1, where u_5 and u_6 are vulnerable, and let $c = 1$. Also, let $S = \{u_1\}$ and $T = \{u_1, u_2\}$. Then

$$\text{ASR}(S, 1) = \frac{\sigma_N(S) + 1}{\sigma_V(S) + 1} = \frac{2 + 1}{0 + 1} = 3$$

and

$$\text{ASR}(T, 1) = \frac{\sigma_N(T) + 1}{\sigma_V(T) + 1} = \frac{3 + 1}{1 + 1} = 2.$$

Thus, we have $S \subset T$ and $\text{ASR}(S, 1) > \text{ASR}(T, 1)$ which means that ASR is non-monotone.

Moreover,

$$\begin{aligned}\text{ASR}(S \mid u_3, 1) &= \text{ASR}(\{u_1, u_3\}, 1) - \text{ASR}(\{u_1\}, 1) \\ &= \frac{\sigma_N(\{u_1, u_3\}) + 1}{\sigma_V(\{u_1, u_3\}) + 1} - \frac{\sigma_N(u_1) + 1}{\sigma_V(u_1) + 1} = \frac{3+1}{2+1} - \frac{2+1}{0+1} \\ &= \frac{4}{3} - 3 = -\frac{5}{3}\end{aligned}$$

and

$$\begin{aligned}\text{ASR}(T \mid u_3, 1) &= \text{ASR}(\{u_1, u_2, u_3\}, 1) - \text{ASR}(\{u_1, u_2\}, 1) \\ &= \frac{\sigma_N(\{u_1, u_2, u_3\}) + 1}{\sigma_V(\{u_1, u_2, u_3\}) + 1} - \frac{\sigma_N(\{u_1, u_2\}) + 1}{\sigma_V(\{u_1, u_2\}) + 1} = \frac{4+1}{2+1} - \frac{3+1}{1+1} \\ &= \frac{5}{3} - 2 = -\frac{1}{3},\end{aligned}$$

i.e., $S \subset T$ and $\text{ASR}(S \mid u_3, 1) < \text{ASR}(T \mid u_3, 1)$. Hence, ASR is not submodular. □

By the previous theorem, it is clear that IMP and ASR MAX are two fundamentally different problems. Next, we present three algorithms for maximizing ASR, where we use the notation

$$\text{ASR}(S \mid u, c) := \text{ASR}(S \cup \{u\}, c) - \text{ASR}(S, c).$$

3.5.1 ASR Greedy

We once again start with the most basic greedy approach for maximizing ASR.

Algorithm 8 ASRGREEDY

```
Require: $U = N \cup V$, $\sigma_N, \sigma_V$, parameter $k$, constant $c$.
1: $S^0 \leftarrow \emptyset, N^0 \leftarrow N, t \leftarrow 1$
2: while $t \leq k$ do
3:     Select $u_t \in N^{t-1}$ s.t. $\text{ASR}(S^{t-1} \mid u_t, c) = \max_{u \in N^{t-1}} \{\text{ASR}(S^{t-1} \mid u, c)\}$.
4:     $S^t \leftarrow S^{t-1} \cup \{u_t\}, N^t \leftarrow N^{t-1} \setminus \{u_t\}$
5:     $t \leftarrow t + 1$
6: return $S^G \leftarrow \arg\max_{S \in \{S^1, \ldots, S^k\}} \text{ASR}(S, c)$
```

ASRGREEDY is a natural heuristic similar to GREEDRATIO that limits the influence maximization to vulnerable nodes, with the difference that it creates a seed set of bounded size k. Indeed, the algorithm performs k iterations, selecting each time the node u that maximizes the additive smoothing ratio of the marginal gains in σ_N and σ_V. Nonetheless, since ASR is non-monotone, a larger ASR can be obtained at any intermediate step, i.e., it is possible that $\text{ASR}(S^t, c) > \text{ASR}(S^{t+n}, c)$, for some natural $n \leq k - t$. For this, the algorithm stores every computed seed set from all iterations and returns the one which provides the maximum ASR. ASRGREEDY performs $O(|N| \cdot k)$ evaluations of the spread function.

3.5.2 Sandwich Approximation Algorithm with Spread Bounds

Before presenting Sandwich Approximation Algorithm with Spread Bounds (SAS), we introduce the bound functions of ASR that the algorithm utilizes.

Lower and Upper Bounds of ASR

We define the lower and upper bound functions of ASR as follows:

$$\mathrm{ASR}^L(S,c) = \frac{\sigma_N(S)+c}{|V|+c}, \quad \mathrm{ASR}^U(S,c) = \frac{\sigma_N(S)+c}{c}.$$

It is trivial to see that both functions are non-negative and also that they indeed constitute bound functions of ASR: $\mathrm{ASR}^L(S,c) \leq \mathrm{ASR}(S,c) \leq \mathrm{ASR}^U(S,c)$, since $0 \leq \sigma_V(S) \leq |V|, \forall S$.

Lemma 3.13. ASR^L and ASR^U are monotone submodular for any parameter $c > 0$.

Proof. First, we show the monotonicity and submodularity of ASR^L. Consider two subsets $S \subseteq T \subseteq N$. Then, since σ_N is monotone, we have that $\sigma_N(S) \leq \sigma_N(T) \Rightarrow \mathrm{ASR}^L(S,c) \leq \mathrm{ASR}^L(T,c)$, i.e., ASR^L is monotone. Furthermore, since σ_N is submodular, we also have that for any $u \in N \setminus T$,

$$\sigma_N(S \cup \{u\}) - \sigma_N(S) \geq \sigma_N(T \cup \{u\}) - \sigma_N(T).$$

With some manipulation (adding and subtracting c, dividing by $|V| + c > 0$), we get

$$\frac{\sigma_N(S \cup \{u\})+c}{|V|+c} - \frac{\sigma_N(S)+c}{|V|+c} \geq \frac{\sigma_N(T \cup \{u\})+c}{|V|+c} - \frac{\sigma_N(T)+c}{|V|+c} \Rightarrow$$

$$\mathrm{ASR}^L(S \cup \{u\}, c) - \mathrm{ASR}^L(S,c) \geq \mathrm{ASR}^L(T \cup \{u\}, c - \mathrm{ASR}^L(T,c) \Rightarrow$$

$$\mathrm{ASR}^L(S \mid u, c) \geq \mathrm{ASR}^L(T \mid u, c),$$

which means that ASR^L is submodular.

The proof for ASR^U is similar. □

As we have shown, ASR, ASR^L and ASR^U meet the criteria allowing us to apply the SA strategy (2.3). Before doing so, we need to also re-define the notion of *dummy elements*: an element u is considered *dummy* if for every subset $S \subseteq N$ it holds that $\sigma_N(S \cup \{u\}) = \sigma_N(S)$ and $\sigma_V(S \cup \{u\}) = \sigma_V(S)$, that is, it offers zero marginal gain to any subset $S \subseteq N$, w.r.t. both spread functions σ_N, σ_V. Now, we will apply the SA strategy to SubsampleGreedy (2.2).

Algorithm 9 SAS

```
Require: $U = N \cup V$, $\sigma_N, \sigma_V$, parameter $k$, constant $c$.
 1: $S \leftarrow N$                                                    ▷ Phase I
 2: $D \leftarrow \{v_1, \dots, v_k\}$, where $v_i$ is a dummy element $\forall i \in [k]$
 3: $\mathcal{N} \leftarrow N$
 4: while $|\mathcal{N}|/k \notin \mathbb{N}$ do
 5:     $\mathcal{N} \leftarrow \mathcal{N} \cup \{u\}$, where $u \notin D$ is a dummy element
                                                                       ▷ Phase II
 6: $t \leftarrow 1, S^O \leftarrow \emptyset, S^L \leftarrow \emptyset, S^U \leftarrow \emptyset$
 7: while $t \leq k$ do
 8:     $R \leftarrow$ uniform random sample of $\mathcal{N}$ with $|\mathcal{N}|/k$ elements.
 9:     Select a random element $v \in D$, $R \leftarrow R \cup \{v\}$
10:     Select $u^O \in R$ s.t. $\text{ASR}(S^O \mid u^O, c) = \max_{u \in R} \{\text{ASR}(S^O \mid u, c)\}$
11:     $S^O \leftarrow S^O \cup \{u^O\}$
12:     Select $u^L \in R$ s.t. $\text{ASR}^L(S^L \mid u^L, c) = \max_{u \in R} \{\text{ASR}^L(S^L \mid u, c)\}$
13:     $S^L \leftarrow S^L \cup \{u^L\}$
14:     Select $u^U \in R$ s.t. $\text{ASR}^U(S^U \mid u^U, c) = \max_{u \in R} \{\text{ASR}^U(S^U \mid u, c)\}$
15:     $S^U \leftarrow S^U \cup \{u^U\}$
16:     $t \leftarrow t + 1$
                                                                       ▷ Phase III
17: $S \leftarrow \arg\max_{S \in \{S^O, S^L, S^U\}} \{\text{ASR}(S, c)\}$
18: $S \leftarrow S \setminus \{v \mid v \text{ is dummy}\}$
19: return $S^G \leftarrow S$
```

The algorithm operates in three phases:

Phase I A set D of k dummy elements is created. Then additional dummy elements (not from D) are added to $\mathcal{N}$, initially set to contain all non-vulnerable nodes, until $|\mathcal{N}|$ is a multiple of k.

Phase II A random sample of $|\mathcal{N}|/k$ elements of $\mathcal{N}$ is created and a dummy element from D is chosen. From these, the algorithm selects the elements u^O, u^L, u^U that contribute the largest marginal gain w.r.t. ASR, ASR^L, ASR^U, respectively. u^O, u^L, u^U are added to S^O, S^L, S^U (initially empty) and the process repeats k times.

Phase III ASR is computed with all three sets S^O, S^L, S^U, the best one is selected and its dummy elements are removed. The algorithm returns this set.

Theorem 3.14. SAS returns a set S^G such that

$$\mathbb{E}[\text{ASR}(S^G, c)] \geq \frac{c}{|V| + c}\left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c),$$

where S^* is an optimal solution of size at most k w.r.t. ASR and $\mathbb{E}[\text{ASR}(S, c)]$ is the expected value of ASR over all possible outputs of SAS.

Proof. See Appendix C. ☐

Proposition 3.15. SAS performs $O(|N|)$ evaluations of the spread function.

