

The Open University of Israel
Department of Mathematics and Computer Science

**Maximizing Sums of Non-monotone Submodular and Linear
Functions: Understanding the Unconstrained Case**

By
Kobi Bodek

Thesis submitted as partial fulfillment of the requirements
towards an M.Sc. degree in Computer Science
The Open University of Israel
Department of Mathematics and Computer Science

Prepared under the supervision of
Professor Moran Feldman and Professor Zeev Nutov

December 2022

Contents

1. Introduction	1
a. Thesis Structure	4
2. Preliminaries	4
a. Set Functions and Notation	4
b. Multilinear Extension	5
c. Value Oracle	5
3. Inapproximability for Monotone Functions	5
4. Algorithm for the General Case	7
5. Inapproximability for Negative Linear Functions	14
6. Results for Positive Linear Functions	17
a. Impossibility of the Naturally Expected Approximation Guarantee	17
b. Reanalysis of Deterministic Double Greedy	20
c. Reanalysis of Randomized Double Greedy	21
7. Appendix	23
a. Extended Result for Non-Positive Linear Functions	23
b. Proof of Theorem 3.1	26

List of Figures

1. Graphical presentation of the existing results for RegularizedUSM with a non-positive linear function	3
--	---

List of Algorithms

1. Non-oblivious Local Search	10
2. Deterministic Double Greedy	20
3. Randomized Double Greedy	22
4. Distorted Measured Continuous Greedy	24

Abstract

Motivated by practical applications, recent works have considered maximization of sums of a submodular function g and a linear function ℓ . Almost all such works, to date, studied only the special case of this problem in which g is also guaranteed to be monotone. Therefore, in this thesis we systematically study the simplest version of this problem in which g is allowed to be non-monotone, namely the unconstrained variant, which we term **Regularized Unconstrained Submodular Maximization** (**RegularizedUSM**).

Our main algorithmic result is the first non-trivial guarantee for general **RegularizedUSM**. For the special case of **RegularizedUSM** in which the linear function ℓ is non-positive, we prove two inapproximability results, showing that the algorithmic result implied for this case by previous works is not far from optimal. Finally, we reanalyze the known Double Greedy algorithm to obtain improved guarantees for the special case of **RegularizedUSM** in which the linear function ℓ is non-negative; and we complement these guarantees by showing that it is not possible to obtain $(1/2, 1)$ -approximation for this case (despite intuitive arguments suggesting that this approximation guarantee is natural).

1 Introduction

The field of submodular optimization has been rapidly developing over the last two decades, partially due to new applications. Some of these applications have also motivated the optimization of composite objective functions that can be represented as the sum of a submodular function g and a linear function ℓ . Let us briefly discuss two such applications

The first application is *optimization with a regularizer*. To avoid overfitting in machine-learning, it is customary to optimize a function of the form $g - \ell$, where g is the quantity that we would like to maximize and ℓ is a (often linear) function that favors small solutions. This function ℓ is known as “regularizer” in the machine learning jargon, or “soft constraint” in the operations research jargon.

The other application we discuss is *optimization with a curvature*. Traditionally, the theoretical study of submodular optimization problems looks for approximation guarantees that apply to all submodular functions, or at least all monotone submodular functions. However, approximation guarantees of this kind are often pessimistic, and do not capture the practical performance of the algorithms analyzed. This has motivated studying how the optimal approximation ratios of various submodular maximization problems depend on various numerical function properties. Historically, the first property of this kind to be defined was the *curvature* property, which was suggested by Conforti and Cornuéjol [5] already in 1984. The curvature measures the distance of the submodular function from being linear, and a strong connection was demonstrated by Sviridenko et al. [18] between optimizing a submodular function with a given curvature and optimizing the sum $g + \ell$ of a monotone submodular function g and a linear function ℓ .

Motivated by the above applications, Sviridenko et al. [18] also initialized the study of the optimization of $g + \ell$ sums. In particular, they described algorithms with optimal approximation guarantees for this problem when g is a non-negative monotone submodular function, ℓ is a linear function and the optimization is subject to either a matroid or a cardinality constraint.¹ Later works obtained faster and semi-streaming algorithms for the same setting [8, 11, 12, 15]. However, in contrast to all these (often tight) results for monotone submodular functions g , much less is known about the case of non-monotone submodular functions. In fact, we are only aware of a single previous work that considered $g + \ell$ sums involving such functions [13].²

Given the rarity of results so far for optimizing $g + \ell$ with a function g that is non-monotone, this thesis is devoted to a systematic study of the simplest problem of this kind, namely, unconstrained maximization of such sums. Formally, we study the **Regularized Unconstrained Submodular Maximization** (**RegularizedUSM**) problem. In this problem, we are given a non-negative submodular function $g: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ and a linear function $\ell: 2^{\mathcal{N}} \rightarrow \mathbb{R}$ over the same ground set \mathcal{N} , and the objective is to output a set $T \subseteq \mathcal{N}$ maximizing the sum $g(T) + \ell(T)$. Unfortunately, it is not possible to prove standard multiplicative approximation ratios for **RegularizedUSM** (implied, e.g., by Theorem 1.3). Therefore, we follow previous works, and look in this work for algorithms that output a (possibly randomized) set $T \subseteq \mathcal{N}$ such that $\mathbb{E}[g(T) + \ell(T)] \geq \max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$ for some coefficients $\alpha, \beta \geq 0$. For convenience, we say that an algorithm having this guarantee is an (α, β) -approximation algorithm.³

It is instructive to begin the study of **RegularizedUSM** with the special case in which the objec-

¹Technically, Sviridenko et al. [18] proved optimal approximation guarantees only for the case in which the coefficient β of ℓ is 1 (see details below). However, their results were extended to the general case of $\beta \geq 0$ by Feldman [8].

²Very recently, another work of this kind appeared as a pre-print [17]. However, the main result of [17] is identical to the result of [13]. In particular, it is important to note that the result of [17] applies only to non-positive ℓ functions, like the result of [13], although this is not explicitly stated in [17].

³Some previous works compare their algorithms against $\alpha \cdot g(OPT) + \beta \cdot \ell(OPT)$, where OPT is a feasible set maximizing $g(OPT) + \ell(OPT)$, instead of comparing against $\max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$ like we do in this thesis. This distinction is usually of little consequence.

tive function g is guaranteed to be monotone (in addition to being non-negative and submodular). We refer below to this special case as “monotone `RegularizedUSM`”. The work of Feldman [8] on constrained maximization of $g + \ell$ immediately implies $(1 - e^{-\beta}, \beta)$ -approximation for monotone `RegularizedUSM` for every $\beta \in [0, 1]$. Our first result provides a matching inapproximability result.

Theorem 1.1. *For every $\beta \geq 0$ and $\varepsilon > 0$, no polynomial time algorithm can guarantee $(1 - e^{-\beta} + \varepsilon, \beta)$ -approximation for monotone `RegularizedUSM` even when the linear function ℓ is guaranteed to be non-positive.*

We would like to draw attention to two properties of Theorem 1.1. First, for $\beta = 1$ the coefficient of g in the inapproximability proved by the theorem is $1 - 1/e$, matching the optimal approximation ratio for the problem of maximizing a monotone submodular function subject to a matroid constraint. Therefore, in a sense, adding the linear part ℓ makes the unconstrained problem as hard as this constrained problem. Interestingly, we get a similar result for `RegularizedUSM` below.

The other noteworthy property of Theorem 1.1 is that it applies to any $\beta \geq 0$, while the algorithmic result of Feldman [8] applies only to $\beta \in [0, 1]$. This difference between the results exists because, when ℓ can take positive values, setting the coefficient β to be larger than 1 might require the algorithm to output a set $T \subseteq \mathcal{N}$ obeying $\ell(T) > \max_{S \subseteq \mathcal{N}} \ell(S)$. However, it turns out that, when ℓ is non-positive, the algorithmic result can be extended to match Theorem 1.1 for every $\beta \geq 0$. To understand how this can be done, we need to discuss the previous work in a bit more detail.

Sviridenko et al. [18] designed two algorithms for maximizing $g + \ell$ sums, one of which was based on the continuous greedy algorithm of Călinescu et al. [4]. It is possible to modify this algorithm to be based instead on a related algorithm called “measured continuous greedy” due to [9]. In general, this does not lead to any result for maximizing $g + \ell$ sums. However, Lu et al. [13] recently observed that one can obtain in this way results when ℓ is non-positive. In particular, it leads to $(1 - e^{-\beta}, \beta)$ -approximation for the special case of monotone `RegularizedUSM` in which ℓ is non-positive for any constant $\beta \geq 0$, which settles the approximability of monotone `RegularizedUSM`.

We now get to the study of (not necessarily monotone) `RegularizedUSM`. The only result that is known to date for this problem is $(1/e, 1)$ -approximation for the special case in which ℓ is non-positive, which was proved by Lu et al. [13] using the technique discussed above. Our main algorithmic contribution is the first algorithm with a non-trivial approximation guarantee for general `RegularizedUSM`.

Theorem 1.2. *For every constant $\beta \in (0, 1]$, let us define $\alpha(\beta) = \beta(1 - \beta)/(1 + \beta)$. Then, for every constant $\varepsilon \in (0, \alpha(\beta))$, there exists a polynomial time $(\alpha(\beta) - \varepsilon, \beta - \varepsilon)$ -approximation algorithm for `RegularizedUSM`.*

We also study in more detail the special cases of `RegularizedUSM` in which ℓ is either non-negative or non-positive. The above mentioned result of Lu et al. [13] for `RegularizedUSM` with a non-positive ℓ can be extended (using the ideas of Feldman [8]) to get $(\beta e^{-\beta}, \beta)$ -approximation for the same special case for any $\beta \in [0, 1]$ ⁴ (see Appendix A for more details). It is not immediately clear, however, how good this extended result is. For example, one can compare it with the inapproximability result of Theorem 1.1 (which applies to the current setting as well), but there is a large gap between the above algorithmic and inapproximability results when the β coefficient of ℓ is relatively large (see Figure 1). This gap exists because Theorem 1.1 holds even in the special case

⁴Technically, this result can be extended to any constant $\beta \geq 0$, but this is not interesting since $\beta e^{-\beta}$ is a decreasing function for $\beta \geq 1$.

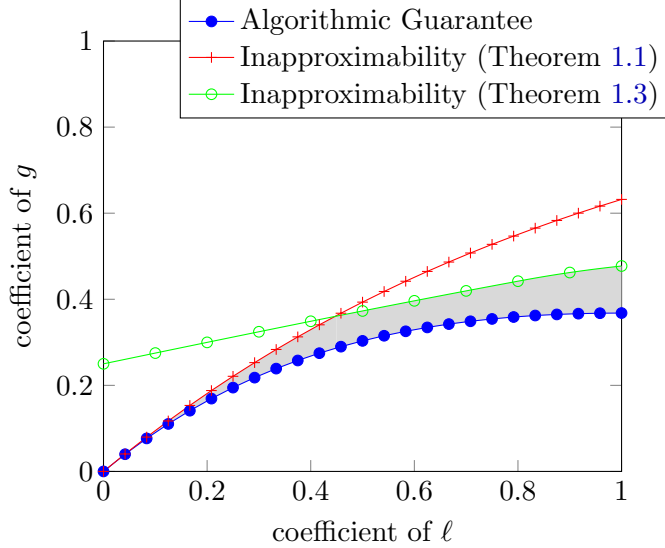


Figure 1: Graphical presentation of the existing results for **RegularizedUSM** with a non-positive linear function ℓ . The x and y axes represent the coefficients of ℓ and g , respectively. The algorithmic guarantee drawn is the $(\beta e^{-\beta}, \beta)$ -approximation obtainable by generalizing Lu et al. [13]. The shaded area represents the gap that still exists between the best known approximation guarantee and inapproximability results.

in which g is monotone. Therefore, we prove the following theorem, which provides an alternative inapproximability result designed for the non-monotone case. Since it is difficult to understand the behavior of the expression stated in Theorem 1.3, we numerically draw it in Figure 1, which demonstrates that Theorem 1.3 closes much of the gap left with regard to **RegularizedUSM** with non-positive linear function ℓ .