Proof. Since SubsampleGreedy performs $O(|N|)$ evaluations of the spread function σ, ASR needs two evaluations of σ and SAS basically executes the algorithm three times, the total number of evaluations stays linear. □

It is worth mentioning that, in the same sense that SAS applies SubsampleGreedy to ASR and its bound functions, it can also apply ASRGreedy (3.5.1) using the SA strategy. We will call this variation SAS-Greedy:

Algorithm 10 SAS-Greedy

```
Require: $U = N \cup V$, $\sigma_N, \sigma_V$, parameter $k$, constant $c$.
 1: $t \leftarrow 0, S^O \leftarrow \emptyset, S^L \leftarrow \emptyset, S^U \leftarrow \emptyset, \mathcal{N}^O \leftarrow N, \mathcal{N}^L \leftarrow N, \mathcal{N}^U \leftarrow N$
 2: while $t < k$ do
 3:     Select $u^O \in \mathcal{N}^O$ s.t. $\mathrm{ASR}(S^O \mid u^O, c) = \max_{u \in \mathcal{N}^O} \{\mathrm{ASR}(S^O \mid u, c)\}$
 4:     $S^O \leftarrow S^O \cup \{u^O\}, \mathcal{N}^O \leftarrow \mathcal{N}^O \setminus \{u^O\}$
 5:     Select $u^L \in \mathcal{N}^L$ s.t. $\mathrm{ASR}^L(S^L \mid u^L, c) = \max_{u \in \mathcal{N}^L} \{\mathrm{ASR}^L(S^L \mid u, c)\}$
 6:     $S^L \leftarrow S^L \cup \{u^L\}, U^L \leftarrow U^L \setminus \{u^L\}$
 7:     Select $u^U \in \mathcal{N}^U$ s.t. $\mathrm{ASR}^U(S^U \mid u^U, c) = \max_{u \in \mathcal{N}^U} \{\mathrm{ASR}^U(S^U \mid u, c)\}$
 8:     $S^U \leftarrow S^U \cup \{u^U\}, \mathcal{N}^U \leftarrow \mathcal{N}^U \setminus \{u^U\}$
 9:     $t \leftarrow t + 1$
10: $S \leftarrow \arg\max_{S \in \{S^O, S^L, S^U\}} \{\mathrm{ASR}(S, c)\}$
11: return $S^G \leftarrow S$
```

Lemma 3.16. SAS-Greedy outputs a solution S^G such that

$$\mathrm{ASR}(S^G, c) \geq \max\left\{\frac{c}{\sigma_V(S^U, c) + c}, \frac{\sigma_V(S^*, c) + c}{|V| + c}\right\}\left(1 - \frac{1}{e}\right)\mathrm{ASR}(S^*, c),$$

where S^* is an optimal solution of size at most k w.r.t. ASR.

Proof. From 2.3, we have that

$$\mathrm{ASR}(S^G, c) \geq \max\left\{\frac{\mathrm{ASR}(S^U, c)}{\mathrm{ASR}^U(S^U, c)}, \frac{\mathrm{ASR}^L(S^*, c)}{\mathrm{ASR}(S^*, c)}\right\}\left(1 - \frac{1}{e}\right)\mathrm{ASR}(S^*, c).$$

Replacing ASR^L and ASR^U with their definitions, we obtain the desired approximation guarantee. □

SAS-Greedy performs $O(|N| \cdot k)$ evaluations of the spread function, which makes it considerably slower than SAS for large values of k.

3.5.3 Iterative Subsample Algorithm with Spread Bounds

Iterative Subsample with Spread Bounds (ISS), like SAS, combines the sandwich approximation technique along with subsampling, but it employs a little more sophisticated bound functions.

The lower-bound function with parameter $Y \subseteq N$ is defined as follows:

$$\widetilde{\mathrm{ASR}^L}(S, c, Y) = \frac{\sigma_N(S) + c}{\widehat{\sigma_{V,Y}}(S) + c},$$

much like ASR, but instead of σ_V in the denominator, it uses σ_V's modular upper bound with parameter Y:

$$\widehat{\sigma_{V,Y}}(S) = \sigma_V(Y) + \sum_{u \in S \setminus Y} \sigma_V(u) - \sum_{u \in Y \setminus S} (\sigma_V(Y) - \sigma_V(Y \setminus \{u\})).$$

For ASR's upper-bound $\widetilde{\text{ASR}}^U$ with parameter $Y \subseteq N$, we will use σ_V's lower modular bound with the same parameter Y, i.e.,

$$\widetilde{\text{ASR}}^U(S, c, Y) = \frac{\sigma_N(S) + c}{\widetilde{\sigma_{V,Y,\pi^Y}}(S) + c},$$

where

$$\widetilde{\sigma_{V,Y,\pi^Y}}(S) = \sum_{u \in S} \sigma_{V,Y,\pi^Y}(u)$$

and π^Y is a random permutation of Y. As stated in 3.1,

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

where π^Y_u is a prefix of π^Y up to u and $\pi^Y_{u^-}$ is the same prefix without u.

Lemma 3.17. $\widetilde{\text{ASR}}^L$ and $\widetilde{\text{ASR}}^U$ are non-monotone and non-submodular.

Proof. We will show this again using 3.1 as a means of counterexample with u_5, u_6 being vulnerable. Let $Y = \{u_2, u_3\}$, $\pi^Y = (u_3, u_2)$, $S = \{u_1, u_2\}$ and $T = \{u_1, u_2, u_3\}$. Then

$$\widetilde{\text{ASR}}^L(S, 1, Y) = \frac{\sigma_N(S) + 1}{\sigma_V(u_1) + \sigma_V(u_2) + 1} = 2$$

and

$$\widetilde{\text{ASR}}^L(T, 1, Y) = \frac{\sigma_N(T) + 1}{\sigma_V(Y) + \sigma_V(u_1) + 1} = \frac{5}{3}.$$

Hence, since $S \subset T$ and $\widetilde{\text{ASR}}^L(S, 1, Y) > \widetilde{\text{ASR}}^L(T, 1, Y)$, $\widetilde{\text{ASR}}^L$ is non-monotone. Similarly,

$$\widetilde{\text{ASR}}^U(S, 1, Y) = \frac{\sigma_N(S) + 1}{\sigma_V(\{u_2, u_3\}) - \sigma_V(u_3) + 1} = 4$$

and

$$\widetilde{\text{ASR}}^U(T, 1, Y) = \frac{\sigma_N(T) + 1}{\sigma_V(\{u_2, u_3\}) + 1} = \frac{5}{3} < \widetilde{\text{ASR}}^U(S, 1, Y),$$

so $\widetilde{\text{ASR}}^U$ is also non-monotone.

Furthermore, if we set $S = \{u_3, u_4\} \subset \{u_1, u_3, u_4\} = T$, then

$$\widetilde{\text{ASR}}^L(S \mid u_2, 1, Y) = \ldots = -\frac{1}{6}$$

and

$$\widetilde{\text{ASR}}^L(T \mid u_2, 1, Y) = \ldots = \frac{1}{3} > \widetilde{\text{ASR}}^L(S \mid u_2, 1, Y),$$

so $\widetilde{\text{ASR}}^L$ is not submodular. Similarly $\widetilde{\text{ASR}}^U$ is proven to be non-submodular as well. □

Algorithm 11 ISS

```
Require: $U = N \cup V$, $\sigma_N, \sigma_V$, parameter $k$, constant $c$.
1: $S \leftarrow N, S^p \leftarrow \emptyset$
2: while true do                                                    ▷ Phase I
3:     $D \leftarrow \{v_1, \ldots, v_k\}$, where $v_i$ is a dummy element $\forall i \in [k]$
4:     $\mathcal{N} \leftarrow N$
5:     while $|\mathcal{N}|/k \notin \mathbb{N}$ do
6:         $\mathcal{N} \leftarrow \mathcal{N} \cup \{u\}$, where $u \notin D$ is a dummy element
                                                                    ▷ Phase II
7:     $t \leftarrow 1, S^O \leftarrow \emptyset, S^L \leftarrow \emptyset, S^U \leftarrow \emptyset$
8:     while $t \leq k$ do
9:         $R \leftarrow$ uniform random sample of $\mathcal{N}$ with $|\mathcal{N}|/k$ elements.
10:        Select a random element $v \in D$, $R \leftarrow R \cup \{v\}$
11:        $u^O \in R$ s.t. $\text{ASR}(S^O \mid u^O, c) = \max_{u \in R} \{\text{ASR}(S^O \mid u, c)\}$
12:        $S^O \leftarrow S^O \cup \{u^O\}$
13:        $u^L \in R$ s.t. $\widetilde{\text{ASR}^L}(S^L \mid u^L, c, S^p) = \max_{u \in R} \left\{\widetilde{\text{ASR}^L}(S^L \mid u, c, S^p)\right\}$
14:        $S^L \leftarrow S^L \cup \{u^L\}$
15:        $u^U \in R$ s.t. $\widetilde{\text{ASR}^U}(S^U \mid u^U, c, \pi^{S^p}) = \max_{u \in R} \left\{\widetilde{\text{ASR}^U}(S^U \mid u, c, \pi^{S^p})\right\}$
16:        $S^U \leftarrow S^U \cup \{u^U\}$
17:        $t \leftarrow t + 1$
                                                                    ▷ Phase III
18:    $S \leftarrow \arg\max_{S \in \{S^O, S^L, S^U\}} \{\text{ASR}(S, c)\}$
19:    $S \leftarrow S \setminus \{v \mid v \text{ is dummy}\}$
20:    if $\text{ASR}(S, c) \leq \text{ASR}(S^p, c)$ then
21:        break
22:    $S^p \leftarrow S$
23: return $S^G \leftarrow S$
```

ISS too works in three phases:

Phase I As in SAS, this is the phase of dummy element creation, as well as N's expansion to $\mathcal{N}$ with enough dummy elements so that it's size becomes divisible by k.

Phase II A random sample of $|\mathcal{N}|/k$ elements of $\mathcal{N}$ is created and a dummy element from D is chosen. From these, the algorithm selects the elements u^O, u^L, u^U that contribute the largest marginal gain w.r.t. ASR, $\widetilde{\text{ASR}^L}$, $\widetilde{\text{ASR}^U}$, respectively, and adds them to S^O, S^L, S^U (initially empty), for k iterations. Throughout these iterations, the bound functions are evaluated using as parameter the same set S^p (and some random permutation π^{S^p} of it), which is actually the seed set constructed in the previous round (initially empty).

Phase III ASR is computed with all three sets S^O, S^L, S^U, the best one is selected and its dummy elements are removed. If this current seed set provides an ASR no bigger than S^p, the algorithms returns it, otherwise it makes it the parameter set and keeps looking for another seed set with better ASR.

Hence, we see that another fundamental difference between SAS and ISS is that the latter uses parameter-dependent bounds that get better at every round by making use of an improved parameter, one that had been constructed in the previous round as the best solution up to that point.

If ISS terminates in I iterations, then it will perform $O(|N| \cdot I)$ evaluations of the spread function.

Other variations of ISS have been proposed, such as ISS^U that applies SUBSAMPLEGREEDY only to $\widetilde{\text{ASR}}^U$ instead of all three functions, as well as ISS-GREEDY that applies ASRGREEDY instead of SUBSAMPLEGREEDY. Whilst the latter does not provide any approximation guarantees and the first one doesn't improve complexity in theory, in practice both seem to be quite effective [5].

3.6 A PageRank approach

In this section, we will deviate a little bit from our previous analysis in order to investigate an alternative approach that operates under the LTM.

PageRank [2] is a link analysis algorithm by Google that ranks pages in terms of relevance in order to sort search results. The World Wide Web is modeled as a network $G = (U, E)$ whose edges represent hyperlinks and each page is represented by a node u, whose *PageRank score* $\text{PR}_G(u)$ indicates the likelihood (limiting probability) that u is ultimately reached through random surfing:

$$\text{PR}_G(u) = \frac{1-d}{|U|} + d \cdot \sum_{v \in n^-(u)} \frac{\text{PR}_G(v)}{|n^+(v)|},$$

where d is a *damping factor* usually set to 0.85 and $n^+(v)$ is the set of the *out-neighbors* of v. Note that $\text{PR}_G(u)$ is computed recursively.

In this section we study a direction introduced in [21], where our problem and the greedy approach are reformulated to optimize a measure inspired by the PageRank model. In this setting, hyperlinks are directed edges and pages are users. The PageRank score of a user is the probability that the user gets activated, in the same manner that a page is reached through a series of hyperlinks from any other page. The goal is to prevent the diffusion of information to vulnerable nodes through edge deletion but at the same time to avoid deteriorating the network's ability to propagate information. In other words, we aim to preserve (as much as possible) the PageRank distribution for all non-vulnerable while minimizing the activation probability of vulnerable nodes.

The measure used in [21] is called PageRank Harm (PRH). The PRH of an edge $e = (u, v)$ is defined as

$$\text{PRH}(e) = d \cdot \frac{\text{PR}_G(u)}{|n^+(u)|},$$

which is basically the contribution of u to the PageRank score of v. We also define $\text{PRH}(F) = \sum_{e \in F} \text{PRH}(e), \forall F \subseteq E$. Deleting e would have an impact to the PageRank score of all nodes. More precisely, if we denote by $\delta(u') = \text{PR}_G(u') - \text{PR}_{G \setminus e}(u')$, then

$$\delta(u') = \begin{cases} \text{PRH}(e) + d \cdot \sum\limits_{\substack{w \in n^-(u') \\ w \neq u'}} \frac{d(w)}{|n^+(w)|}, & \text{if } u' = v \\ \text{PRH}(e) - d \cdot \frac{\text{PR}_{G \setminus e}(u)}{|n^+(u)|-1} + d \cdot \sum\limits_{\substack{w \in n^-(u') \\ w \neq u'}} \frac{d(w)}{|n^+(w)|}, & \text{if } u' \in n^+(u), u' \neq v \\ d \cdot \sum\limits_{w \in n^-(u')} \frac{d(w)}{|n^+(w)|}, & \text{if } u' \notin n^+(u) \end{cases}$$

Note that $d(u')$ is also computed recursively, which means that it decreases exponentially with the length of the path from u to w. Indeed, let

$$\mathcal{P}_{u,w} := u \to v \to u_1 \to \ldots \to u_{\nu-2} \to w$$

be a simple path of length $\nu = |\mathcal{P}_{u,w}|$ from u to w, starting with edge e. Then

$$\delta(w) = d^{\nu-1} \frac{\delta(v)}{|n^+(v)| \cdot \prod_{i=1}^{\nu-2} |n^+(u_i)|}$$

and $d < 1$.

Before formulating the problem, we list some properties of the LTM from [11] that we are going to need next. The *path probability* of a path $\mathcal{P}_{u,v} := u_0 \to u_1 \to \ldots \to u_{\nu-1}$, where $u_0 := u$ and $u_{\nu-1} = v$ is defined as $\Pr[\mathcal{P}_{u,v}] = \prod\limits_{i=0}^{\nu-2} p_{u_i,u_{i+1}}$. Also, let $\mathcal{P}_{S,v}$ be the set of all simple paths from all nodes in S to v that do not contain any other node from S, and let G_v be the *activation graph* of v induced by $\mathcal{P}_{S,v}$. We also define $G_T = \bigcup\limits_{v \in T} G_v, \forall T \subseteq U$, as the *activation graph* of T. Furthermore,

$$\Pr[v, \mathcal{P}_{S,v}, F] = \sum_{\substack{\mathcal{P} \in \mathcal{P}_{S,v} \\ \mathcal{P} \cap F \neq \emptyset}} \Pr[\mathcal{P}]$$

denotes the *activation probability* of v by the paths in $\mathcal{P}_{S,v}$ that contain edges in F. Finally, the *aggregate path probability* for nodes in a subset $T \subseteq U$ and some $F \subseteq E$ is defined as

$$g_T(F) = \Pr\left[T, \bigcup_{v \in T} \mathcal{P}_{S,v}, F\right] = \sum_{v \in T} \min\{\Pr[v, \mathcal{P}_{S,v}, F], \Pr[v, \mathcal{P}_{S,v}, E] - \mathrm{p}\},$$

where $\mathrm{p} \in [0, 1]$ is the threshold and the *aggregate path probability gain* of e to F as

$$g_T(F \mid e) = g_T(F \cup \{e\}) - g_T(F).$$

Proposition 3.18. The function $g_T : 2^E \to \mathbb{R}$ is monotone non-decreasing submodular.

Proof. See Appendix C. □

We now define the problem using this terminology [21]:

PAGERANK EDGE DELETION (PED)

Input: Graph $G = (U, E)$, N, V such that $N \cup V = U$ and $N \cap V = \emptyset$, seed set $S \subseteq N$, $G_V = (U_V, E_V)$, PRH(e) for all $e \in E_V$, threshold p $\in [0, 1]$.
Output: $F^* \subseteq E_V$ such that

$$\text{PRH}(F^*) = \min_{F \subseteq E_V} \{\text{PRH}(F)\}$$
$$\text{and} \tag{3.8}$$
$$\Pr\left[V, \bigcup_{v\in V} \mathcal{P}_{S,v}, F^*\right] = \sum_{v\in V} (\Pr[v, \mathcal{P}_{S,v}, E_V] - \text{p}).$$

PED is NP-hard [21, 33]. Among other algorithms, the authors of [21] suggest AGGREGATE GREEDY EDGE DELETION (AGED).

Algorithm 12 AGED

```
Require: $G = (N \cup V, E)$, seed set $S$, activation graph $G_V = (U_V, E_V)$, threshold
    p $\in [0,1]$, damping factor $d$, PageRank distribution PR$_G$.
 1: for $e = (u, v) \in E_V$ do
 2:     PRH$(e) \leftarrow d \cdot \frac{\text{PR}_G(u)}{|n^+(u)|}$
 3: $F^0 \leftarrow \emptyset, t \leftarrow 1$
 4: while Pr $\left[V, \bigcup_{v\in V} \mathcal{P}_{S,v}, F^{t-1}\right] < \sum_{v\in V}(\Pr[v, \mathcal{P}_{S,v}, E] - \text{p})$ do
 5:     Reconstruct $G_{V,t} = (U_{V,t}, E_{V,t})$
 6:     Select $e_t \in E_{V,t}$ s.t. $\frac{\text{PRH}(e_t)}{g_V(F^{t-1}|e_t)} = \min_{e\in E_{V,t}} \left\{\frac{\text{PRH}(e)}{g_V(F^{t-1}|e)}\right\}$
 7:     $F^t \leftarrow F^{t-1} \cup \{e_t\}$
 8:     $t \leftarrow t + 1$
 9: Delete $F^t$ from $G$
10: return $F^G \leftarrow F^t$
```

Theorem 3.19 ([21, 33])**.** AGED returns a solution F^G such that

$$\text{PRH}(F^G) \leq (1 + \ln(\mu_D))\text{PRH}(F^*),$$

where F^* is the optimal solution and

$$\mu_D = \min\left\{\max_{\substack{e\in F^G\\ e\neq e_1}}\left\{\frac{g_V(e_1)}{g_V(e)}\right\}, \frac{\text{PRH}(e_T)}{g_V(e_T)}\Big/\frac{\text{PRH}(e_1)}{g_V(e_1)}, \frac{\Pr[V, \bigcup_{v\in V}\mathcal{P}_{S,v}, F^G]}{g_V(e_T)}\right\},$$

where e_1 is the first edge and e_T is the last edge added to F^G.

For the proof of this theorem, the following lemma will be of use:

Lemma 3.20. Let $0 < y_1 \leq y_2 \leq \ldots \leq y_n$ and $x_1 \geq x_2 \geq \ldots \geq x_n > 0$. If $S = \sum_{i=1}^{n-1} y_i(x_i - x_{i+1}) + y_n x_n$, then

$$S \leq \max_{i\in[n]}\{y_i x_i\}\left[1 + \ln\left(\min\left\{\frac{x_1}{x_n}, \frac{y_n}{y_1}\right\}\right)\right].$$

Proof. See Appendix C. $\square$

Next, we construct the proof of Theorem 3.19 based on the proof of [33].

Proof of Theorem 3.19. First we give an equivalent definition of PED in terms of a linear integer problem PED^I:

$$\text{PRH} = \min_{F^* \subseteq E_V} \{\text{PRH}(F^*)\}$$

such that

$$\sum_{e \in F^*} g_V(F \mid e) \geq g_V(E) - g_V(F), \forall F \subseteq E_V.$$

To see the equivalence, note that since $g_V(F \mid e)$ represents the probability that some path to V contains e, then $\sum_{e \in F^*} g_V(F \mid e)$ denotes the probability that V is activated by paths containing edges of F^* and therefore it should hold that

$$\sum_{e \in F^*} g_V(F \mid e) \geq |V| \cdot \text{p}. \tag{3.9}$$

On the other hand, the condition of PED is equivalent to

$$\text{Pr}[v, \mathcal{P}_{S,v}, E_V] - \text{p} \leq \text{Pr}[v, \mathcal{P}_{S,v}, F] \Rightarrow \text{p} \geq \text{Pr}[v, \mathcal{P}_{S,v}, E_V] - \text{Pr}[v, \mathcal{P}_{S,v}, F],$$

for all $v \in V$. By adding these inequalities and combining with 3.9 we get the desired condition.

Now we consider the following relaxation PED^L of PED^I:

$$\text{PRH} = \min_{F^* \subseteq E_V} \{\text{PRH}(F^*)\}$$

such that

$$\sum_{e \in F^*} g_V(F^t \mid e) \geq g_V(E_V) - g_V(F^t), t \in [T-1],$$

where T is the last iteration of the algorithm. Furthermore, we set

$$\mu_1 = \max_{\substack{e \in F^G \\ e \neq e_1}} \left\{ \frac{g_V(e_1)}{g_V(e)} \right\}, \mu_2 = \frac{\text{PRH}(e_t)}{g_V(e_t)} \Big/ \frac{\text{PRH}(e_1)}{g_V(e_1)} \text{ and } \mu_3 = \frac{\text{Pr}[V, \bigcup_{v \in V} \mathcal{P}_{S,v}, F^G]}{g_V(e_T)}.$$

Let $x^* = (x^1, x^2 - x^1, \ldots, x^T - x^{T-1})$, where $x^t = \frac{\text{PRH}(e_t)}{g_V(e_t)}$. For a given e, there exists a $t \leq T$ such that $g_V(S^{t-1} \mid e) > 0$ and $g_V(S^t \mid e) = 0$. Given that $0 < x^1 \leq \ldots \leq x^t$ and $g_V(F^0 \mid e) \geq \ldots \geq g_V(F^{t-1} \mid e) > 0$, if we set

$$S = x^1 \cdot g_V(F^0 \mid e) + \sum_{i=1}^{t-1} (x^{i+1} - x^i) \cdot g_V(F^i \mid e),$$

then from Lemma 3.20 we obtain

$$\begin{aligned} S &\leq \max_{i=[t]} \{x^i g_V(F^{i-1} \mid e) \left[1 + \ln\left(\min\left\{\frac{g_V(F^0 \mid e)}{g_V(F^{t-1} \mid e)}, \frac{x^t}{x^1}\right\}\right)\right] \\ &\leq \text{PRH}(e)[1 + \ln(\min\{\mu_1, \mu_2\})]. \end{aligned}$$