Theorem 1.3. *Given a value $\beta \geq 0$, let us define*

$$\alpha(\beta) = \min_{\substack{t \geq 1 \\ r \in (0, 1/2]}} \left\{ \frac{t + 1 + \sqrt{(t+1)^2 - 8tr}}{4t} - \frac{r}{t+1} \cdot \left[1 - \beta - 2 \ln \left(\frac{t + 1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \right] \right\}.$$

*Then, for every $\varepsilon > 0$, no polynomial time algorithm can guarantee $(\alpha(\beta) + \varepsilon, \beta)$ -approximation for **RegularizedUSM** even when the linear function ℓ is guaranteed to be non-positive.*

It is interesting to note that, for $\beta = 1$, Theorem 1.3 matches the state-of-the-art inapproximability result of Oveis Gharan and Vondrák [16] for maximizing a non-negative submodular function subject to matroid constraint. Therefore, at least at the level of the known inapproximability results, **RegularizedUSM** with a non-positive ℓ is as hard as maximizing a non-negative submodular function subject to a matroid constraint.

It remains to consider the special case of **RegularizedUSM** with a non-negative ℓ . Here $g + \ell$ is a non-negative submodular function on its own right, and therefore, **RegularizedUSM** becomes a special case of the well-studied problem of Unconstrained Submodular Maximization (USM). The optimal approximation ratio for USM is $1/2$ due to an inapproximability result of Feige et al. [7], and the first algorithm to obtain this approximation ratio was the “Double Greedy” algorithm of Buchbinder et al. [3]. Specifically, Buchbinder et al. [3] described two variants of their algorithm,

a deterministic variant guaranteeing $1/3$ -approximation, and a randomized variant guaranteeing $1/2$ -approximation. We refer below to these two variants as `DeterministicDG` and `RandomizedDG`, respectively. Interestingly, we are able to show in the next two theorems that the performance of `DeterministicDG` and `RandomizedDG` for `RegularizedUSM` is even better than what one would expect based on the guarantees of these algorithms for general `USM`.

Theorem 1.4. *When ℓ is non-negative, `DeterministicDG` is an $(\alpha, 1 - \alpha)$ -approximation algorithm for `RegularizedUSM` for all $\alpha \in [0, 1/3]$ at the same time (the algorithm is oblivious to the value of α).*

Theorem 1.5. *When ℓ is non-negative, `RandomizedDG` is an $(\alpha, 1 - \alpha/2)$ -approximation algorithm for `RegularizedUSM` for all $\alpha \in [0, 1/2]$ at the same time (the algorithm is oblivious to the value of α).*

We conclude this section with an interesting observation. Up to this point, the most well studied $g + \ell$ maximization problem was maximizing the sum of a non-negative monotone submodular function g and a linear function ℓ subject to a matroid constraint. When ℓ is positive, the optimal approximation guarantee for this problem is $(1 - 1/e, 1)$ [18], which is natural since $1 - 1/e$ is the optimal approximation ratio for maximizing such a function g subject to a matroid constraint [14]. Thus, one might expect to get $(1/2, 1)$ -approximation for `RegularizedUSM` with a non-negative ℓ . However, both Theorems 1.4 and 1.5 fail to prove such a guarantee, and we are able to show that this is not a coincidence.

Theorem 1.6. *Even when the linear function ℓ is guaranteed to be non-negative, no polynomial time algorithm can guarantee $(1/2, 1)$ -approximation for `RegularizedUSM`.*

Thesis Structure. In Section 2 we give a few formal definitions and explain the notation used throughout the thesis. Then, we prove our inapproximability result for monotone `RegularizedUSM` (Theorem 1.1) in Section 3. Our results for general `RegularizedUSM`, `RegularizedUSM` with non-positive ℓ and `RegularizedUSM` with non-negative ℓ can be found in Sections 4, 5 and 6, respectively.

2 Preliminaries

Set Functions and Notation. Given a set function $f: 2^{\mathcal{N}} \rightarrow \mathbb{R}$, an element $u \in \mathcal{N}$ and a set $S \subseteq \mathcal{N}$, the marginal contribution of u to S with respect to f is $f(u | S) \triangleq f(S \cup \{u\}) - f(S)$. A set function $f: 2^{\mathcal{N}} \rightarrow \mathbb{R}$ is called *submodular* if it satisfies the intuitive property of diminishing returns. More formally, f is submodular if $f(u | S) \geq f(u | T)$ for every two sets $S \subseteq T \subseteq \mathcal{N}$ and element $u \in \mathcal{N} \setminus T$. An equivalent definition of submodularity is that f is submodular if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for every two sets $S, T \subseteq \mathcal{N}$.

The set function f is called *monotone* if $f(S) \leq f(T)$ for every two sets $S \subseteq T \subseteq \mathcal{N}$, and it is called *linear* if there exist values $\{a_u \in \mathbb{R} \mid u \in \mathcal{N}\}$ such that $f(S) = \sum_{u \in S} a_u$ for every set $S \subseteq \mathcal{N}$.⁵ One can verify that any linear set function is submodular, but the reverse does not necessarily hold. Additionally, given a set S , a set function f and an element u , we often use $S + u$, $S - u$ and $f(u)$ as shorthands for $S \cup \{u\}$, $S \setminus \{u\}$ and $f(\{u\})$, respectively.

⁵Linear set functions are also known as *modular* functions.

Multilinear extension. It is often useful to consider continuous extensions of set functions, and there are multiple ways in which this can be done. The proofs of our inapproximability results employ one such extension known as the *multilinear extension* (due to [4]). Formally, given a set function $f: 2^{\mathcal{N}} \rightarrow \mathbb{R}$, its multilinear extension is the function $F: [0, 1]^{\mathcal{N}} \rightarrow \mathbb{R}$ defined, for every vector $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, by $F(\mathbf{x}) = \mathbb{E}[f(\mathbf{R}(\mathbf{x}))]$, where $\mathbf{R}(\mathbf{x})$ is a random subset of \mathcal{N} including every element $u \in \mathcal{N}$ with probability x_u , independently.

One can verify that, as is suggested by its name, the multilinear extension F is a multilinear function of the coordinates of its input vector. Furthermore, F is an extension of the set function f in the sense that for every set $S \subseteq \mathcal{N}$ we have $F(\mathbf{1}_S) = f(S)$, where $\mathbf{1}_S$ is the characteristic vector of the set S (i.e., a vector that has the value 1 in coordinates corresponding to elements of S , and the value 0 in the other coordinates).

Lovász extension. Given a vector $x \in [0, 1]^{\mathcal{N}}$ and a scalar $\lambda \in [0, 1]$, let $T_\lambda(x) = \{u \in \mathcal{N} | x_u \geq \lambda\}$ be the set of elements in \mathcal{N} whose coordinate in x is at least λ . Then,

$$\hat{f}(x) = \int_{\lambda=0}^1 f(T_\lambda(x)) d\lambda.$$

Lovász extension can also be interpreted in probabilistic terms as the expected value of f over the set $T_\lambda(x)$, where λ is selected uniformly at random from the range $[0, 1]$.

Value Oracle. As is standard in the submodular optimization literature, we assume in this thesis that algorithms access their set function inputs only through *value oracles*. A value oracle for a set function f is a black box that given a set $S \subseteq \mathcal{N}$ returns $f(S)$. One advantage of this convention is that it makes it possible to use information theoretic arguments to prove unconditional inapproximability results (i.e., inapproximability results that are not based on any complexity assumption). Nevertheless, if necessary, these inapproximability results can usually be adapted to apply also to succinctly represented functions (instead of functions accessed via value oracles) at the cost of introducing some complexity assumption [6].

3 Inapproximability for Monotone Functions

In this section we show an inapproximability for monotone **RegularizedUSM** (Theorem 1.1). All our inapproximability results in this thesis are proved using Theorem 3.1. Since the proof of this theorem is a relatively straightforward adaptation of the symmetry gap framework of Vondrák [19], we defer it to Appendix B.

Theorem 3.1. *Consider an instance (g, ℓ) of **RegularizedUSM** consisting of a non-negative submodular function $g: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ and a linear function $\ell: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$, and assume that there exists a group \mathcal{G} of permutations over \mathcal{N} such that the equalities $g(S) = g(\sigma(S))$ and $\ell(S) = \ell(\sigma(S))$ hold for all sets $S \subseteq \mathcal{N}$ and permutations $\sigma \in \mathcal{G}$. Let G and L be the multilinear extensions of g and ℓ respectively, and for every vector $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, let us denote $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{x})]$, i.e., $\bar{\mathbf{x}}$ is the expected vector $\sigma(\mathbf{x})$ when σ is picked uniformly at random out of \mathcal{G} . For any two constants $\alpha, \beta \geq 0$, if $\max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive and*

$$\max_{\mathbf{x} \in [0, 1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] ,$$

*then no polynomial time algorithm for **RegularizedUSM** can guarantee $((1 + \varepsilon)\alpha, (1 + \varepsilon)\beta)$ -approximation for any positive constant ε . Furthermore, this inapproximability guarantee holds also*

when we restrict attention to instances (g', ℓ') of **RegularizedUSM** having the following additional properties.

- If ℓ is non-negative or non-positive, then we can assume that ℓ' also has the same property.
- If g is monotone, then we can assume that g' is monotone as well.

In the common case in which the linear function ℓ is a non-positive, the following observation allows us to produce slightly cleaner results using Theorem 3.1.

Observation 3.2. *If ℓ is non-positive and $\alpha > 0$, then one can replace the term “ $((1+\varepsilon)\alpha, (1+\varepsilon)\beta)$ -approximation” in Theorem 3.1 with the term “ $(\alpha + \varepsilon, \beta)$ -approximation”.*

Proof. Recall that Theorem 3.1 proves, under some conditions, that no polynomial time algorithm for **RegularizedUSM** has $((1 + \varepsilon)\alpha, (1 + \varepsilon)\beta)$ -approximation. Furthermore, if we reduce the value of the constant parameter ε of the theorem by a factor of α , then the theorem also shows that no such algorithm can guarantee $(\alpha + \varepsilon, \beta + \varepsilon\beta/\alpha)$ -approximation. This implies the observation since, when ℓ is non-positive, any $(\alpha + \varepsilon, \beta)$ -approximation algorithm for **RegularizedUSM** is also an $(\alpha + \varepsilon, \beta + \varepsilon\beta/\alpha)$ -approximation algorithm. \square

To prove Theorem 1.1 using Theorem 3.1, we need to define an instance \mathcal{I} of monotone **RegularizedUSM**. Specifically, consider a ground set \mathcal{N} of size $n \geq 2$ and a value $r \in (0, 1]$, and let us define

$$g(S) = \min\{|S|, 1\} \quad \text{and} \quad \ell(S) = -r \cdot |S| \quad \forall S \subseteq \mathcal{N} .$$

Lemma 3.3. *For any constants $\varepsilon > 0$, $\beta \geq 0$ and $\alpha = 1 - e^{-\beta} + \varepsilon$, when n is large enough, there exists a value $r \in (0, 1]$ such that the inequality of Theorem 3.1 applies to \mathcal{I} and $\max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive.*