Hence $(1 + \ln(\min\{\mu_1, \mu_2\})^{-1} x^*$ is dual feasible for PED^L and therefore, if

$$S' = x^1 \left(g_V(E) - g_V(F^0)\right) + \sum_{i=1}^{T-1} (x^{i+1} - x^i) \left(g_V(E) - g_V(F^i)\right) - x^T (g_V(E) - g_V(F^T)),$$

we have

$$(1 + \ln(\min\{\mu_1, \mu_2\}))^{-1} S' \leq \text{PRH}(F^L) \leq \text{PRH}(F^*),$$

where F^L is an optimal solution for PED^L. But

$$\text{PRH}(F^G) = \sum_{t=1}^{T} \text{PRH}(e_t) = \sum_{t=1}^{T} x^t g_V(S^{t-1} \mid e_t) = S',$$

which implies that

$$\text{PRH}(F^G) \leq (1 + \ln(\min\{\mu_1, \mu_2\}) \text{PRH}(F^*). \tag{3.10}$$

Now we define a T-vector u^t that has all zeros except for its t-th coordinate which is x^t. Then

$$u^t \cdot (g_V(F^0 \mid e), \ldots, g_V(f^{T-1} \mid e)) = x^t g_V(F^{t-1} \mid e) \leq \text{PRH}(e),$$

so $u^t, t = 1, \ldots, T-1$ is dual feasible for PED^L. Then,

$$\begin{aligned} \max_{t \in [T]} u^t (g_V(E) - g_V(F^0), \ldots, g_V(E) - g_V(F^{T-1})) &= \max_{t \in [T]} \theta^t (g_V(E) - g_V(F^{t-1}) \\ &\leq \text{PRH}(F^L) \leq \text{PRH}(F^*). \end{aligned}$$

Since $1 < x^1 \leq \ldots \leq x^T$ and $g_V(E) - g_V(F^0) \geq \ldots \geq g_V(E) - g_V(F^{T-1}) > 0$, we can again apply the previous lemma and obtain

$$\text{PRH}(F^G) = S' \leq \max_t \{x^t (g_V(E) - g_V(F^{t-1}))\} \left[1 + \ln\left(\frac{g_V(E) - g_V(F^0)}{g_V(E) - g_V(F^{T-1})}\right)\right].$$

Hence,

$$\text{PRH}(F^G) \leq (1 + \ln \mu_3) \text{PRH}(F^*) \tag{3.11}$$

and 3.10 and 3.11 conclude the proof. □

CHAPTER 4 CONCLUSION & FUTURE WORK

In this study, we explored various algorithms for influence maximization in social networks, focusing on the challenge of protecting vulnerable users while maximizing the spread of influence. The analysis covered both classical and innovative strategies, providing insights into their effectiveness and computational efficiency.

We began with DSMax, introducing an intuitive algorithm and a more sophisticated method that leverages modular bounds. This approach provided a foundation for understanding the dynamics of influence maximization in the presence of vulnerable nodes. Next, we examined RSMax by deploying GreedRatio and MMaxRatio algorithms. GreedRatio stands out for its simplicity and efficiency, offering a curvature-dependent approximation factor that makes it a more than satisfactory choice for general influence maximization tasks. In comparison, MMaxRatio uses a more intricate Minorization-Maximization approach, achieving a more complex approximation factor than GreedRatio, but potentially more refined solutions.

The introduction of ASR aimed to enhance the evaluation metric for our problem. SAS and ISS both aim to refine the approach to influence maximization under the constraints of vulnerable users. SAS offers a practical balance of efficiency and accuracy, its approximation factor and complexity making it a strong choice for many applications. ISS, however, takes SAS's main idea a step further by iteratively improving the parameter bounds used in each round. Although it does not provide a straightforward approximation guarantee, it is a potentially more powerful algorithm for achieving higher accuracy through iterative refinement.

Comparatively, while GreedRatio may initially appear more straightforward and robust, its reliance on a simpler measure underscores its potential limitations compared to the more nuanced methodologies of SAS and ISS. On the other hand, SAS might not do as well in terms of approximation, but appears to be more efficient. Likewise, ISS's use of tighter bounds may seem to provide more accurate solutions, but not an approximation guarantee. In conclusion, all algorithms have their weaknesses but nevertheless contribute their particular value to our repertoire, and their selection should depend on the specific requirements of the problem instance we are addressing.

Future research should empirically validate the proposed algorithms in real-world, large social networks to understand their practical performance and scalability. Conducting experiments with diverse datasets would offer valuable insights and benchmarks for these methods, possibly allowing for a more individualized design based on the specific characteristic of particular datasets. Naturally, further refinement of the algorithms in a more general sense to enhance scalability and efficiency, especially for large-scale social networks, represents another critical direction in developing more sophisticated approximation methods. Yet another apparent direction is to develop more advanced behavioral analysis and prediction techniques leveraging deep learning and machine learning approaches, in order to understand how users behave within the network or to even explore new networks.

Moreover, exploring dynamic social networks, where the structure and relationships between users evolve over time, presents an important challenge. Several social and environmental factors can change a user's state at any time, so incorporating temporal changes through interactive strategies into the influence maximization model could lead to more accurate and effective strategies for long-term influence propagation. In addition to accuracy and efficiency, the social and ethical implications of influence maximization are also crucial to reinforcing the social responsibility factor, especially in protecting vulnerable users. Future research should focus on developing frameworks that integrate ethical considerations into algorithm design, preventing misuse and ensuring fair treatment for all network participants.

Lastly, further exploration of the PageRank-inspired approach under different influence diffusion models and varying network structures could offer new insights. Investigating how adaptive strategies can be integrated into the PageRank framework may lead to more effective and dynamic influence maximization methods.

In summary, this thesis addresses the significant contributions to the field of influence maximization in social networks by emphasizing the protection of vulnerable users. Future research is necessary to advance the field, offering more robust, scalable, and ethically sound solutions for influence spread in an increasingly connected world.

BIBLIOGRAPHY

[1] Bai W., Iyer R., Wei K. and Bilmes J., *Algorithms for Optimizing the Ratio of Submodular Functions*, Proceedings of The 33rd International Conference on Machine Learning, PMLR, 2016.

[2] Brin S. and Page L., *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, CiteSeerX 10.1.1.115.5930, 1998.

[3] Buchbinder N., Feldman M., Naor J.S. and Schwartz, R., *Submodular maximization with cardinality constraints*, Society for Industrial and Applied Mathematics, 2014.

[4] Budak C., Agrawal D. and El Abbadi A., *Limiting the spread of misinformation in social networks*, Proceedings of the 20th International Conference on World Wide Web, Association for Computing Machinery, 2011.

[5] Chen H., Loukides G., Pissis S.P. and Chan H., *Influence maximization in the presence of vulnerable nodes: A ratio perspective*, Theoretical Computer Science, 852, 84-103. Advance online publication, 2021.

[6] Chen W., Wang Y. and Yang S., *Efficient influence maximization in social networks*, Knowledge Discovery and Data Mining, 2009.

[7] Chen W., Yuan Y. and Zhang L., *Scalable Influence Maximization in Social Networks under the Linear Threshold Model*, 2010 IEEE International Conference on Data Mining, 2010.

[8] Cheng S., Shen H., Huang J., Zhang G. and Cheng X., *StaticGreedy: Solving the Scalability-Accuracy Dilemma in Influence Maximization*, Association for Computing Machinery, New York, 2013.

[9] Domingos P, and Richardson M., *Mining the network value of customers*, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2001.

[10] Feige U., Mirrokni V. and Vondrák J., *Maximizing Non-Monotone Submodular Functions*, SIAM J. Comput., 2011.

[11] Goyal A., Lu W. and Lakshmanan L., *SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model*, 2011 IEEE 11th International Conference on Data Mining, 2011.

[12] Gupta S., *A Conceptual Framework That Identifies Antecedents and Consequences of Building Socially Responsible International Brands*, Thunderbird International Business Review, 2015.

[13] Gwadera R., Loukides G., *Cost-Effective Viral Marketing in the Latency Aware Independent Cascade Model*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2017.

[14] Iwata S., Fleischer L. and Fujishige S., *A Combinatorial, Strongly Polynomial-Time Algorithm for Minimizing Submodular Functions*, Association for Computing Machinery, New York, 2001.

[15] Iyer R. and Bilmes J., *Algorithms for Approximate Minimization of the Difference Between Submodular Functions, with Applications*, Uncertainty in Artificial Intelligence - Proceedings of the 28th Conference, UAI 2012.

[16] Iyer R., Jegelka S. and Bilmes J., *Curvature and Optimal Algorithms for Learning and Minimizing Submodular Functions*, Advances in Neural Information Processing Systems, 2013.

[17] Kempe D., Kleinberg J. and Tardos E., *Maximizing the Spread of Influence through a Social Network*, Association for Computing Machinery, 2003 .

[18] Krause A., Golovin D., *Submodular Function Maximization*, Cambridge University Press, 2014.

[19] Leskovec J., Krause A., Guestrin C., Faloutsos C., Vanbriesen J. and Glance N., *Cost-effective outbreak detection in networks*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.

[20] Li Y., Fan J., Wang Y. and Tan K., *Influence Maximization on Social Graphs: A Survey*, in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 10, 2018.

[21] Loukides G. and Gwadera R., *Preventing the diffusion of information to vulnerable users while preserving PageRank*, Springer International Publishing AG, 2017.

[22] Lu W., Chen W. and Lakshmanan L.V.S., *From competition to Complementarity: Comparative Influence Diffusion and Maximization*, VLDB Endowment, 2015.

[23] Minoux M., *Accelerated greedy algorithms for maximizing submodular set functions.*,Optimization Techniques, 1978

[24] Mirzasoleiman B., Badanidiyuru A., Karbasi A., Vondrák J. and Krause A., *Lazier Than Lazy Greedy*, Proceedings of the AAAI Conference on Artificial Intelligence, 2014.

[25] Mitrovic M., Bun M., Krause A. and Karbasi A., *Differentially Private Submodular Maximization: Data Summarization in Disguise*, In: Proceedings of the 34th International Conference on Machine Learning, PMLR, 2017.

[26] Narasimhan M. and Bilmes J., *A submodular-supermodular procedure with applications to discriminative structure learning*, In UAI, 2005.

[27] Nemhauser G., Wolsey L. and Fischer M.L., *An Analysis of Approximation For Maximizing Submodular Set Functions-I*, Mathematical Programming 14, North-Holland Publishing Company, 1978.

[28] Perrault P., Healey J., Wen Z. and Valko M., *On the Approximation Relationship between Optimizing Ratio of Submodular (RS) and Difference of Submodular (DS) Functions*, 2022.

[29] Richardson M. and Domingos P., *Mining knowledge-sharing sites for viral marketing*, Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2002.

[30] Steffes E. and Burgee L., *Social ties and online word of mouth*, Internet Research, 2009.

[31] Tang Y., Xiao X. and Shi Y., *Influence Maximization: Near-Optimal Time Complexity Meets Practical Efficiency*, Proceedings of the ACM SIGMOD International Conference on Management of Data, 2014.

[32] Varshney D., Kumar S. and Gupta V., *Predicting information diffusion probabilities in social networks: A Bayesian networks based approach*, Knowledge-Based Systems, 2017.

[33] Wolsey L. A., *An analysis of the greedy algorithm for the submodular set covering problem*, Combinatorica, 1982.

[34] Zimeras S., *Markov Random Field and Social Networks*, in:Virtual Communities, Social Networks and Collaboration, 2012.

APPENDIX A

EVALUATING THE SPREAD FUNCTION

In the context of influence maximization in social networks, it is essential to define the key concepts and models used. We start by defining a directed graph (digraph) and its components.