Proof. Observe that $\max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)] \geq \alpha - \beta r = 1 - e^{-\beta} + \varepsilon - \beta r$ because S can be chosen as a singleton subset of \mathcal{N} . Let us now study the left hand side of the inequality of Theorem 3.1. Since both g and ℓ are unaffected when an arbitrary permutation is applied to the ground set, we can choose \mathcal{G} as the group of all permutations over \mathcal{N} . Thus, for every vector $\mathbf{x} \in [0, 1]^{\mathcal{N}}$,

$$\bar{\mathbf{x}} = \frac{\|\mathbf{x}\|_1}{n} \cdot \mathbf{1}_{\mathcal{N}} .$$

Therefore,

$$\begin{aligned} \max_{\mathbf{x} \in [0, 1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &= \max_{x \in [0, 1]} [G(x \cdot \mathbf{1}_{\mathcal{N}}) + L(x \cdot \mathbf{1}_{\mathcal{N}})] \\ &= \max_{x \in [0, 1]} [1 - (1 - x)^n - xrn] = 1 - r - r(n - 1)[1 - r^{1/(n-1)}] , \end{aligned}$$

where the last equality holds since the maximum is obtained for $x = 1 - \sqrt[n-1]{r}$. Note now that if we denote $y = (n - 1)^{-1}$, then by L'Hôpital's rule,

$$\lim_{n \rightarrow \infty} (n - 1)[1 - r^{1/(n-1)}] = \lim_{y \rightarrow 0} \frac{1 - r^y}{y} = \lim_{y \rightarrow 0} \frac{-r^y \ln r}{1} = -\ln r ,$$

and therefore, for a large enough n , $(n - 1)[1 - r^{1/(n-1)}] \geq -\ln r - \varepsilon$; which implies

$$\max_{\mathbf{x} \in [0, 1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq 1 - r - r[-\ln r - \varepsilon] \leq 1 - r(1 - \ln r) + \varepsilon .$$

Given the above bounds, we get that the inequality of Theorem 3.1 holds for any $r > 0$ obeying

$$1 - e^{-\beta} + \varepsilon - \beta r \geq 1 - r(1 - \ln r) + \varepsilon .$$

Since the last inequality is equivalent to

$$r - r \ln r \geq e^{-\beta} + \beta r ,$$

it holds for $r = e^{-\beta} \subseteq (0, 1]$. Furthermore, for this choice of r ,

$$\max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \geq 1 - e^{-\beta} + \varepsilon - \beta r = 1 - (1 + \beta)e^{-\beta} + \varepsilon \geq 1 - \frac{1 + \beta}{1 + \beta} + \varepsilon = \varepsilon > 0 . \quad \square$$

Theorem 1.1, which we repeat here for convenience, now follows by combining Theorem 3.1, Observation 3.2 and Lemma 3.3 since g is a non-negative monotone submodular function and ℓ is a non-positive linear function.

Theorem 1.1. *For every $\beta \geq 0$ and $\varepsilon > 0$, no polynomial time algorithm can guarantee $(1 - e^{-\beta} + \varepsilon, \beta)$ -approximation for monotone **RegularizedUSM** even when the linear function ℓ is guaranteed to be non-positive.*

4 Algorithm for the General Case

In this section we describe and analyze the only non-trivial algorithm known to date (as far as we know) for general **RegularizedUSM**. Using this algorithm we prove Theorem 1.2, which we repeat here for convenience.

Theorem 1.2. *For every constant $\beta \in (0, 1]$, let us define $\alpha(\beta) = \beta(1 - \beta)/(1 + \beta)$. Then, for every constant $\varepsilon \in (0, \alpha(\beta))$, there exists a polynomial time $(\alpha(\beta) - \varepsilon, \beta - \varepsilon)$ -approximation algorithm for **RegularizedUSM**.*

Our algorithm is based on a non-oblivious local search, i.e., a local search guided by an auxiliary function rather than the objective function. Non-oblivious local searches have been used previously in the context of submodular maximization by, for example, Feige et al. [7] and Filmus and Ward [10]. The auxiliary function used by our algorithm is a function $h: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ defined as follows. For every set $S \subseteq \mathcal{N}$,

$$h(S) = \mathbb{E}[g(S(\beta))] + \beta(1 + \beta) \cdot \ell(S) ,$$

where $S(\beta)$ is a random subset of S that includes every element of S with probability β , independently.

Ideally, we would like to find a local maximum with respect to h , i.e., a set $T \subseteq \mathcal{N}$ such that the value of $h(T)$ cannot be increased either by adding a single element to T , or by removing a single element from T . However, there are two issues that make the task of finding such a local maximum difficult.

- We do not know how to exactly evaluate the expectation in the definition of h in polynomial time. Therefore, whenever we need to calculate expressions involving h , we have to approximate them using sampling, which introduces estimation errors that have to be taken into account.

- A straightforward local search algorithm changes its current solution whenever adding or removing a single element improves this solution. However, the time complexity of such a naïve algorithm can be exponential. Therefore, our algorithm adds or removes an element only when this is beneficial enough, which means that the algorithm finds an approximate local maximum rather than a true one. Employing this idea is not trivial given the errors introduced by the sampling, as mentioned above. However, we manage to prove that, for the value Δ defined by our algorithm, with high probability: (i) the algorithm only makes changes that increase the value of $h(T)$ by $\Delta/2$ or more, and (ii) the algorithm continues to make changes as long as there exists some possible change that increases the value of $h(T)$ by at least $3\Delta/2$.

The quality of the approximate local maximum produced by our algorithm is controlled by the parameter ε of Theorem 1.2. Setting a lower value for ε decreases Δ , which increases the time complexity of our algorithm, but also makes the approximate local maximum produced closer to being a true local maximum, and thus, improves the approximation guarantee.

Let \hat{S} be a subset of \mathcal{N} maximizing $(\alpha(\beta) - \varepsilon) \cdot g(\hat{S}) + (\beta - \varepsilon) \cdot \ell(\hat{S})$. To implement the solutions described in the last two bullets, it is useful to assume that the ground set \mathcal{N} does not include elements that have some problematic properties. The following reduction shows that we can assume that this is indeed the case without loss of generality.

Reduction 1. *While proving Theorem 1.2, we may assume that every element $u \in \mathcal{N}$ obeys*

$$\alpha(\beta) \cdot g(u) + \beta \cdot \ell(u) \geq 0 \quad \text{and} \quad \max\{g(u) + \ell(u), g(\emptyset)\} \leq \beta \cdot [g(\hat{S}) + \ell(\hat{S})] .$$

Proof. Let us begin by proving the second part of the reduction. If there exists an element u for which

$$\max\{g(u) + \ell(u), g(\emptyset) + \ell(\emptyset)\} = \max\{g(u) + \ell(u), g(\emptyset)\} > \beta \cdot [g(\hat{S}) + \ell(\hat{S})] ,$$

then one can obtain an algorithm with the guarantee stated in Theorem 1.2 by simply returning the set $T \in \{\emptyset\} \cup \{\{u\} \mid u \in \mathcal{N}\}$ maximizing $g(T) + \ell(T)$ because

$$\begin{aligned} g(T) + \ell(T) &\geq \max\{g(u) + \ell(u), g(\emptyset) + \ell(\emptyset)\} > \beta \cdot [g(\hat{S}) + \ell(\hat{S})] \\ &= \frac{\beta}{\beta - \varepsilon} \cdot [(\beta - \varepsilon) \cdot g(\hat{S}) + (\beta - \varepsilon) \cdot \ell(\hat{S})] \geq \frac{\beta}{\beta - \varepsilon} \cdot [(\alpha(\beta) - \varepsilon) \cdot g(\hat{S}) + (\beta - \varepsilon) \cdot \ell(\hat{S})] \\ &= \frac{\beta}{\beta - \varepsilon} \cdot \max_{S \subseteq \mathcal{N}} [(\alpha(\beta) - \varepsilon) \cdot g(S) + (\beta - \varepsilon) \cdot \ell(S)] \geq \max_{S \subseteq \mathcal{N}} [(\alpha(\beta) - \varepsilon) \cdot g(S) + (\beta - \varepsilon) \cdot \ell(S)] . \end{aligned}$$

The penultimate inequality holds since g is non-negative, the last equality follows from the definition of \hat{S} , and the last inequality holds since the maximum on both sides of this inequality is non-negative (this can be seen by choosing $S = \emptyset$). It is also worth mentioning the above algorithm, namely, outputting the set $T \in \{\emptyset\} \cup \{\{u\} \mid u \in \mathcal{N}\}$ maximizing $g(T) + \ell(T)$ can be implemented to run in linear time since it only has to consider $|\mathcal{N}| + 1$ candidate sets.

It remains to prove the first part of the reduction. Assume that there exists an algorithm *ALG* that has the guarantee stated in Theorem 1.2 for instances obeying the first part of the reduction, and let us explain how to get an algorithm that has the same guarantee for general instances of *RegularizedUSM*. Towards this goal, let us define

$$\mathcal{N}' = \{u \in \mathcal{N} \mid \alpha(\beta) \cdot g(u) + \beta \cdot \ell(u) \geq 0\} .$$

In other words, \mathcal{N}' is the subset of \mathcal{N} that includes all the elements obeying the first part of the reduction. We claim that $\hat{S} \subseteq \mathcal{N}'$. If this is not true, then, by the submodularity and non-negativity of g , for any element $u \in \hat{S} \setminus \mathcal{N}'$ we have

$$\begin{aligned} (\alpha(\beta) - \varepsilon) \cdot [g(\hat{S}) - g(\hat{S} - u)] + (\beta - \varepsilon) \cdot [\ell(\hat{S}) - \ell(\hat{S} - u)] &\leq (\alpha(\beta) - \varepsilon) \cdot g(u \mid \emptyset) + (\beta - \varepsilon) \cdot \ell(u) \\ &\leq (\alpha(\beta) - \varepsilon) \cdot g(u) + (\beta - \varepsilon) \cdot \ell(u) \leq \frac{\beta - \varepsilon}{\beta} \cdot [\alpha(\beta) \cdot g(u) + \beta \cdot \ell(u)] < 0 , \end{aligned}$$

which contradicts the definition of \hat{S} (the penultimate inequality holds since $\alpha(\beta) \leq \beta$ and g is non-negative).

Observe now that by the definition of \mathcal{N}' , we can execute ALG on \mathcal{N}' , which produces a set T obeying

$$\mathbb{E}[g(T) + \ell(T)] \geq \max_{S \subseteq \mathcal{N}'} [(\alpha(\beta) - \varepsilon) \cdot g(S) + (\beta - \varepsilon) \cdot \ell(S)] = \max_{S \subseteq \mathcal{N}} [(\alpha(\beta) - \varepsilon) \cdot g(S) + (\beta - \varepsilon) \cdot \ell(S)] ,$$

where the equality holds since the maximum in the rightmost side is obtained for $S = \hat{S}$, and \hat{S} is a subset of \mathcal{N}' as we have proved above. Since one can construct \mathcal{N}' in linear time, executing ALG on \mathcal{N}' is the promised algorithm that achieves the guarantee of Theorem 1.2 without assuming the first part of the reduction. \square

From this point until the end of the section, we denote by n the size of the ground set \mathcal{N} . We are now ready to describe our algorithm (given as Algorithm 1). This algorithm implicitly assumes that Reduction 1 was applied, that n is large enough and that $\max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} > 0$.⁶ The algorithm maintains a solution T , which it updates in iterations. In each iteration, the algorithm calculates for every element u an estimate ω_u of the contribution of u to the g component of the auxiliary function h . Then, Line 5 of the algorithm looks for an element $u \in \mathcal{N} \setminus T$ which, based on the estimate ω_u , will increase $h(T)$ by Δ if added to T . If such an element u is found, the algorithm adds it to T and continues to the next iteration. Otherwise, Line 6 looks for an element $u \in T$ which will increase $h(T)$ by Δ if removed from T (again, based on the estimate ω_u). If such an element u is found, then the algorithm removes it from T and continues to the next iteration. However, if both Lines 5 and 6 fail to find an appropriate element, the algorithm assumes that it has encountered an approximate local maximum, and terminates. Somewhat surprisingly, when this happens the algorithm outputs a sample \hat{T} of $T(\beta)$ rather than the solution T itself (unless the value of this sample is negative, in which case the algorithm falls back to the solution \emptyset). We show below that if T is an approximate local maximum of the auxiliary function h , then $T(\beta)$ is in expectation a good solution with respect to the objective function.