Definition A.1. A digraph G is an ordered pair $G = (U, E)$ where U is a set of *nodes* (vertices) and E is a set of *directed edges*. Each edge (u, v) in E represents a directed connection from node u to node v. Additionally, u is called an *in-neighbor* of v while v is an *out-neighbor* of u.

A social network can be represented as such a digraph $G = (U, E)$. Here, the nodes U represent individuals or entities, and the directed edges E represent the influence relationships between these entities. Specifically, an edge (u, v) indicates that node v follows or is influenced by node u. A stochastic model M defines the probabilities associated with the edges in the social network G [17, 22, 31]. Each directed edge (u, v) in E has an associated probability $p_{u,v}$, which represents the likelihood that node u will influence node v. The model M can be represented in different ways. One common representation of M is a probability matrix P, where $P[u, v] = p_{u,v}$. Each entry in this matrix corresponds to the influence probability of an edge in the network. Alternatively, M could be a probabilistic graphical model that captures more complex dependencies and conditional probabilities between nodes. Examples of such models include Bayesian networks or Markov random fields, which can model joint distributions over sets of nodes [32, 34].

Definition A.2. The *spread function* σ is a set function $\sigma : 2^U \to \mathbb{R}$ parameterized by the stochastic model M.

Evaluating the spread function σ typically involves approximations due to the complexity of exact computation. Several methods can be used, including Monte Carlo simulations, which repeatedly simulate the diffusion process to estimate the expected spread, dynamic programming techniques to approximate the spread, various sampling methods to estimate the spread, and heuristic methods for faster approximations [17, 6, 8]. Here, we focus on approximations performed via Monte Carlo methods. Specifically, we have two basic techniques:

- Simulations, where we directly simulate the random process of diffusion on a given seed set S. Let A_r be the set of nodes activated in the r-th round of the simulation algorithm, with $A_0 = S$. During the $(r+1)$-th round, a node $u \in A_r$ tries to activate all of its inactive neighbors with a certain probability, that is, neighbor $v \notin A_r$ gets activated with probability $p_{u,v}$. For all successful activations, the corresponding nodes are added to A_{r+1} and the process continues until no activation is possible. The algorithm then returns the number of activated nodes and is repeated for a sufficient number of times; $\sigma(S)$ is estimated as the average of all outcomes.

- Snapshots that we produce in advance. As we have already discussed (1.1), activation of a node v from a node u with probability $p_{u,v}$ can be viewed as the flipping of a biased coin. In this sense, if we flip all coins in advance to see which activations will actually occur, we can create a *snapshot* graph by deleting all edges that were not selected for propagation, i.e., that do not provide an activation. After having created a sufficient number of snapshots, we run on all of them some algorithm that returns the number of nodes reachable from S; $\sigma(S)$ is again estimated as the average of all outcomes.

Both of these Monte Carlo methods need a large "sufficient number" of repetitions when employed for large, complex networks and NaïveGreedy will have to run the selected technique $O(|U| \cdot k)$ times, rendering our greedy algorithm inefficient for real-life problems [8]. We will take a look at a couple of the most basic algorithms proposed to ameliorate time complexity of the naïve approach by reducing the number of estimations of the spread functions. CELF and StaticGreedy preserve NaïveGreedy's approximation factor of $1 - e^{-1}$, while significantly reducing the Monte Carlo simulations needed.

First, we present the Cost-Effective Lazy Forward selection (CELF) from [19].

Algorithm CELF

```
Require: $U = \{u_1, \dots, u_n\}$, $\sigma : 2^U \to \mathbb{R}$, integer $k < |U|$, $R$ .
 1: $S^0 \leftarrow \emptyset, U^0 \leftarrow U, t \leftarrow 1$
 2: for $u \in U$ do
 3:     $d_u \leftarrow +\infty$
 4: while $t \leq k$ do
 5:     for $u \in U^{t-1}$ do
 6:         $cur_u \leftarrow$false
 7:     while true do
 8:         Select $u_t \in U^{t-1}$ s.t. $d_{u_t} = \max_{u \in U^{t-1}} \{d_u\}$.
 9:         if $cur_{u_t}$ then
10:             $S^t \leftarrow S^{t-1} \cup \{u_t\}, U^t \leftarrow U^{t-1} \setminus \{u_t\}, t \leftarrow t+1$
11:             break
12:         else
13:             $s_{u_t} \leftarrow 0$
14:             for $j = 1, \dots, R$ do
15:                 $s_{u_t} += \sigma(S^{t-1} \mid u_t)$
16:             $d_{u_t} \leftarrow s_{u_t}/R$
17:             $cur_{u_t} \leftarrow$true
18: return $S^G \leftarrow S^k$
```

The logic behind CELF is that the marginal gain of node v in the current iteration cannot exceed its marginal gain in the previous iterations. That is, suppose that S is the current seed set, and $d_v = \sigma(S \mid v)$ is the marginal gain of v. Since σ is submodular, we know that for any other $S' \supset S$, i.e., for any other seed set from a subsequent iteration, $\sigma(S \mid v) \geq \sigma(S' \mid v)$. Therefore, after having estimated $\sigma(\emptyset \mid v) = \sigma(v)$ and added the node with maximum gain (first execution of inner **while**), the algorithm at each iteration re-estimates $\sigma(S \mid v)$ for nodes $v \in U \setminus S$ that had the highest d_v in the previous iteration, until it finds the one that stays on top and adds it to the seed set. The Monte Carlo simulations are taking place in lines 14 and 15.

Even though CELF does not improve the worst-case time complexity of NaïveGreedy, in practice it can improve efficiency to 700 times [19].

Next, we look over the StaticGreedy algorithm proposed in [8].

Algorithm StaticGreedy

```
Require: $U = \{u_1, \ldots, u_n\}$, $\sigma : 2^U \to \mathbb{R}$, integer $k < |U|$, $R$.
1:  $S^0 \leftarrow \emptyset, U^0 \leftarrow U, t \leftarrow 1$
2:  for $r = 1, \ldots, R$ do
3:      Generate snapshot $G'_r$ by removing each edge $(u, v)$ with probability $1 - p_{u,v}$
4:  while $t \leq k$ do
5:      for $u \in U^{t-1}$ do
6:          $s_u \leftarrow 0$
7:      for $j = 1, \ldots, R$ do
8:          for $u \in U^{t-1}$ do
9:              $s_u += \sigma_j(S^{t-1} \mid u)$
10:     Select $u_t \in U^{t-1}$ s.t. $s_{u_t}/R = \max_{u \in U^{t-1}} \{s_u/R\}$.
11:     $S^t \leftarrow S^{t-1} \cup \{u_t\}, U^t \leftarrow U^{t-1} \setminus \{u_t\}$
12:     $t \leftarrow t + 1$
13: return $S^G \leftarrow S^k$
```

StaticGreedy generates a sufficient (not very large) number R of Monte Carlo snapshots $G'_1, \ldots, G'_R$ at the very beginning, instead of a large number of Monte Carlo simulations in every iteration. Then it uses this same set of *static* snapshots in all iterations, performing R evaluations for each node's marginal gain w.r.t. each static snapshot and greedily selecting the node with the maximum average. The Monte Carlo snapshot method is described in lines 7 - 9; in line 9, σ_j denotes the restriction of the spread function on $G'_j, \forall j \in [R]$. StaticGreedy can accelerate NaïveGreedy up to 100 times [8].

We can easily see that these algorithms can be used to improve the greedy selection process of every algorithm involving evaluations of σ, so their contribution becomes very valuable in a general context. They significantly reduce the computational overhead, making them very practical for real-world applications.

APPENDIX B SUBMODULAR FUNCTIONS

Let $U = \{u_1, \dots, u_n\}$ be a finite set with n elements and $f : 2^U \to \mathbb{R}$ be a real function on 2^U, the power set of U. Such a function will be called a *set function*. We simply write $f(u)$ instead of $f(\{u\})$.

A set function $f : 2^U \to \mathbb{R}$ is said to be *submodular* if and only if, for every $S \subseteq U, T \subseteq U$:

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T).$$

Another more intuitive characterization of submodular set functions in terms of the *marginal gains* :

$$f(S \mid u_i) = f(S \cup \{u_i\}) - f(S)$$

is given by the next theorem:

Theorem B.1 ([27]). The following properties are equivalent:

- function $f : 2^U \to \mathbb{R}$ is submodular
- f satisfies the *diminishing returns* property, i.e., for every S, T such that $S \subseteq T \subseteq U$ and for every $u_i \notin T$, $f(T \mid u_i) \leq f(S \mid u_i)$.

Proof. Consider $S \subseteq T \subseteq U$ and $u_j \notin T$. Since f is submodular, it holds that $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for any two sets $S, T \subseteq U$. We set $S := S \cup \{u_j\}, T := T$. Then

$$\begin{aligned}
&f(S \cup \{u_j\}) + f(T) \geq f((S \cup \{u_j\}) \cup T) + f((S \cup \{u_j\}) \cap T) \Rightarrow \\
&f(S \cup \{u_j\}) + f(T) \geq f(T \cup \{u_j\}) + f(S) \Rightarrow \\
&f(S \cup \{u_j\}) - f(S) \geq f(T \cup \{u_j\}) - f(T)
\end{aligned}$$

For the other direction, let $T \setminus S = \{u_{i_1}, \dots, u_{i_k}\}$. Since $T \cap S \subseteq S$ and therefore $(T \cap S) \cup \{u_{i_1}, \dots, u_{i_{j-1}}\} \subseteq S \cup \{u_{i_1}, \dots, u_{i_{j-1}}\}$ for all $j = 1 \dots, k$ (where for $k = 1$ we consider $\{u_{i_0}\} = \emptyset$), we have that

$$f(S \cup \{u_{i_1}, \dots, u_{i_{j-1}}\} \mid u_j) \leq f((T \cap S) \cup \{u_{i_1}, \dots, u_{i_{j-1}}\} \mid u_j) \;\; \forall j = 1, \dots, k \Rightarrow$$

$$\begin{aligned}
f(S \cup \{u_{i_1}, \dots, u_{i_j}\}) - f(S \cup \{u_{i_1}, \dots, u_{i_{j-1}}\}) \leq f(T \cap S \cup \{u_{i_1}, \dots, u_{i_j}\}) \\
-f(T \cap S \cup \{u_{i_1}, \dots, u_{i_{j-1}}\}) \;\; \forall j = 1, \dots, k.
\end{aligned}$$

If we add these k inequalities we get $f(T \cup S) - f(S) \leq f(T) - f(T \cap S)$. □

If the inverse inequality of the diminishing returns property holds, then f is said to be *supermodular*, otherwise if the equality holds then the function is *modular*.

Next, we present some other useful equivalent definitions of submodular functions.

Proposition B.2 ([27])**.** The following statements are equivalent:

(i) $f : 2^U \to \mathbb{R}$ is submodular.