It is clear that Algorithm 1 runs in polynomial time, and therefore, we concentrate in the rest of this section on proving its approximation guarantee. Algorithm 1 makes multiple estimation during its execution. We say that an estimate ω_u is *good* if $|\omega_u - \mathbb{E}[g(u \mid T(\beta) - u)]| \leq \Delta/2$ (for the set T at the time in which the estimate was made), otherwise the estimate is *bad*.

Lemma 4.1. *With high probability (a probability approaching 1 when n tends to infinity), all the estimates made by Algorithm 1 are good.*

⁶Let us explain why the problem becomes easy if either of the last two assumptions is violated. If n is bounded by a constant, it is possible to use exhaustive search to find the set $T \subseteq \mathcal{N}$ maximizing $g(T) + h(T)$, and one can verify that such a set has the properties guaranteed by Theorem 1.2. Additionally, if $\max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} = 0$, then the submodularity of g guarantees that g is the zero function, which means that we can get the guarantee of Theorem 1.2 by outputting the set $\{u \in \mathcal{N} \mid \ell(u) > 0\}$.

Algorithm 1: Non-oblivious Local Search (β, ε)

- 1 Let $\Delta \leftarrow \frac{\varepsilon}{2n} \cdot \max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\}$.
- 2 Let $T \leftarrow \{u \in \mathcal{N} \mid \ell(u) > 0\}$.
- 3 **for** $i = 1$ **to** $\lceil 4n^2/\varepsilon \rceil + 1$ **do**
- 4 **for every** $u \in \mathcal{N}$ **do** Let ω_u be an estimate of $\beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)]$ obtained by taking the average of $\beta \cdot g(u \mid T(\beta) - u)$ for $k = \lceil 128n^4\varepsilon^{-2}\beta^2 \cdot \ln(10n^4/\varepsilon) \rceil$ independent samples of $T(\beta)$.
- 5 **if** there exists $u \in \mathcal{N} \setminus T$ such that $\omega_u + \beta(1 + \beta) \cdot \ell(u) \geq \Delta$ **then** Update $T \leftarrow T + u$.
- 6 **else if** there exists $u \in T$ such that $\omega_u + \beta(1 + \beta) \cdot \ell(u) \leq -\Delta$ **then** Update $T \leftarrow T - u$.
- 7 **else** Exit the “for” loop.
- 8 Let \hat{T} be a sample of $T(\beta)$.
- 9 **if** $g(\hat{T}) + \ell(\hat{T}) \geq 0$ **then return** \hat{T} .
- 10 **else return** \emptyset .

Proof. Consider a particular estimate ω_u made by Algorithm 1, and recall that the algorithm makes this estimate by averaging $\beta \cdot g(u \mid T(\beta) - u)$ for k independent samples of $T(\beta)$. Let T_1, T_2, \dots, T_k denote the samples of $T(\beta)$ used by the algorithm, and let us define, for every integer $1 \leq i \leq k$,

$$X_i = \frac{g(u \mid T_i - u) - \mathbb{E}[g(u \mid T(\beta) - u)]}{4n^2\Delta/\varepsilon}.$$

Note that we deterministically have $|X_i| \leq 1$ because for every set $S \subseteq \mathcal{N}$ and element $u \in \mathcal{N}$ it holds that

$$g(u \mid S - u) \leq g(u \mid \emptyset) = g(u) - g(\emptyset) \leq g(u) \leq \frac{2n\Delta}{\varepsilon}$$

(the first inequality follows from the submodularity of g , the second from g 's non-negativity, and the last from the definition of Δ), and

$$\begin{aligned} g(u \mid S - u) &= g(S + u) - g(S - u) \geq -g(S - u) \\ &\geq -\max\{g(\emptyset), |S - u| \cdot \max_{v \in \mathcal{N}} g(v)\} \geq n \cdot \max\{g(\emptyset), \max_{v \in \mathcal{N}} g(v)\} = -\frac{2n^2\Delta}{\varepsilon}, \end{aligned}$$

where the first and last inequalities hold since g is non-negative and the last equality follows from the definition of Δ . To justify the second inequality, note that, by the submodularity and non-negativity of g ,

$$g(S - u) \leq g(\emptyset) + \sum_{v \in S - u} g(v \mid \emptyset) = (1 - |S - u|) \cdot g(\emptyset) + \sum_{v \in S - u} g(v) \leq \max\{g(\emptyset), |S - u| \cdot \max_{v \in \mathcal{N}} g(v)\}.$$

We can now upper bound the probability that ω_u is a bad estimate as follows.

$$\begin{aligned} \Pr[|\omega_u - \beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)]| > \Delta/2] &= \Pr\left[\frac{4\beta n^2\Delta}{\varepsilon} \cdot \frac{\sum_{i=1}^k X_i}{k} > \frac{\Delta}{2}\right] = \Pr\left[\sum_{i=1}^k X_i > \frac{\varepsilon k}{8\beta n^2}\right] \\ &\leq 2e^{-(\varepsilon k / (8\beta n^2))^2 / (2k)} = 2e^{-\varepsilon^2 k / (128\beta^2 n^4)} \leq 2^{-\ln(10n^4/\varepsilon)} = \frac{\varepsilon}{5n^4}, \end{aligned}$$

where the first inequality follows from the Chernoff-like Theorem A.1.16 of [1], and the second inequality holds by the definition of k . To conclude the proof of the lemma, it remains to observe

that Algorithm 1 makes at most $5n^3/\varepsilon$ estimates since it makes only n estimates per iteration, and has at most $\lceil 4n^2/\varepsilon \rceil + 1 \leq 5n^2/\varepsilon$ iterations (the inequality holds for a large enough n). Therefore, by the union bound, the probability that any estimate made by this algorithm is bad can be upper bounded by $1/n$. \square

Using the previous lemma, we can now prove that, with high probability, Algorithm 1 terminates with T being an approximate local maximum.

Lemma 4.2. *With high probability, when Algorithm 1 terminates we have*

$$h(T) \geq h(T + u) - 3\Delta/2 \quad \forall u \in \mathcal{N} \setminus T \quad \text{and} \quad h(T) \geq h(T - u) - 3\Delta/2 \quad \forall u \in T .$$

Proof. We prove that the lemma holds deterministically when all the estimates made by Algorithm 1 are good, which is a high probability event by Lemma 4.1. Our first step is to show that given this assumption the value of $h(T)$ increases by at least $\Delta/2$ following every iteration of the algorithm unless this iteration terminates by Line 7 (and therefore, does not modify T). If Algorithm 1 added an element $u \in \mathcal{N} \setminus T$ to T during the iteration, then the new value of $h(T)$ is

$$\begin{aligned} h(T + u) &= h(T) + \beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)] + \beta(1 + \beta) \cdot \ell(u) \\ &\geq h(T) + (\omega_u - \Delta/2) + \beta(1 + \beta) \cdot \ell(u) \geq h(T) + \Delta/2 , \end{aligned}$$

where the last inequality follows from the condition of Line 5. Similarly, if Algorithm 1 removed an element $u \in T$ from T during the iteration, then, by the condition on Line 6, the new value of $h(T)$ is

$$\begin{aligned} h(T - u) &= h(T) - \beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)] - \beta(1 + \beta) \cdot \ell(u) \\ &\geq h(T) - (\omega_u + \Delta/2) - \beta(1 + \beta) \cdot \ell(u) \geq h(T) + \Delta/2 . \end{aligned}$$

We now argue that Algorithm 1 must reach Line 7 at some point. Assume towards a contradiction that this does not happen, which by the above observation implies that the algorithm increases the value of $h(T)$ by at least $\Delta/2$ in each one of its $\lceil 4n^2/\varepsilon + 1 \rceil$ iterations. Additionally, if we denote by T' the final value of the set T , then the initialization of T and the non-negativity of g guarantee together that the original value of $h(T)$ before the first iteration of Algorithm 1 is at least $\beta(1 + \beta) \cdot \ell(\{u \in \mathcal{N} \mid \ell(u) > 0\}) \geq \beta(1 + \beta) \cdot \ell(T')$. Using these two results, we can lower bound the value of $h(T')$ by

$$\begin{aligned} h(T') &\geq \beta(1 + \beta) \cdot \ell(T') + \left\lceil \frac{4n^2}{\varepsilon} + 1 \right\rceil \cdot \frac{\Delta}{2} > \beta(1 + \beta) \cdot \ell(T') + \frac{4n^2}{\varepsilon} \cdot \frac{\Delta}{2} \\ &= \beta(1 + \beta) \cdot \ell(T') + n \cdot \max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} \geq \beta(1 + \beta) \cdot \ell(T') + \mathbb{E}[g(T'(\beta))] = h(T') , \end{aligned}$$

which is a contradiction. The strict inequality holds since $\Delta > 0$ by our assumption that $\max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} > 0$, and the second inequality holds since the submodularity and non-negativity of g guarantee that for every set $S \subseteq \mathcal{N}$

$$\begin{aligned} g(S) &\leq g(\emptyset) + \sum_{u \in S} g(u \mid \emptyset) = (1 - |S|) \cdot g(\emptyset) + \sum_{u \in S} g(u) \leq \max\left\{g(\emptyset), \sum_{u \in S} g(u)\right\} \\ &\leq \max\{g(\emptyset), |S| \cdot \max_{u \in \mathcal{N}} g(u)\} \leq n \cdot \max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} . \end{aligned}$$

The last contradiction implies that our assumption was wrong, and Algorithm 1 terminates after reaching Line 7. When this happens, since the condition of Line 5 evaluated to FALSE, for every element $u \in \mathcal{N} \setminus T$,

$$\begin{aligned} h(T) &\geq h(T) + \omega_u + \beta(1 + \beta) \cdot \ell(u) - \Delta \\ &\geq h(T) + \beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)] + \beta(1 + \beta) \cdot \ell(u) - 3\Delta/2 = h(T + u) - 3\Delta/2 , \end{aligned}$$

where the second inequality holds since we assume that all the estimates made by Algorithm 1 are good. Similarly, since the condition of Line 6 evaluated to FALSE, for every element $u \in T$,

$$\begin{aligned} h(T) &\geq h(T) - \omega_u - \beta(1 + \beta) \cdot \ell(u) - \Delta \\ &\geq h(T) - \beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)] - \beta(1 + \beta) \cdot \ell(u) - 3\Delta/2 = h(T - u) - 3\Delta/2 . \quad \square \end{aligned}$$

The last lemma shows that with high probability the final set T is an approximate local maximum with respect to h . Lemma 4.4 shows that this implies that $T(\beta)$ is a good solution in expectation. To prove Lemma 4.4, we need the following known lemma.

Lemma 4.3 (Lemma 2.2 of [7]). *Let $f: 2^X \rightarrow \mathbb{R}_{\geq 0}$ be a submodular function, and given a set $A \subseteq X$, let us denote by A_p a random subset of A where each element appears with probability $p \in [0, 1]$ (not necessarily independently). Then,*

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

Recall that \hat{S} is a subset of \mathcal{N} that maximizes the expression $(\alpha(\beta) - \varepsilon) \cdot g(\hat{S}) + (\beta - \varepsilon) \cdot \ell(\hat{S})$.