(ii) $f(T) \le f(S) + \sum_{u \in T \setminus S} f(S \mid u) - \sum_{u \in S \setminus T} f(S \cup T \setminus \{u\} \mid u)$

Proof. Let $S, T \subseteq U$ such that $T \setminus S = \{u_1, \dots, u_n\}$ and $S \setminus T = \{v_1, \dots, v_m\}$. Then

$$\begin{aligned} f(S \cup T) - f(S) &= \sum_{i=1}^{n} f(S \cup \{u_1, \dots, u_i\}) - f(S \cup \{u_1, \dots, u_{i-1}\}) \\ &= \sum_{i=1}^{n} f(S \cup \{u_1, \dots, u_{i-1}\} \mid u_i) \\ &\le \sum_{i=1}^{n} f(S \mid u_i) = \sum_{u \in T \setminus S} f(S \mid u). \end{aligned}$$

Similarly,

$$\begin{aligned} f(S \cup T) - f(T) &= \sum_{i=1}^{m} f(T \cup \{v_1, \dots, v_i\}) - f(T \cup \{v_1, \dots, v_{i-1}\}) \\ &= \sum_{i=1}^{m} f(T \cup \{v_1, \dots, v_i\} \setminus \{v_i\} \mid v_i) \\ &\ge \sum_{i=1}^{m} f(T \cup S \setminus \{v_i\} \mid v_i) = \sum_{u \in S \setminus T} f(S \cup T \setminus \{u\} \mid u). \end{aligned}$$

□

For some real parameter $\theta \ge 0$, let $C(\theta)$ denote the class of submodular functions satisfying the property $f(A \mid u) \ge -\theta$ for all $A \subset U$ and $u \in U \setminus A$.

Proposition B.3 ([27])**.** Let $f : 2^U \to \mathbb{R}$ be a submodular function in $C(\theta)$. Then

$$f(T) \le f(S) + \sum_{u \in T \setminus S} f(S \mid u) + |S \setminus T|\theta, \forall S, T \subseteq U.$$

Proof. Immediate result from the previous proposition. □

The following proposition states that submodularity is preserved under non-negative linear combinations.

Proposition B.4. Let $f_1, \dots, f_n : 2^U \to \mathbb{R}$ be submodular functions and $\alpha_1, \dots, \alpha_n \ge 0$. Then $f = \sum_{i=1}^{n} \alpha_i f_i$ is also submodular.

Proof. Let $S, T \subseteq U$ such that $S \subseteq T$ and $u \in U \setminus T$. Then. since $f_1, \dots, f_n$ are submodular, we have that

$$f_i(S \mid u) \le f_i(T \mid u), \ \forall i \in [n]$$

and, since $\alpha_i \geq 0, \forall i \in [n]$, it holds that

$$\alpha_i f_i(S \mid u) \leq \alpha_i f_i(T \mid u), \forall i \in [n].$$

Therefore,

$$\begin{aligned} f(S \mid u) &= \alpha_1 f_1(S \mid u) + \dots \alpha_n f_n(S \mid u) \\ &\geq \alpha_1 f_1(T \mid u) + \dots \alpha_n f_n(T \mid u) \\ &= f(T \mid u), \end{aligned}$$

which means that f is also submodular. $\square$

Next, we define [1, 15] the modular lower bound of a submodular function $f : 2^U \to \mathbb{R}$, using the subdifferential $\partial_f(Y), Y \subseteq U$:

$$\partial_f(Y) = \{y \in \mathbb{R}^U \mid f(X) - y(X) \geq f(Y) - y(Y), \forall X \subseteq U\}, Y \subseteq U$$

where $y(X) = \sum_{j \in X} y(j)$. The extreme points of $\partial_f(Y)$ can be easily obtained using a greedy algorithm. Let σ be a permutation of U, with $|U| = n$, such that σ assigns the elements of Y in the first $|Y|$ positions. That is, σ defines a chain with elements

$$S_0^\sigma = \emptyset, S_i^\sigma = \{\sigma(1), \dots, \sigma(i)\}, \forall i \in [n],$$

and $Y = S_{|Y|}^\sigma$. We say that σ's chain *contains* Y. Then this chain defines an extreme point $h_f^{Y,\sigma}$, i.e., a subgradient corresponding to f, as follows:

$$h_{\sigma,Y}^f(\sigma(1)) = f(S_1^\sigma), h_{\sigma,Y}^f(\sigma(i)) = f(S_i^\sigma) - f(S_{i-1}^\sigma), \forall i > 1$$

and

$$h_{\sigma,Y}^f(S) = \sum_{u \in S} h_{\sigma,Y}^f(S).$$

Observe that $h_{\sigma,Y}^f(S_i^\sigma) = f(S_i^\sigma)$. With a slight change in notation, $h_{\sigma,Y}^f$ represents a modular lower bound of f with parameter Y (tight at Y) which we will denote by

$$\widetilde{f_{Y,\sigma^Y}}(X) = \sum_{u \in X} f_{Y,\sigma^Y}(u),$$

where σ^Y is a random permutation of the elements of Y and

$$f_{Y,\sigma^Y}(u) = \begin{cases} f(\sigma_u^Y) - f(\sigma_{u^-}^Y), & \text{if } u \in Y \\ 0, & \text{otherwise} \end{cases},$$

where $\sigma_{u^-}^Y$ denotes the prefix of all elements in σ^Y that come before u and σ_u^Y is $\sigma_{u^-}^Y$ along with u. Here, instead of a permutation of U whose chain contains Y, we have considered a permutation only on Y and have set the values of all other elements to 0. As a result, we have $\widetilde{f_{Y,\sigma^Y}}(X) = f(Y)$, for every $X \supseteq Y$.

We proceed with the definition of the modular upper bound of f, similarly as before but with the use of the superdifferential of f:

$$\partial^f(Y) = \{y \in \mathbb{R}^U \mid f(X) - y(X) \leq f(Y) - y(Y), \forall X \subseteq U\}, Y \subseteq U.$$

Computing the extreme points of $\partial^f(Y)$ yields two supergradients:

$$m^f_{X,1}(Y) = f(X) + \sum_{u \in Y \setminus X} f(u) - f(\emptyset)) - \sum_{u \in X \setminus Y} (f(X) - f(X \setminus \{u\}))$$

and

$$m^f_{X,2}(Y) = f(X) + \sum_{u \in Y \setminus X} f(X \cup \{u\}) - f(X)) - \sum_{u \in X \setminus Y} (f(U) - f(U \setminus \{u\})).$$

These represent modular upper bounds of f with parameter X (tight at X). We denote them by $\widehat{f_{X,1}}(Y)$ and $\widehat{f_{X,2}}(Y)$, respectively. Obviously, $\widehat{f_{X,1}}(X) = \widehat{f_{X,2}}(X) = f(X)$. It is also easy to see that

$$\widehat{f_{X,1}}(X \setminus \{v\}) = f(X \setminus \{v\}) \quad \text{and} \quad \widehat{f_{X,2}}(X \cup \{v\}) = f(X \cup \{v\}).$$

We now define the submodular and supermodular curvature [1, 16].

Definition B.5. Let $f : 2^U \to \mathbb{R}_{\geq 0}$ be a non-negative set function. The *submodular curvature* of f is defined as

$$\kappa_f = 1 - \min_{u \in U} \frac{f(U) - f(U \setminus \{u\})}{f(u)}.$$

The submodular curvature κ_f indicates how close to modular f is, with $\kappa_f = 0$ meaning that f is modular and $\kappa_f = 1$ meaning f is fully curved.

We also define the submodular curvature of f with respect to $X \subseteq U$ as

$$\kappa_f(X) = 1 - \frac{\sum_{u \in X}(f(X) - f(X \setminus \{u\}))}{\sum_{u \in X} f(u)}.$$

Note that $f(X) - f(X \setminus \{u\})$ represents the smallest contribution of each element u of X to the total contribution of X, while $f(u)$ represents the largest such contribution, i.e.,

$$f(X) \geq \sum_{u \in X}(f(X) - f(X \setminus \{u\})) \text{ and } f(X) \leq \sum_{u \in X} f(u).$$

We refer to the quantity $\sum_{u \in X}(f(X) - f(X \setminus \{u\}))$ as the *simple lower bound* of f and to $\sum_{u \in X} f(u)$ as the *simple upper bound* of f.

Similarly, we have the supermodular curvatures:

Definition B.6. Let $f : 2^U \to \mathbb{R}_{\geq 0}$ be a non-negative set function. The *supermodular curvature* of f is defined as

$$\kappa^f = 1 - \min_{u \in U} \frac{f(u)}{f(U) - f(U \setminus \{u\})}$$

and the supermodular curvature of f with respect to $X \subseteq U$ as

$$\kappa^f(X) = 1 - \frac{\sum_{u \in X} f(u)}{\sum_{u \in X}(f(X) - f(X \setminus \{u\}))}.$$

Note that $\kappa^f = 0$ also means that f is modular, while if f is submodular we have $\kappa^f < 0$ or κ^f is undefined (if no u exists such that $f(U) - f(U \setminus \{u\}) > 0$).

Proposition B.7. Let $f : 2^U \to \mathbb{R}_{\geq 0}$ be a non-negative submodular function. It holds that $\kappa_f \geq \kappa_f(X), \forall X \subseteq U$ and $\kappa^f \leq \kappa^f(X), \forall X \subseteq U$.

Proof. We define the auxiliary function $k_f : 2^U \to \mathbb{R}$,

$$k_f(X) = 1 - \min_{u \in X} \frac{f(X) - f(X \setminus \{u\})}{f(u)}$$

and consider $S \subseteq T \subseteq U$. For any $u \in S$ it holds that

$$\frac{f(S \setminus \{u\} \mid u)}{f(u)} \geq \frac{f(T \setminus \{u\} \mid u)}{f(u)}$$

due to the submodularity of f. If $\min\limits_{u \in S} \frac{f(S \setminus \{u\} | u)}{f(u)} = \frac{f(S \setminus \{v\} | v)}{f(v)}$ for some $v \in S$, then

$$\min_{u \in S} \frac{f(S \setminus \{u\} \mid u)}{f(u)} \geq \frac{f(T \setminus \{v\} \mid v)}{f(v)} \geq \min_{u \in T} \frac{f(T \setminus \{u\} \mid u)}{f(u)} \Rightarrow k_f(S) \leq k_f(T).$$

This means that k_f is monotone increasing and $k_f(X) \leq k_f(U) = \kappa_f, \forall X \subseteq U$. On the other hand,

$$1 - k_f(X) = \min_{u \in X} \frac{f(X \setminus \{u\} \mid u)}{f(u)} \leq \frac{f(X \setminus \{u\} \mid u)}{f(u)}, \forall u \in X$$

and

$$\begin{aligned} 1 - \kappa_f(X) &= \frac{\sum_{u \in X} f(X \setminus \{u\} \mid u)}{\sum_{u \in X} f(u)} \\ &\geq \frac{\sum_{u \in X} (1 - k_f(X)) f(u)}{\sum_{u \in X} f(u)} \\ &= 1 - k_f(X), \end{aligned}$$

i.e., $k_f(X) \geq \kappa_f(X)$. Finally we get

$$\kappa_f(X) \leq \kappa_f, \forall X \subseteq U.$$

With a very similar argument we can also prove that

$$\kappa^f(X) \geq \kappa^f, \forall X \subseteq U.$$

□

We close this appendix with two probabilistic properties of submodular functions.