Lemma 4.4. *If the set T obeys*

$$h(T) \geq h(T + u) - 3\Delta/2 \quad \forall u \in \mathcal{N} \setminus T \quad \text{and} \quad h(T) \geq h(T - u) - 3\Delta/2 \quad \forall u \in T ,$$

then

$$\mathbb{E}[g(T(\beta)) + \ell(T(\beta))] \geq (\alpha(\beta) - 3\varepsilon/4) \cdot g(\hat{S}) + (\beta - 3\varepsilon/4) \cdot \ell(\hat{S}) .$$

Proof. By the first part of Lemma 4.2, for every element $u \in \mathcal{N} \setminus T$,

$$h(T) \geq h(T + u) - 3\Delta/2 ,$$

or equivalently $h(u \mid T) \leq 3\Delta/2$. Therefore, by the submodularity of g ,

$$\begin{aligned} \mathbb{E}[g(T(\beta) \cup (\hat{S} \setminus T))] + (1 + \beta) \cdot \ell(\hat{S} \setminus T) & \tag{1} \\ &\leq \mathbb{E}[g(T(\beta))] + \sum_{u \in \hat{S} \setminus T} \{\mathbb{E}[g(u \mid T(\beta) - u)] + (1 + \beta) \cdot \ell(u)\} \\ &= \mathbb{E}[g(T(\beta))] + \beta^{-1} \cdot \sum_{u \in \hat{S} \setminus T} h(u \mid T) \leq \mathbb{E}[g(T(\beta))] + 3\beta^{-1} |\hat{S} \setminus T| \Delta/2 . \end{aligned}$$

Similarly, since the second part of Lemma 4.2 implies that for every $u \in T$ we have $h(u \mid T - u) \geq -3\Delta/2$, the submodularity of g gives us

$$\begin{aligned} \mathbb{E}[g(T(\beta) \cap \hat{S})] - \beta(1 + \beta) \cdot \ell(T \setminus \hat{S}) & \tag{2} \\ &\leq \mathbb{E}[g(T(\beta))] - \sum_{u \in T \setminus \hat{S}} \{\beta \cdot \mathbb{E}[g(u \mid T(\beta) - u)] - \beta(1 + \beta) \cdot \ell(u)\} \\ &= \mathbb{E}[g(T(\beta))] - \sum_{u \in \hat{S} \setminus T} h(u \mid T - u) \leq \mathbb{E}[g(T(\beta))] + 3|T \setminus \hat{S}| \Delta/2 . \end{aligned}$$

Adding β times Inequality (1) to Inequality (2) now yields

$$\begin{aligned} & \beta \cdot \mathbb{E}[g(T(\beta) \cup (\hat{S} \setminus T))] + \mathbb{E}[g(T(\beta) \cap \hat{S})] + \beta(1 + \beta) \cdot [\ell(\hat{S} \setminus T) - \ell(T \setminus \hat{S})] \\ & \leq (1 + \beta) \cdot \mathbb{E}[g(T(\beta))] + 3[|\hat{S} \setminus T| + |T \setminus \hat{S}|]\Delta/2 \leq (1 + \beta) \cdot \mathbb{E}[g(T(\beta))] + 3n\Delta/2 . \end{aligned} \quad (3)$$

We can now use Lemma 4.3 to lower bound the first two terms on the leftmost side of the last inequality as follows.

$$\begin{aligned} & \beta \cdot \mathbb{E}[g(T(\beta) \cup (\hat{S} \setminus T))] + \mathbb{E}[g(T(\beta) \cap \hat{S})] \\ & \geq \beta(1 - \beta) \cdot g(\hat{S} \setminus T) + \beta^2 \cdot g(\hat{S} \cup T) + \beta \cdot g(T \cap \hat{S}) + (1 - \beta) \cdot g(\emptyset) \\ & \geq \beta(1 - \beta) \cdot [g(\hat{S} \setminus T) + g(T \cap \hat{S})] \geq \beta(1 - \beta) \cdot g(\hat{S}) , \end{aligned}$$

where the second inequality follows from the non-negativity of g , and the last inequality holds by g 's submodularity (and non-negativity). Plugging this inequality into Inequality (3) now gives

$$\beta(1 - \beta) \cdot g(\hat{S}) + \beta(1 + \beta) \cdot [\ell(\hat{S} \setminus T) - \ell(T \setminus \hat{S})] \leq (1 + \beta) \cdot \mathbb{E}[g(T(\beta))] + 3n\Delta/2 ,$$

and rearranging this inequality yields

$$\begin{aligned} \mathbb{E}[g(T(\beta)) + \ell(T(\beta))] & = \mathbb{E}[g(T(\beta))] + \beta \cdot \ell(T) \\ & \geq \frac{\beta(1 - \beta) \cdot g(\hat{S}) - 3n\Delta/2}{1 + \beta} + \beta \cdot [\ell(\hat{S} \setminus T) - \ell(T \setminus \hat{S})] + \beta \cdot \ell(T) \\ & \geq \alpha(\beta) \cdot g(\hat{S}) + \beta \cdot \ell(\hat{S}) - 3n\Delta/2 . \end{aligned}$$

To complete the proof of the lemma, it remains to show that $3n\Delta/2 \leq (3\varepsilon/4) \cdot [g(\hat{S}) + \ell(\hat{S})]$. Towards this goal, observe that

$$\max_{u \in \mathcal{N}} g(u) \leq \max_{u \in \mathcal{N}} \left\{ g(u) + \frac{1 + \beta}{2\beta^2} \cdot [\alpha(\beta) \cdot g(u) + \beta \cdot \ell(u)] \right\} = \frac{1 + \beta}{2\beta} \cdot \max_{u \in \mathcal{N}} [g(u) + \ell(u)] ,$$

where the inequality follows from the first part of Reduction 1. Using this inequality and the non-negativity of g , we can get

$$\begin{aligned} \frac{3n\Delta}{2} & = \frac{3\varepsilon}{4} \cdot \max\{g(\emptyset), \max_{u \in \mathcal{N}} g(u)\} \leq \frac{3\varepsilon(1 + \beta)}{8\beta} \cdot \max\{g(\emptyset), \max_{u \in \mathcal{N}} [g(u) + \ell(u)]\} \\ & \leq \frac{3\varepsilon(1 + \beta)}{8} \cdot [g(\hat{S}) + \ell(\hat{S})] \leq \frac{3\varepsilon}{4} \cdot [g(\hat{S}) + \ell(\hat{S})] , \end{aligned}$$

where the penultimate inequality follows from the second part of Reduction 1, and the last inequality uses the observation that the second part of Reduction 1 and the non-negativity of g imply together that $g(\hat{S}) + \ell(\hat{S})$ is non-negative. \square

We are now ready to prove Theorem 1.2.

Proof of Theorem 1.2. Recall that \hat{T} is a sample of $T(\beta)$ for the value of the set T when Algorithm 1 terminates. Lemmata 4.2 and 4.4 prove together that there exists a high probability event \mathcal{E} such that

$$\mathbb{E}[g(\hat{T}) + \ell(\hat{T}) \mid \mathcal{E}] \geq (\alpha(\beta) - 3\varepsilon/4) \cdot g(\hat{S}) + (\beta - 3\varepsilon/4) \cdot \ell(\hat{S}) .$$

The last two lines of Algorithm 1 guarantee that this algorithm always outputs a set whose value is at least $g(\hat{T}) + h(\hat{T})$ because $g(\emptyset) + \ell(\emptyset) = g(\emptyset) \geq 0$. Therefore, if we denote by \bar{T} the set outputted by Algorithm 1, then we also have

$$\mathbb{E}[g(\bar{T}) + \ell(\bar{T}) \mid \mathcal{E}] \geq (\alpha(\beta) - 3\varepsilon/4) \cdot g(\hat{S}) + (\beta - 3\varepsilon/4) \cdot \ell(\hat{S}) .$$

The last two lines of Algorithm 1 also guarantee that the output set \bar{T} of Algorithm 1 always has a non-negative value, and therefore, $\mathbb{E}[g(\bar{T}) + h(\bar{T}) \mid \bar{\mathcal{E}}] \geq 0$. Combining this inequality with the previous one using the law of total expectation yields

$$\begin{aligned} \mathbb{E}[g(\bar{T}) + h(\bar{T})] &\geq \Pr[\mathcal{E}] \cdot \mathbb{E}[g(\bar{T}) + h(\bar{T}) \mid \mathcal{E}] \\ &\geq (1 - o(1)) \cdot [(\alpha(\beta) - 3\varepsilon/4) \cdot g(\hat{S}) + (\beta - 3\varepsilon/4) \cdot \ell(\hat{S})] \\ &\geq (\alpha(\beta) - \varepsilon) \cdot g(\hat{S}) + (\beta - \varepsilon) \cdot \ell(\hat{S}) = \max_{S \subseteq \mathcal{N}} [(\alpha(\beta) - \varepsilon) \cdot g(S) + (\beta - \varepsilon) \cdot \ell(S)] , \end{aligned}$$

where the second inequality holds since $g(\bar{T}) + h(\bar{T})$ is always non-negative, the equality follows from the definition of \hat{S} , and $o(1)$ represents a term that diminishes when n goes to infinity. To justify the third inequality, note that

$$o(1) \cdot [(\alpha(\beta) - 3\varepsilon/4) \cdot g(\hat{S}) + (\beta - 3\varepsilon/4) \cdot \ell(\hat{S})] \leq o(1) \cdot (\beta - 3\varepsilon/4) \cdot [g(\hat{S}) + \ell(\hat{S})] \leq (\varepsilon/4) \cdot [g(\hat{S}) + \ell(\hat{S})] ,$$

where the last inequality here holds for large enough values of n because the second part of Reduction 1 and the non-negativity of g imply together that $g(\hat{S}) + \ell(\hat{S})$ is non-negative. \square

5 Inapproximability for Negative Linear Functions

In this section we prove Theorem 1.3, which we repeat here for convenience.

Theorem 1.3. *Given a value $\beta \geq 0$, let us define*

$$\alpha(\beta) = \min_{\substack{t \geq 1 \\ r \in (0, 1/2]}} \left\{ \frac{t + 1 + \sqrt{(t + 1)^2 - 8tr}}{4t} - \frac{r}{t + 1} \cdot \left[1 - \beta - 2 \ln \left(\frac{t + 1 - \sqrt{(t + 1)^2 - 8tr}}{2} \right) \right] \right\} .$$

*Then, for every $\varepsilon > 0$, no polynomial time algorithm can guarantee $(\alpha(\beta) + \varepsilon, \beta)$ -approximation for **RegularizedUSM** even when the linear function ℓ is guaranteed to be non-positive.*

The proof of Theorem 1.3 is based on Theorem 3.1, and therefore, we start this proof by describing an instance \mathcal{I} of **RegularizedUSM**. This instance is very similar to the instance used by Oveis Gharan and Vondrák [16] to prove their hardness result for maximizing a non-negative (not necessarily monotone) submodular function subject to a matroid constraint. Specifically, the instance \mathcal{I} has 3 parameters: an integer $n \geq 1$, a real value $t \geq 1$ and a real value $r \in (0, 1/2]$. The ground set of \mathcal{I} is $\mathcal{N} = \{a, b\} \cup \{a_i, b_i \mid i \in [n]\}$, and its objective functions are $\ell(S) = -r \cdot |S \cap \{a_i, b_i \mid i \in [n]\}|$ and

$$\begin{aligned} g(S) &= t \cdot (|S \cap \{a, b\}| \bmod 2) + \mathbf{1}[a \notin S] \cdot \mathbf{1}[S \cap \{a_i \mid i \in [n]\} \neq \emptyset] \\ &\quad + \mathbf{1}[b \notin S] \cdot \mathbf{1}[S \cap \{b_i \mid i \in [n]\} \neq \emptyset] . \end{aligned}$$

One can verify that g is indeed a non-negative submodular function. Additionally, the functions g and ℓ are both symmetric in the sense that the following types of swaps do not affect the values of these functions.