Theorem B.8 ([10]). Let $f : 2^U \to \mathbb{R}$ be submodular. Denote by $A(p)$ a random subset of A where each element appears with probability p, not necessarily independently. Then

$$\mathbb{E}[f(A(p))] \geq (1 - p) \cdot f(\emptyset) + p \cdot f(A).$$

Proof. We will use induction on the size of A:

- For $A = \emptyset$, we have $\mathbb{E}[f(\emptyset)] = f(\emptyset)$
- Assume $A = A' \cup \{u\}$ and that the property holds for A', i.e.,
$$\mathbb{E}[f(A'(p))] \geq (1-p)\cdot f(\emptyset) + p \cdot f(A').$$
- We observe that $A'(p) = A(p) \cap A'$, so
$$\begin{aligned}\mathbb{E}[f(A(p))] &= \mathbb{E}[f(A'(p))] + \mathbb{E}[f(A(p)) - f(A'(p))] \\ &= \mathbb{E}[f(A'(p))] + \mathbb{E}[f(A(p)) - f(A(p) \cap A')] \\ &\geq \mathbb{E}[f(A'(p))] + \mathbb{E}[f(A' \cup A(p)) - f(A')]. && \text{(by submodularity)}\end{aligned}$$
We also observe that, when $u \in A(p)$, which occurs with probability p, then $A' \cup A(p) = A$, otherwise $A' \cup A(p) = A'$. Thus,
$$\begin{aligned}\mathbb{E}[f(A(p))] &\geq \mathbb{E}[f(A'(p))] + p\cdot(f(A) - f(A')) \\ &\geq (1-p)\cdot f(\emptyset) + p\cdot f(A') + p\cdot(f(A) - f(A')) && \text{(by the inductive hypothesis)} \\ &= (1-p)\cdot f(\emptyset) + p\cdot f(A).\end{aligned}$$

□

Theorem B.9 ([3]). Let $f : 2^U \to \mathbb{R}$ be submodular. Denote by $A(p)$ a random subset of A where each element appears with probability at most p, not necessarily independently. Then
$$\mathbb{E}[f(A(p))] \geq (1-p)\cdot f(\emptyset).$$

Proof. We sort $A = \{u_1, u_2 \ldots, u_n\}$ in a decreasing order of probability, i.e., if E_i is and indicator for the event $u_i \in A(p)$, and $p_i = \Pr[u_i \in A(p)] = \mathbb{E}[E_i]$, then $p_1 \geq p_2 \geq \ldots \geq p_n$. Let $A_i = \{u_1, u_2, \ldots, u_i\}$. Then
$$\begin{aligned}\mathbb{E}[f(A(p))] &= \mathbb{E}\left[f(\emptyset) + \sum_{i=1}^{n} E_i \cdot f(A_{i-1} \cap A(p) \mid u_i)\right] \\ &\geq \mathbb{E}\left[f(\emptyset) + \sum_{i=1}^{n} E_i \cdot f(A_{i-1} \mid u_i)\right] && \text{(by submodularity)} \\ &= f(\emptyset) + \sum_{i=1}^{n} \mathbb{E}[E_i]\cdot f(A_{i-1} \mid u_i) \\ &= f(\emptyset) + \sum_{i=1}^{n} p_i \cdot (f(A_i) - f(A_{i-1})) \\ &= (1-p_1)\cdot f(\emptyset) + \sum_{i=1}^{n-1}(p_i - p_{i+1})\cdot f(A_i) + p_n \cdot f(A) \\ &\geq (1-p)\cdot f(\emptyset) && (p_i - p_{i+1} \geq 0 \text{ and } p_1 \leq p)\end{aligned}$$

□

APPENDIX C

PROOFS OF CHAPTERS 2&3

C.1 Chapter 2

Naïve Greedy

Proof of Proposition 2.1. From Proposition B.3, we have that

$$\sigma(T) \leq \sigma(S) + \sum_{u \in T \setminus S} \sigma(S \mid u) + |S \setminus T|\theta, \forall S, T \subseteq U.$$

Let $\ell < k$ be the number of iterations of the algorithm. Then

$$\sigma(S^\ell \mid u) \leq d_\ell \leq 0, |S^* \setminus S^\ell| \leq k \Rightarrow \sum_{u \in S^* \setminus S^\ell} \sigma(S^\ell \mid u) \leq k d_\ell \leq 0.$$

By setting $T := S^*, S = S^\ell$ and since $|S^\ell \setminus S^*| \leq \ell$ we get

$$\begin{aligned}
\sigma(S^*) &\leq \sigma(S^\ell) + \sum_{u \in S^* \setminus S^\ell} \sigma(S^\ell \mid u) + |S^\ell \setminus S^*|\theta \\
&\leq \sigma(\emptyset) + \sum_{i=1}^{\ell} d_{i-1} + \ell\theta \Rightarrow \\
\sigma(S^*) &\leq \sigma(S^G) + \ell\theta.
\end{aligned}$$

For $\theta = 0$ we get $\sigma(S^*) \leq \sigma(S^G)$, which means that the greedy solution is optimal. $\square$

Proof of Theorem 2.2. Let $S^* \in \arg\max\{\sigma(S) : |S| \leq k\}$ be an optimal solution of size k and enumerate $S^* = \{u_1^*, \dots, u_k^*\}$. Then for $0 \leq t < \ell$ we have

$$\begin{aligned}
\sigma(S^*) &\leq \sigma(S^* \cup S^t) && (\sigma \text{ is monotone}) \\
&= \sigma(S^t) + \sum_{i=1}^{k} \sigma(S^t \cup \{u_1^*, \dots, u_{i-1}^*\} \mid u_i^*) && (\text{telescoping sum}) \\
&\leq \sigma(S^t) + \sum_{u \in S^*} \sigma(S^t \mid u) && (\sigma \text{ is submodular}) \\
&\leq \sigma(S^t) + \sum_{u \in S^*} \sigma(S^{t+1}) - \sigma(S^t) && (\text{greedy selection}) \\
&\leq \sigma(S^t) + k(\sigma(S^{t+1}) - \sigma(S^t)) && (|S^*| \leq k).
\end{aligned}$$

We set $\sigma_t := \sigma(S^*) - \sigma(S^t)$, so the previous inequality is written as

$$\sigma_t \leq k(\sigma_t - \sigma_{t+1}) \Rightarrow \sigma_{t+1} \leq \left(1 - \frac{1}{k}\right)\sigma_t \Rightarrow$$

$$\sigma_\ell \leq \left(1 - \frac{1}{k}\right)^\ell \sigma_0 \leq \left(1 - \frac{1}{k}\right)^\ell \sigma(S^*) \leq e^{-\ell/k}\sigma(S^*) \Rightarrow$$

$$\sigma(S^*) - \sigma(S^t) \leq e^{-\ell/k}\sigma(S^*) \Rightarrow \sigma(S^t) \geq (1 - e^{-\ell/k})\sigma(S^*).$$

□

Subsample Greedy

We will show a lemma that is going to be useful for the proof of Theorem 2.5. When referring to the ground set U, we will mean the set we obtain after Phase I completes, i.e., $|U| = m = Nk$ for some $N \in \mathbb{N}$.

Lemma C.1 ([25])**.** It holds that

$$\mathbb{E}[\sigma(S^{t-1} \mid u_t)] \geq \left(1 - \frac{1}{e}\right) \cdot \left(\frac{\mathbb{E}[\sigma(S^*)] - \mathbb{E}[\sigma(S^{t-1})]}{k}\right) - \alpha$$

for some parameter $\alpha > 0$.

Proof. Fix t and let $M \subseteq U \cup D$ be the set that maximizes $\sum_{v \in M} \sigma(S^{t-1} \mid v)$. Also denote by A the event $R^t \cup \{v_t\} \cap M \neq \emptyset$.

Now, if we sort $U \cup D = \{v^{(1)}, \dots, v^{(m)}, v^{(m+1)}, \dots, v^{(m+k)}\}$ with the criterion $\sigma(S^{t-1} \mid v^{(i)}) \geq \sigma(S^{t-1} \mid v^{(i+1)}), \forall i \in [m+k-1]$, then $M = \{v^{(1)}, \dots, v^{(k)}\}$, since M consists of the k elements that give the largest marginal gain to S^{t-1}. In addition, consider that dummy elements come after the actual ones in M, and let μ be the index such that $v^{(\mu)} \in U$ and $v^{(\mu+1)} \in D$.

Let A_i be the event that i is the smallest index such that $v^{(i)} \in R^t \cup \{v_t\}$, then $A_1, \dots, A_{m+k}$ are mutually exclusive and $A = \bigcup_{i=1}^k A_i$, which means that $\Pr[A] = \sum_{i=1}^k \Pr[A_i]$. We observe that

$$\Pr[A_1] = \frac{1}{k}, \quad \Pr[A_i] = \frac{\binom{m-i}{m/k-1}}{\binom{m}{m/k}}, i = 2, \dots, \mu,$$

$$\Pr[A_i] = \frac{\binom{m-\mu}{m/k}}{\binom{m}{m/k}} \cdot \frac{1}{k} \cdot \left(1 - \frac{1}{k}\right)^{i-\mu-1}, i = \mu+1, \ldots, k$$

and that $\Pr[A_i]$ is a decreasing function, and so is $\Pr[A_i \mid A]$. Therefore,

$$\begin{aligned}
\mathbb{E}[\sigma(S^{t-1} \mid u_t) \mid A] &= \sum_{i=1}^{k} \mathbb{E}[\sigma(S^{t-1} \mid u_t) \mid A_i] \cdot \Pr[A_i \mid A] \\
&\geq \frac{1}{k} \sum_{i=1}^{k} \left(\sigma(S^{t-1} \mid v^{(i)}) - \alpha\right) \quad \text{(Chebyshev's sum inequality)} \\
&= \frac{1}{k} \sum_{v \in M} \sigma(S^{t-1} \mid v) - \alpha
\end{aligned}$$

Moreover,

$$\begin{aligned}
\Pr[A] &= 1 - \Pr[M \cap (R^t \cup \{v_t\}) = \emptyset] = 1 - \frac{\binom{m-\mu}{m/k}}{\binom{m}{m/k}} \frac{\mu}{k} \\
&= \ldots \\
&\geq 1 - \left(1 - \frac{1}{k}\right)^{\mu} \frac{\mu}{k} \geq 1 - \frac{\mu e^{-\mu/k}}{k} \\
&\geq 1 - \frac{1}{e}.
\end{aligned}$$

Finally, let M' be a set such that $S^* \setminus S^{t-1} \subseteq M'$ and M' is completed with dummy elements so that $|M'| = k$. Then,

$$\begin{aligned}
\mathbb{E}[\sigma(S^{t-1} \mid u_t)] &= \mathbb{E}[\sigma(S^{t-1} \mid u_t) \mid A]\Pr[A] + \mathbb{E}[\sigma(S^{t-1} \mid u_t) \mid \overline{A}](1 - \Pr[A]) \\
&\geq \left(1 - \frac{1}{e}\right)\left(\frac{1}{k} \sum_{v \in M} \sigma(S^{t-1} \mid v) - \alpha\right) - \frac{1}{e}\alpha \\
&\geq \left(1 - \frac{1}{e}\right)\left(\frac{1}{k} \sum_{v \in M'} \sigma(S^{t-1} \mid v)\right) - \alpha \quad \text{(by definition of } M'\text{)} \\
&\geq \left(1 - \frac{1}{e}\right) \frac{\sigma(S^* \cup S^{t-1}) - \sigma(S^{t-1})}{k} - \alpha \quad \text{(by submodularity)} \\
&\geq \left(1 - \frac{1}{e}\right) \frac{\sigma(S^*) - \sigma(S^{t-1})}{k} - \alpha. \quad \text{(by monotonicity)}
\end{aligned}$$

Taking the expectation over t concludes our proof. □

Proof of Theorem 2.5. From the previous lemma we have

$$\mathbb{E}[\sigma(S^{t-1} \mid u_t)] \geq \left(1 - \frac{1}{e}\right) \frac{\sigma(S^*) - \sigma(S^{t-1})}{k} - \alpha,$$

which yields

$$\begin{aligned}\sigma(S^*) - \mathbb{E}[\sigma(S^t)] &\leq \left(1 - \frac{1-1/e}{k}\right) \cdot (\sigma(S^*) - \mathbb{E}[\sigma(S^{t-1})]) + \alpha \\ &\leq \left(1 - \frac{1-1/e}{k}\right)^t \cdot (\sigma(S^*) - \mathbb{E}[\sigma(S^0)]) + \sum_{i=0}^{t-1} \left(1 - \frac{1}{k}\right)^i \alpha \\ &\leq \left(1 - \frac{1-1/e}{k}\right)^t \sigma(S^*) + ta\end{aligned}$$

and for $t = k$ we obtain

$$\begin{aligned}\mathbb{E}[\sigma(S^G)] &\geq \left[1 - \left(1 - \frac{1-1/e}{k}\right)^k\right] \sigma(S^*) - k\alpha \\ &\geq \left(1 - e^{-(1-1/e)}\right) \sigma(S^*) - k\alpha\end{aligned}$$

Next, we should mention that parameter α represents a differential-privacy related term that can be omitted in our case (see [25] for more details), giving us the desired approximation guarantee

$$\mathbb{E}[\sigma(S^G)] \geq \left(1 - e^{-(1-1/e)}\right) \sigma(S^*).$$

□

Sandwich Approximation

Proof of Theorem 2.6. We denote by $S^{L,*}, S^{U,*}$ the optimal solutions w.r.t. σ^L, σ^U, respectively, and S^L is the greedily selected set maximizing σ^L. Since σ^L, σ^U are submodular, it holds (from NAÏVEGREEDY) that

$$\sigma^L(S^L) \geq \left(1 - \frac{1}{e}\right) \sigma^L(S^{L,*}), \quad \sigma^U(S^U) \geq \left(1 - \frac{1}{e}\right) \sigma^U(S^{U,*}).$$

Thus

$$\begin{aligned}\sigma(S^G) \geq \sigma(S^L) \geq \sigma^L(S^L) & & \text{(by definitions of } S^G, \text{ lower bound)} \\ \geq \left(1 - \frac{1}{e}\right) \sigma^L(S^{L,*}) & & (\sigma^L \text{ is submodular}) \\ \geq \left(1 - \frac{1}{e}\right) \sigma^L(S^*) & & \text{(by definition of } S^{L,*}) \\ = \frac{\sigma^L(S^*)}{\sigma(S^*)} \left(1 - \frac{1}{e}\right) \sigma(S^*) & &\end{aligned}$$

and

$$\begin{aligned}\sigma(S^G) \geq \sigma(S^U) = \frac{\sigma(S^U)}{\sigma^U(S^U)} \cdot \sigma^U(S^U) & & \text{(by definition of } S^G) \\ \geq \frac{\sigma(S^U)}{\sigma^U(S^U)} \cdot \left(1 - \frac{1}{e}\right) \sigma^U(S^{U,*}) & & (\sigma^U \text{ is submodular}) \\ \geq \frac{\sigma(S^U)}{\sigma^U(S^U)} \cdot \left(1 - \frac{1}{e}\right) \sigma^U(S^*) & & \text{(by definition of } S^{U,*}) \\ \geq \frac{\sigma(S^U)}{\sigma^U(S^U)} \cdot \left(1 - \frac{1}{e}\right) \sigma(S^*) & & \text{(by definition of upper bound)}\end{aligned}$$

By combining the above results we get the required inequality. □

C.2 Chapter 3

Proof of Lemma 3.4. If f is submodular the proof is trivial. We suppose that f is not submodular and we define $\alpha_f = \min_{\substack{S \subset T \subseteq U \\ u \notin T}} \{f(S \mid u) - f(T \mid v)\} < 0$. Let ρ be a submodular set function with $\alpha_\rho = \min_{\substack{S \subset T \subseteq U \\ u \notin T}} \{\rho(S \mid u) - \rho(T \mid v)\} > 0$ and set $\sigma = f + \frac{|\alpha|}{\alpha_\rho}\rho$, for some $\alpha \in \mathbb{R}$. If we choose $\alpha \leq \alpha_f < 0$, we get

$$\begin{aligned}
\alpha_\sigma &= \min_{\substack{S \subset T \subseteq U \\ u \notin T}} \{\sigma(S \mid u) - \sigma(T \mid v)\} \\
&= \min_{\substack{S \subset T \subseteq U \\ u \notin T}} \{f(S \mid u) - f(T \mid v)\} + \frac{|\alpha|}{\alpha_\rho} \min_{\substack{S \subset T \subseteq U \\ u \notin T}} \{\rho(S \mid u) - \rho(T \mid v)\} \\
&= \alpha_f + |\alpha| \geq 0,
\end{aligned}$$

which means that σ is submodular and therefore $f = \sigma - \frac{|\alpha|}{\alpha_\rho}\rho$ can be expressed as the difference of two submodular functions. □

Proof of Theorem 3.14. Let $S^L \subseteq N$. Since $\text{ASR}(S^L, c) \geq \text{ASR}^L(S^L, c)$ and the expected value is monotone, we have that

$$\mathbb{E}[\text{ASR}(S^L, c)] \geq \mathbb{E}[\text{ASR}^L(S^L, c)].$$

Now, let $S^{L,*}$ be an optimal solution of size at most k w.r.t. ASR^L. We observe that S^L is constructed by the SubsampleGreedy algorithm (see 2.2) and ASR^L is monotone submodular, therefore

$$\mathbb{E}[\text{ASR}^L(S^L, c)] \geq \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^L(S^{L,*}, c).$$

Hence,

$$\begin{aligned}
\mathbb{E}[\text{ASR}(S^L, c)] &\geq \mathbb{E}[\text{ASR}^L(S^L, c)] \\
&\geq \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^L(S^{L,*}, c) \\
&\geq \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^L(S^*, c) \\
&\geq \frac{\text{ASR}^L(S^*, c)}{\text{ASR}(S^*, c)} \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c) \\
&\geq \frac{\sigma_V(S^*) + c}{|V| + c} \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c) \\
&\geq \frac{c}{|V| + c} \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c)
\end{aligned}$$

In addition, since we also have

$$\mathbb{E}[\text{ASR}^U(S^U, c)] \geq \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^U(S^{U,*}, c)$$

for any solution $S^{U,*}$ optimal w.r.t. ASR^U (from 2.5), it holds that

$$\begin{aligned}
\text{ASR}(S^U, c) = \frac{\sigma_N(S^U) + c}{\sigma_V(S^U) + c} &= \frac{\text{ASR}^U(S^U) \cdot c}{\sigma_V(S^U) + c} \geq \text{ASR}^U(S^U, c) \cdot \frac{c}{|V| + c} \Rightarrow \\
\mathbb{E}[\text{ASR}(S^U, c)] &\geq \mathbb{E}\left[\text{ASR}^U(S^U, c) \cdot \frac{c}{|V| + c}\right] \\
&\geq \frac{c}{|V| + c} \cdot \mathbb{E}\left[\text{ASR}^U(S^U, c)\right] \\
&\geq \frac{c}{|V| + c} \cdot \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^U(S^{U,*}, c) \\
&\geq \frac{c}{|V| + c} \cdot \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}^U(S^*, c) \\
&\geq \frac{c}{|V| + c} \cdot \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c).
\end{aligned}$$

Finally, again from SubsampleGreedy,

$$\begin{aligned}
\mathbb{E}[\text{ASR}(S^O, c)] &\geq \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c) \\
&\geq \frac{c}{|V| + c} \cdot \left(1 - e^{-(1-1/e)}\right) \cdot \text{ASR}(S^*, c)
\end{aligned}$$

and after removing the dummy elements, $S^G \in \{S^{O'}, S^{L'}, S^{U'}\}$, which concludes the proof. □

Proof of Proposition 3.18. It suffices to show that the function $f_v(E) = \Pr[v, \mathcal{P}_{S,v}, E]$, $v \in T$ is monotone non-decreasing submodular, and then the same will hold for $g_T(E) = \sum_{v \in T} f_v(E)$ by B.4.

First of all, let E_V be the edge set of G_v, $E_1 \subseteq E_2 \subseteq E_V$ and $e \notin E_2$. In addition, let $\mathcal{P}_{S,v}^F$ be the set of paths from S to v containing edges in $F \subseteq E_V$. We distinguish three cases:

Case I $\mathcal{P}_{S,v}^e \subseteq \mathcal{P}_{S,v}^{E_1}$.

Then adding e will not affect neither $\mathcal{P}_{S,v}^{E_1}$ nor $\mathcal{P}_{S,v}^{E_2}$, i.e., $f_v(E_1 \mid e) = 0$ and $f_v(E_2 \mid e) = 0$, so the submodularity property holds in a trivial manner.

Case II $\mathcal{P}_{S,v}^e \subseteq \mathcal{P}_{S,v}^{E_2}$ but $\mathcal{P}_{S,v}^e \subsetneq \mathcal{P}_{S,v}^{E_1}$.

Then adding e will only add paths to $\mathcal{P}_{S,v}^{E_1}$, i.e., $f_v(E_1 \mid e) \geq 0 = f_v(E_2 \mid e)$.

Case III $\mathcal{P}_{S,v}^e \subsetneq \mathcal{P}_{S,v}^{E_2}$.

Then adding e will add to $\mathcal{P}_{S,v}^{E_1}$ all the paths that it will add to $\mathcal{P}_{S,v}^{E_2}$ plus all those other paths of $\mathcal{P}_{S,v}^e$ contained in $\mathcal{P}_{S,v}^{E_2} \setminus \mathcal{P}_{S,v}^{E_1}$, i.e., $f_v(E_1 \mid e) \geq f_v(E_2 \mid e)$.

Hence, f_v is submodular.

Moreover, f_v is monotone since for every $e \notin F$ it holds that $f_v(F \cup \{e\}) \geq f_v(F)$ and non-decreasing since $f_v(E_2) \geq f_v(E_1)$, for the aforementioned E_1, E_2. □

Proof of Lemma 3.20. Since $y_i \leq \max_{i\in[n]}\{y_i x_i\}/x_i$, we get

$$\begin{aligned} S &\leq \max_{i\in[n]}\{y_i x_i\}\left[\sum_{i=1}^{n-1}\left(1-\frac{x_{i+1}}{x_i}\right)+1\right] \\ &\leq \max_{i\in[n]}\{y_i x_i\}\left[1+\sum_{i=1}^{n-1}\ln\left(\frac{x_{i+1}}{x_i}\right)\right] \\ &= \max_{i\in[n]}\{y_i x_i\}\left[1+\ln\left(\frac{x_1}{x_n}\right)\right], \end{aligned}$$

where the second inequality holds because $1-1/x \leq \ln x, \forall x \geq 1$. Additionally, $S = y_1 x_1 + \sum_{i=1}^{n-1}(y_{i+1}-y_i)x_{i+1}$ and since $x_i \leq \max_{i\in[n]}\{y_i x_i\}/y_i$ we get by a very similar argument that

$$S \leq \max_{i\in[n]}\{y_i x_i\}\left[1+\ln\left(\frac{y_n}{x_1}\right)\right],$$

which completes our proof. □