- Any swap of the identities of the elements of $\{a_i \mid i \in [n]\}$.
- Swapping the identifies of a with b plus swapping the identities of a_i and b_i for every $i \in [n]$.

Let \mathcal{G} be the group of permutations obtaining by combining swaps of these two kinds in any way.

In the next lemma, G and L are the multilinear extensions of g and ℓ , respectively, and $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{x})]$.

Lemma 5.1. *Let r and t be the values for which the maximum is obtained in the definition of $\alpha(\beta)$. Then, for any constant $\varepsilon > 0$ and a large enough n , $\max_{S \subseteq \mathcal{N}}[(\alpha(\beta) + \varepsilon) \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive and*

$$\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{S \subseteq \mathcal{N}} [\alpha(\beta) \cdot g(S) + \beta \cdot \ell(S)] .$$

Proof. Observe that the definition of \mathcal{G} guarantees that the vector $\bar{\mathbf{x}}$ obeys $\bar{x}_a = \bar{x}_b$ and $\bar{x}_{a_i} = \bar{x}_{b_j}$ for every $i, j \in [n]$. Therefore, if we define for two values $z, w \in [0, 1]$ the vector $\mathbf{y}(z, w)$ as follows

$$y_u(z, w) = \begin{cases} z & \text{if } u \in \{a, b\} , \\ w & \text{if } u \in \{a_i, b_i \mid i \in [n]\} , \end{cases}$$

then

$$\begin{aligned} \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &= \max_{z, w \in [0,1]} [G(\mathbf{y}(z, w)) + L(\mathbf{y}(z, w))] \\ &= \max_{z, w \in [0,1]} \{2(1-z)[tz + 1 - (1-w)^n] - 2rwn\} . \end{aligned}$$

Observe now that

$$(1-w)^n \geq e^{-wn}(1-w^2n) = e^{-wn} - \frac{(wn)^2 \cdot e^{-wn}}{n} = e^{-wn} - O(n^{-1}) ,$$

where the last equality holds because the maximum value of the function x^2e^{-x} for $x \geq 0$ is the constant $4e^{-2}$. Plugging this observation into the previous equation now yields

$$\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{z, w \in [0,1]} \{2(1-z)[tz + 1 - e^{-wn}] - 2rwn\} + O(n^{-1}) .$$

The derivative of $(1-z)[tz + 1 - e^{-wn}]$ with respect to z is $t + e^{-wn} - 1 - 2tz$, which is a decreasing function of z that takes the value 0 only when $z = (t + e^{-wn} - 1)/(2t)$ —note that this is a value in $[0, 1/2] \subseteq [0, 1]$. Therefore, we get from the previous inequality,

$$\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{w \in [0,1]} \{(t + 1 - e^{-wn})^2/(2t) - 2rwn\} + O(n^{-1}) . \quad (4)$$

The derivative of the argument of the max operation on the right hand side with respect to wn (i.e., when wn is treated as a single variable) is $e^{-wn}(t + 1 - e^{-wn})/t - 2r$, which is a quadratic expression in e^{-wn} whose roots are

$$e^{-wn} = \frac{-(t+1) \pm \sqrt{(t+1)^2 - 8tr}}{-2} = \frac{(t+1) \mp \sqrt{(t+1)^2 - 8tr}}{2} .$$

One can verify that, since $r \in (0, 1/2]$, the above roots are real values, and moreover, the smaller among them falls within the range $(0, 1]$, while the larger root is at least 1. This implies that the

operand of the max operation in the right hand side of Inequality (4) is maximized for the w value obeying

$$e^{-wn} = \frac{(t+1) - \sqrt{(t+1)^2 - 8tr}}{2} ,$$

and moreover, the w value obeying this inequality belongs to $[0, 1]$ when n is large enough. Thus,

$$\begin{aligned} \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &\leq \frac{(t+1 + \sqrt{(t+1)^2 - 8tr})^2}{8t} + 2r \cdot \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) + O(n^{-1}) \\ &= \frac{(t+1)(t+1 + \sqrt{(t+1)^2 - 8tr})}{4t} - r \cdot \left[1 - 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \right] + O(n^{-1}) . \end{aligned}$$

Let us now consider the right hand side of the inequality of the lemma. Since we can choose $S = \{a, b_1\}$,

$$\max_{S \subseteq \mathcal{N}} [(\alpha(\beta) + \varepsilon) \cdot g(S) + \beta \cdot \ell(S)] \geq (\alpha(\beta) + \varepsilon)(t+1) - \beta r .$$

Therefore, the inequality of the lemma holds whenever

$$\begin{aligned} (\alpha(\beta) + \varepsilon)(t+1) - \beta r &\geq \frac{(t+1)(t+1 + \sqrt{(t+1)^2 - 8tr})}{4t} \\ &\quad - r \cdot \left[1 - 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \right] + O(n^{-1}) , \end{aligned}$$

or equivalently,

$$\alpha(\beta) + \varepsilon \geq \frac{t+1 + \sqrt{(t+1)^2 - 8tr}}{4t} - \frac{r}{t+1} \cdot \left[1 - \beta - 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \right] + O(n^{-1}) ,$$

which is true by the definitions of $\alpha(\beta)$, r and t when n is large enough.

To complete the proof of the lemma, it remains to argue that $\max_{S \subseteq \mathcal{N}} [(\alpha(\beta) + \varepsilon) \cdot g(S) + \beta \cdot \ell(S)] \geq (\alpha(\beta) + \varepsilon)(t+1) - \beta r$ is strictly positive. Since $\varepsilon(t+1)$ is strictly positive, it suffices to show that $\alpha(\beta) \cdot (t+1) - \beta r$ is non-negative. By the definitions of $\alpha(\beta)$, r and t ,

$$\alpha(\beta) \cdot (t+1) - \beta r = \frac{t+1}{4t} \cdot [t+1 + \sqrt{(t+1)^2 - 8tr}] - r \cdot \left[1 - 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \right] . \quad (5)$$

The derivative of the right hand side of this equality with respect to r is

$$\begin{aligned} &\frac{t+1}{4t} \cdot \frac{-8t}{2\sqrt{(t+1)^2 - 8tr}} - 1 + 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) + 2r \cdot \frac{\frac{8t}{4\sqrt{(t+1)^2 - 8tr}}}{\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2}} \\ &= -\frac{t+1}{\sqrt{(t+1)^2 - 8tr}} - 1 + 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) \\ &\quad + \frac{8tr}{\sqrt{(t+1)^2 - 8tr} \cdot [t+1 - \sqrt{(t+1)^2 - 8tr}]} = 2 \ln \left(\frac{t+1 - \sqrt{(t+1)^2 - 8tr}}{2} \right) . \end{aligned}$$

Since this derivative is an increasing function of r , the right hand side of Equation (5) is minimized when the derivative is 0, i.e., when $t+1 - \sqrt{(t+1)^2 - 8tr} = 2$, or equivalently $r = [(t+1)^2 - (t-1)^2]/8t = 1/2$. Thus, the right hand side of Equation (5) is always at least

$$\frac{t+1}{4t} \cdot [t+1 + (t-1)] - \frac{1}{2} = \frac{t+1}{2} - \frac{1}{2} = \frac{t}{2} \geq 0 . \quad \square$$

Theorem 1.3 now follows by combining Theorem 3.1, Observation 3.2 and Lemma 5.1.

6 Results for Positive Linear Functions

In this section we study `RegularizedUSM` in the special case in which the linear function ℓ is non-negative. As explained in Section 1, following related known results, it is natural to expect a $(1/2, 1)$ -approximation for this case since $1/2$ is the best possible approximation ratio for unconstrained maximization of a non-negative submodular function. However, we show in Section 6.1 that this cannot be done (Theorem 1.6).

Let us now define $f \triangleq g + \ell$. As explained in Section 1, since f is a non-negative submodular function on its own right, one can optimize it using any algorithm for `Unconstrained Submodular Maximization` (USM). The first algorithm to obtain a tight approximation ratio of $1/2$ for USM was an algorithm called “Double Greedy” due to Buchbinder et al. [3]. Buchbinder et al. [3] described two variants of their algorithm, a deterministic variant that we term `DeterministicDG` and guarantees $1/3$ -approximation, and a randomized variant that we term `RandomizedDG` and guarantees $1/2$ -approximation. It should also be noted that the original analysis of [3] proves slightly stronger results than the above stated approximation ratios. Specifically, their analysis shows that `DeterministicDG` always outputs a set of value at least

$$\frac{f(S) + f(\emptyset) + f(\mathcal{N})}{3} \geq \frac{1}{3}g(S) + \frac{2}{3}\ell(S)$$

for any set S , where the inequality holds since the function ℓ is non-negative; which implies that `DeterministicDG` is a $(1/3, 2/3)$ -approximation algorithm. Similarly, the analysis of Buchbinder et al. [3] shows that `RandomizedDG` outputs a set whose expected value is at least

$$\frac{2f(S) + f(\emptyset) + f(\mathcal{N})}{4} \geq \frac{1}{2}g(S) + \frac{3}{4}\ell(S) ,$$

which implies that `RandomizedDG` is a $(1/2, 3/4)$ -approximation algorithm.

Theorems 1.4 and 1.5 show that `DeterministicDG` and `RandomizedDG`, respectively, guarantee (α, β) -approximation for many additional pairs of α and β . The proofs of these theorems can be found in Sections 6.2 and 6.3, respectively.

6.1 Impossibility of the Naturally Expected Approximation Guarantee

In this section we prove the following theorem. We note that the technique used in the proof of this theorem can also prove a somewhat stronger result. However, since the improvement represented by this stronger result is not very significant, we chose to state in the theorem the cleaner and more conceptually important result rather than the strongest result achievable.

Theorem 1.6. *Even when the linear function ℓ is guaranteed to be non-negative, no polynomial time algorithm can guarantee $(1/2, 1)$ -approximation for `RegularizedUSM`.*

Before getting to the proof of Theorem 1.6, we need to prove the following two technical lemmata.

Lemma 6.1. *For every constant $c \geq 1/2$, the function $x^c \cdot \left(\frac{4}{(5-x)^2} - 1 \right)$ is a non-increasing function of x for $x \in [0, 1]$.*

Proof. The derivative of the function from the lemma with respect to x is

$$\begin{aligned} cx^{c-1} \cdot \left(\frac{4}{(5-x)^2} - 1 \right) + x^c \cdot \frac{8}{(5-x)^3} &= \frac{x^{c-1}}{(5-x)^3} \cdot [4c(5-x) - c(5-x)^3 + 8x] \\ &= \frac{x^{c-1}}{(5-x)^3} \cdot [cx^3 - 15cx^2 + (71c+8)x - 105c] \\ &\leq \frac{cx^{c-1}}{(5-x)^3} \cdot [-14x^2 + 87x - 105] . \end{aligned}$$

The rightmost hand side of the last inequality is always non-positive because the roots of the quadratic function $-14x^2 + 87x - 105$ are

$$x_{1,2} = \frac{-87 \pm \sqrt{87^2 - 4 \cdot 14 \cdot 105}}{2 \cdot (-14)} = \frac{87 \mp \sqrt{1689}}{28} \geq \frac{87 - \sqrt{1689}}{28} > 1 . \quad \square$$

Lemma 6.2. *For every constant $x \geq 0$, $\sqrt[n]{1-x/n} \geq 1 - O(n^{-2})$.*

Proof. Observe that, for large enough n ,

$$\begin{aligned} \sqrt[n]{1-x/n} &= 1 - \int_{1-x/n}^1 \frac{d \sqrt[n]{y}}{dy} dy = 1 - \int_{1-x/n}^1 \frac{y^{\frac{1}{n}-1}}{n} dy \geq 1 - \int_{1-x/n}^1 \frac{1}{ny} dy = 1 - \frac{\ln y|_{1-x/n}^1}{n} \\ &= 1 + \frac{\ln(1-x/n)}{n} \geq 1 - \frac{x/n}{n(1-x/n)} = 1 - \frac{x}{n(n-x)} = 1 - O(n^{-2}) . \quad \square \end{aligned}$$

The proof of Theorem 1.6 is based on Theorem 3.1, and therefore, we need to describe an instance \mathcal{I} of **RegularizedUSM** that has an integer parameter $n \geq 2$. The ground set of the instance \mathcal{I} is $\mathcal{N} = \{a, b\} \cup \{c_i \mid i \in [n]\}$, and its objective functions are given, for every $S \subseteq \mathcal{N}$, by $\ell(S) = 1/3$ and

$$g(S) = 2 \cdot [(S \cap \{a, b\}) \bmod 2] + \mathbf{1}[\{a, b\} \cap S \neq \emptyset] \cdot \mathbf{1}[\{c_i \mid i \in [n]\} \not\subseteq S] .$$

One can verify that g is indeed a non-negative submodular function. Additionally, the functions g and ℓ are both symmetric in the sense that swapping the identities of a and b does not change the values of these functions for any set, and the same applies to any swap of the identities of the elements of $\{c_i \mid i \in [n]\}$. Let \mathcal{G} be the group of permutations obtaining by combining swaps of these two kinds in any way.

In the next lemma, G and L are the multilinear extensions of g and ℓ , respectively, and $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{x})]$.

Lemma 6.3. *For a large enough n ,*

$$\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{S \subseteq \mathcal{N}} \left[0.4998 \cdot g(S) + \frac{n-1.0003}{n-1} \cdot \ell(S) \right] ,$$

and the right hand side of the inequality is strictly positive.

Proof. Observe that the definition of \mathcal{G} guarantees that the vector $\bar{\mathbf{x}}$ obeys $\bar{\mathbf{x}}_a = \bar{\mathbf{x}}_b$ and $\bar{\mathbf{x}}_{c_i} = \bar{\mathbf{x}}_{c_j}$ for every $i, j \in [n]$. Therefore, if we define for two values $z, w \in [0, 1]$ the vector $\mathbf{y}(z, w)$ as follows

$$y_u(z, w) = \begin{cases} z & \text{if } u \in \{a, b\} , \\ w & \text{if } u \in \{c_i \mid i \in [n]\} , \end{cases}$$

then

$$\begin{aligned} \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &= \max_{z, w \in [0,1]} [G(\mathbf{y}(z, w)) + L(\mathbf{y}(z, w))] \\ &= \max_{z, w \in [0,1]} \left[4z(1-z) + (2z - z^2)(1-w^n) + \frac{nw}{3} \right]. \end{aligned}$$

Using the derivative with respect to z of the argument of the max operation in the rightmost side of the last equation, one can show that the maximum is obtained when $z = 1 - 2/(5 - w^n)$ —note that this value of z is indeed a number in the range $[1/2, 3/5] \subseteq [0, 1]$. Thus,

$$\begin{aligned} \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &\leq \max_{w \in [0,1]} \left[\frac{8(3-w^n)}{(5-w^n)^2} + \frac{(3-w^n)(7-w^n)(1-w^n)}{(5-w^n)^2} + \frac{nw}{3} \right] \\ &= \max_{w \in [0,1]} \left[\frac{(3-w^n)^2}{5-w^n} + \frac{nw}{3} \right] = \max_{w \in [0,1]} \left[1 - w^n + \frac{4}{5-w^n} + \frac{nw}{3} \right]. \end{aligned} \quad (6)$$

Consider now the argument of the max operation in the rightmost side of the last inequality. The derivative of this argument with respect to w is

$$nw^{n-1} \cdot \left(\frac{4}{(5-w^n)^2} - 1 \right) + \frac{n}{3}.$$

Let us denote the above expression by $D(w)$. Since $w \in [0, 1]$ and $\frac{4}{(5-w^n)^2} - 1 \leq \frac{4}{16} - 1 = -\frac{3}{4}$, for a large enough n ,

$$n(w^n)^{0.999} \cdot \left(\frac{4}{(5-w^n)^2} - 1 \right) + \frac{n}{3} \leq D(w) \leq nw^n \cdot \left(\frac{4}{(5-w^n)^2} - 1 \right) + \frac{n}{3}.$$

Lemma 6.1 shows that both bounds on $D(w)$ are non-increasing functions of w^n . Furthermore, one can verify that the lower bound on $D(w)$ is positive for $w^n = 0.411$ and the upper bound on $D(w)$ is negative for $w^n = 0.412$. Thus, $D(w)$ is positive for $w^n \leq 0.411$ and negative for $w^n \geq 0.412$, which implies that the argument of the max operation in the rightmost side of Inequality (6) is maximized for some value w such that $w^n \in [0.411, 0.412]$. Hence,

$$\begin{aligned} \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] &\leq \max_{w \in [\sqrt[3]{0.411}, \sqrt[3]{0.412}]} \left[1 - w^n + \frac{4}{5-w^n} + \frac{nw}{3} \right] \\ &\leq 1 - 0.411 + \frac{4}{5-0.412} + \frac{n\sqrt[3]{0.412}}{3} \leq 1.461 + \frac{n\sqrt[3]{0.412}}{3} \\ &\leq 1.461 + \frac{n\sqrt[3]{e^{-0.886}}}{3} \leq 1.461 + \frac{n\sqrt[3]{(1-0.886/n)^n/(1-0.886^2/n)}}{3} \\ &= 1.461 + \frac{n-0.886}{3\sqrt[3]{1-0.886^2/n}} \leq 1.1657 + \frac{n}{3(1-O(n^{-2}))}, \end{aligned} \quad (7)$$

where the last inequality holds by Lemma 6.2.

Since, for every value $x \in [0, 2/3]$,

$$\frac{n}{3(1-x)} - \frac{n}{3} = \frac{n[1-(1-x)]}{3(1-x)} = \frac{nx}{3(1-x)} \leq nx,$$

for large enough n , Inequality (7) implies

$$\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq 1.1657 + \frac{n}{3} + n \cdot O(n^{-2}) = 1.1657 + \frac{n}{3} + O(n^{-1}).$$

It is now time to consider the right hand side of the inequality that we need to prove. Specifically, since we can choose $S = \{a\} \cup \{c_i \mid i \in [n-1]\}$,

$$\max_{S \subseteq \mathcal{N}} \left[0.4998 \cdot g(S) + \frac{n-1.0003}{n-1} \cdot \ell(S) \right] \geq 3 \cdot 0.4998 + \frac{n-1.0003}{3} = 1.4993 + \frac{n-1}{3} > 0 .$$

Furthermore, by combining this inequality with the previous one, we get that the inequality of the lemma holds whenever

$$1.4993 + \frac{n-1}{3} \geq 1.1657 + \frac{n}{3} + O(n^{-1}) ,$$

or equivalently

$$0.3336 \geq 1/3 + O(n^{-1}) ,$$

which is true for large enough n values. \square

By combining Theorem 3.1 and Lemma 6.3, we get that, even when the linear function ℓ is non-negative, no polynomial time algorithm for **RegularizedUSM** can guarantee $(0.4998(1+\varepsilon) + (n-1.0003)(1+\varepsilon)/(n-1))$ -approximation for any $\varepsilon > 0$ and large enough n . Theorem 1.6 now follows by choosing $\varepsilon = 0.0003/n$.

6.2 Reanalysis of Deterministic Double Greedy

In this section we prove Theorem 1.4, which we repeat here for convenience. The algorithm **DeterministicDG** referred to by this theorem is given as Algorithm 2 (recall that $f \triangleq g + \ell$).

Theorem 1.4. *When ℓ is non-negative, **DeterministicDG** is an $(\alpha, 1-\alpha)$ -approximation algorithm for **RegularizedUSM** for all $\alpha \in [0, 1/3]$ at the same time (the algorithm is oblivious to the value of α).*

Algorithm 2: DeterministicDG

- 1 Denote the elements of \mathcal{N} by u_1, u_2, \dots, u_n in an arbitrary order.
 - 2 Let $X_0 \leftarrow \emptyset$ and $Y_0 \leftarrow \emptyset$.
 - 3 **for** $i = 1$ **to** n **do**
 - 4 Let $a_i \leftarrow f(u_i \mid X_{i-1})$ and $b_i \leftarrow -f(u_i \mid Y_{i-1} - u_i)$.
 - 5 **if** $a_i \geq b_i$ **then** Let $X_i \leftarrow X_{i-1} + u_i$ and $Y_i \leftarrow Y_{i-1}$.
 - 6 **else** Let $X_i \leftarrow X_{i-1}$ and $Y_i \leftarrow Y_{i-1} - u_i$.
 - 7 **return** $X_n (= Y_n)$.
-

The heart of the proof of Theorem 1.4 is the following lemma. To state this lemma, we need to define, for every integer $0 \leq i \leq n$ and set $S \subseteq \mathcal{N}$, $S^{(i)} = (S \cup X_i) \cap Y_i$.

Lemma 6.4. *For every integer $1 \leq i \leq n$, value $\alpha \in [0, 1/3]$ and set $S \subseteq \mathcal{N}$, $\alpha \cdot [f(X_i) - f(X_{i-1})] + (1-2\alpha) \cdot [f(Y_i) - f(Y_{i-1})] \geq \alpha \cdot [f(S^{(i-1)}) - f(S^{(i)})]$.*

Before we get to the proof of Lemma 6.4, let us show why it implies Theorem 1.4.

Proof of Theorem 1.4. Fix some $\alpha \in [0, 1/3]$ and set $S \subseteq \mathcal{N}$. Summing up Lemma 6.4 over all integer $1 \leq i \leq n$, we get

$$\alpha \cdot \sum_{i=1}^n [f(X_i) - f(X_{i-1})] + (1 - 2\alpha) \cdot \sum_{i=1}^n [f(Y_i) - f(Y_{i-1})] \geq \alpha \cdot \sum_{i=1}^n [f(S^{(i-1)}) - f(S^{(i)})] .$$

The sums in the last inequality are telescopic sums, and collapsing them yields

$$\alpha \cdot [f(X_n) - f(X_0)] + (1 - 2\alpha) \cdot [f(Y_n) - f(Y_0)] \geq \alpha \cdot [f(S^{(0)}) - f(S^{(n)})] .$$

One can observe that $X_n = Y_n = S^{(n)}$, $f(X_0) = g(\emptyset) \geq 0$, $f(Y_0) = g(\mathcal{N}) + \ell(\mathcal{N}) \geq \ell(S)$ and $S^{(0)} = S$. Plugging all these observations into the previous inequality yields

$$\alpha \cdot f(X_n) + (1 - 2\alpha) \cdot [f(X_n) - \ell(S)] \geq \alpha \cdot [f(S) - f(X_n)] .$$

It remains to rearrange the last inequality, and plug in $f(S) = g(S) + \ell(S)$, which implies

$$f(X_n) \geq \alpha \cdot g(S) + (1 - \alpha) \cdot \ell(S) .$$

The theorem now follows since: (i) X_n is the output set of Algorithm 2, and (ii) the last inequality holds for every $\alpha \in [0, 1/3]$ and set $S \subseteq \mathcal{N}$. \square

Let us now prove Lemma 6.4.

Proof of Lemma 6.4. Buchbinder et al. [3] showed that Algorithm 2 guarantees⁷

$$[f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})] \geq f(S^{(i-1)}) - f(S^{(i)}) . \quad (8)$$

Furthermore, we prove below that we also have the inequality

$$f(Y_i) - f(Y_{i-1}) \geq 0 . \quad (9)$$

These two inequalities imply the lemma together since the inequality guaranteed by the lemma is equal to $\alpha \cdot (8) + (1 - 3\alpha) \cdot (9)$ —note that the coefficients α and $1 - 3\alpha$ in this expression are non-negative for the range of possible values for α .

It remains to prove Inequality (9). If $Y_i = Y_{i-1}$, then Inequality (9) trivially holds as an equality. Consider now the case of $Y_i \neq Y_{i-1}$. By Lines 5 and 6 of Algorithm 2, this case happens only when $b_i > a_i$, and Y_i is set to $Y_{i-1} - u_i$ when this happens. Therefore, we get in this case

$$f(Y_i) - f(Y_{i-1}) = f(Y_{i-1} - u_i) - f(Y_{i-1}) = b_i > \frac{a_i + b_i}{2} \geq 0 ,$$

where the last inequality holds since Buchbinder et al. [3] also showed that $a_i + b_i \geq 0$. \square

6.3 Reanalysis of Randomized Double Greedy

In this section we prove Theorem 1.5, which we repeat here for convenience. The algorithm `RandomizedDG` referred to by this theorem is given as Algorithm 3 (recall that $f \triangleq g + \ell$).

Theorem 1.5. *When ℓ is non-negative, `RandomizedDG` is an $(\alpha, 1 - \alpha/2)$ -approximation algorithm for `RegularizedUSM` for all $\alpha \in [0, 1/2]$ at the same time (the algorithm is oblivious to the value of α).*

Algorithm 3: RandomizedDG

```

1 Denote the elements of  $\mathcal{N}$  by  $u_1, u_2, \dots, u_n$  in an arbitrary order.
2 Let  $X_0 \leftarrow \emptyset$  and  $Y_0 \leftarrow \emptyset$ .
3 for  $i = 1$  to  $n$  do
4   Let  $a_i \leftarrow f(u_i \mid X_{i-1})$  and  $b_i \leftarrow -f(u_i \mid Y_{i-1} - u_i)$ .
5   if  $b_i \leq 0$  then Let  $X_i \leftarrow X_{i-1} + u_i$  and  $Y_i \leftarrow Y_{i-1}$ .
6   else if  $a_i \leq 0$  then Let  $X_i \leftarrow X_{i-1}$  and  $Y_i \leftarrow Y_{i-1} - u_i$ .
7   else
8     with probability  $\frac{a_i}{a_i + b_i}$  do Let  $X_i \leftarrow X_{i-1} + u_i$  and  $Y_i \leftarrow Y_{i-1}$ .
9     otherwise Let  $X_i \leftarrow X_{i-1}$  and  $Y_i \leftarrow Y_{i-1} - u_i$ . // Occurs with prob.  $\frac{b_i}{a_i + b_i}$ .
10  return  $X_n (= Y_n)$ .
```

The heart of the proof of Theorem 1.5 is the following lemma. To state this lemma, we need to define, like in Section 6.2, $S^{(i)} = (S \cup X_i) \cap Y_i$ for every integer $0 \leq i \leq n$ and set $S \subseteq \mathcal{N}$.

Lemma 6.5. *For every integer $1 \leq i \leq n$, value $\alpha \in [0, 1/2]$ and set $S \subseteq \mathcal{N}$, $(\alpha/2) \cdot \mathbb{E}[f(X_i) - f(X_{i-1})] + (1 - 3\alpha/2) \cdot \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq \alpha \cdot \mathbb{E}[f(S^{(i-1)}) - f(S^{(i)})]$.*

Before we get to the proof of Lemma 6.5, let us show why it implies Theorem 1.5.

Proof of Theorem 1.5. Fix some $\alpha \in [0, 1/2]$ and set $S \subseteq \mathcal{N}$. Summing up Lemma 6.5 over all integer $1 \leq i \leq n$, we get

$$\frac{\alpha}{2} \sum_{i=1}^n \mathbb{E}[f(X_i) - f(X_{i-1})] + (1 - 3\alpha/2) \cdot \sum_{i=1}^n \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq \alpha \cdot \sum_{i=1}^n \mathbb{E}[f(S^{(i-1)}) - f(S^{(i)})] .$$

Due to the linearity of the expectation, the sums in the last inequality are telescopic sums. Collapsing these sums yields

$$\frac{\alpha}{2} \mathbb{E}[f(X_n) - f(X_0)] + (1 - 3\alpha/2) \cdot \mathbb{E}[f(Y_n) - f(Y_0)] \geq \alpha \cdot \mathbb{E}[f(S^{(0)}) - f(S^{(n)})] .$$

Observe now that, like in the proof of Theorem 1.4, we have $X_n = Y_n = S^{(n)}$, $f(X_0) = g(\emptyset) \geq 0$, $f(Y_0) = g(\mathcal{N}) + \ell(\mathcal{N}) \geq \ell(S)$ and $S^{(0)} = S$. Plugging all these observations into the previous inequality yields

$$\frac{\alpha}{2} \mathbb{E}[f(X_n)] + (1 - 3\alpha/2) \cdot \mathbb{E}[f(X_n) - \ell(S)] \geq \alpha \cdot \mathbb{E}[f(S) - f(X_n)] .$$

It remains to rearrange the last inequality, and plug in $f(S) = g(S) + \ell(S)$, which implies

$$\mathbb{E}[f(X_n)] \geq \alpha \cdot g(S) + (1 - \alpha/2) \cdot \ell(S) .$$

The theorem now follows since: (i) X_n is the output set of Algorithm 2, and (ii) the last inequality holds for every $\alpha \in [0, 1/2]$ and set $S \subseteq \mathcal{N}$. \square

Let us now prove Lemma 6.5.

⁷Technically, Buchbinder et al. [3] proved Inequality (8) only for the special case in which S is a set maximizing f . However, their analysis does not use this property.

Proof of Lemma 6.5. Buchbinder et al. [3] showed that Algorithm 3 guarantees⁸

$$\mathbb{E}[f(X_i) - f(X_{i-1})] + \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq 2\mathbb{E}[f(S^{(i-1)}) - f(S^{(i)})] . \quad (10)$$

Given this inequality, to prove the lemma it suffices to show that

$$(1 - 2\alpha) \cdot \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq 0$$

(because adding this inequality to $\alpha/2$ times Inequality (10) yields the inequality that we want to prove). Below we prove the stronger claim that the inequality $f(Y_i) \geq f(Y_{i-1})$ holds deterministically. One observe that this stronger claim indeed implies $(1 - 2\alpha) \cdot \mathbb{E}[f(Y_i) - f(Y_{i-1})]$ because $1 - 2\alpha$ is non-negative in the range of allowed values for α .

If $Y_i = Y_{i-1}$, then the inequality $f(Y_i) \geq f(Y_{i-1})$ trivially holds as an equality. Therefore, we assume from now on $Y_i \neq Y_{i-1}$, which implies $Y_i = Y_{i-1} - u_i$. Due to the condition in Line 5 of Algorithm 3, Y_i can be set to $Y_{i-1} - u_i$ only when $b_i > 0$, and thus,

$$f(Y_i) = f(Y_{i-1} - u_i) = f(Y_{i-1}) + b_i > f(Y_{i-1}) . \quad \square$$

A Extended Result for Non-Positive Linear Functions

In this section, using ideas from Feldman [8], we extend Lu, Yang and Gao's result [13], which we repeat here for convenience.

Theorem A.1. [Theorem 1.5 of [13]] *There exists a randomized algorithm that given a non-monotone submodular function $g: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ and a randomized modular function $\ell: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\leq 0}$, returns a set $S \subseteq \mathcal{N}$ obeying*

$$\mathbb{E}[g(S) + \ell(S)] \geq e^{-1} \cdot g(OPT) + \ell(OPT),$$

where $OPT \in \arg \max\{g(A) + \ell(A) : A \subseteq \mathcal{N}\}$. Furthermore, the algorithm performs $O(n)$ value oracle queries.

This $(e^{-1}, 1)$ -approximation result can be extended to $(\beta e^{-\beta}, \beta)$ -approximation for the same special case for any $\beta \in [0, 1]$. First we find a fractional solution for a known continuous relaxation, subject to a solvable polytope $P \subseteq [0, 1]^{\mathcal{N}}$, whose value approximates the value of the optimal integral solution. Then the fractional solution can be rounded by various known methods (that we will not repeat here). The following theorem finds a fractional solution with the desired approximation ratio.

Theorem A.2. *There exists a polynomial time algorithm for the above problem that given a value $\varepsilon \in (0, 1)$ outputs a vector $x \in P$ such that with high probability $G(x) + \ell(x) \geq (\beta e^{-\beta}) \cdot g(OPT) + \beta \cdot \ell(OPT) - O(\varepsilon) \cdot m$, where $m = \max_{u \in \mathcal{N}}\{g(u \mid \emptyset)\}$.*

Like in the work of Feldman [8], we give a non-formal proof that uses some non-formal simplifications which are known by having techniques for getting rid of them (see, e.g., [8]). The algorithm we use for the non-formal proof is given as Algorithm 4. The algorithm gets a non-negative submodular function g , a non-positive linear function ℓ , constant β , solvable polytope $P \subseteq [0, 1]^{\mathcal{N}}$ and finds a point $x \in P$ that approximately maximizes $G(x) + \ell(x)$, where G is the multilinear extension of G and the vector operation \vee represents coordinate-wise maximum.

⁸Again, the proof of [3] was technically stated only for the case in which S is a set maximizing f , but it extends without modification to any set $S \subseteq \mathcal{N}$.

For the rest of this section, we identify the linear function ℓ with a vector $\ell \in \mathbb{R}^{\mathcal{N}}$ and denote the value of the function for a vector $x \in [0, 1]^{\mathcal{N}}$ and for a set $S \subseteq \mathcal{N}$ as $\ell(x) = \ell \cdot x$ and as $\ell(S) = \ell(\mathbf{1}_S) = \sum_{u \in S} \ell_u$ respectively.

Like the original measured continuous greedy algorithm, this algorithm initialize the solution to be $\mathbf{1}_{\emptyset}$ and calculates a weight vector $w(t)$ at every time point based on the derivatives of the multilinear extension G . The algorithm differs from the original measured continuous greedy algorithm in the direction it grows the solution $y(t)$ over time. First, it finds a vector $z(t) \in P$ that maximizing a weighted combination of $w(t)$ and the linear function ℓ . Then, this vector $z(t)$ determines the direction in which the solution $y(t)$ is grown at every time $t \in [0, \beta]$.

Algorithm 4: Distorted Measured Continuous Greedy(g, ℓ, β, P)

- 1 Let $y(0) \leftarrow \mathbf{1}_{\emptyset}$.
 - 2 **foreach** time $t \in [0, \beta)$ **do**
 - 3 **foreach** $u \in \mathcal{N}$, Let $w_u(t) \leftarrow G(y(t) \vee \mathbf{1}_{\{u\}}) - G(y(t))$.
 - 4 Let $z(t)$ be the vector in P maximizing $z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell)$.
 - 5 Increase $y(t)$ at a rate of $\frac{dy(t)}{dt} = (1 - y(t)) \cdot z(t)$.
 - 6 **return** $y(\beta)$.
-

Lemma 1.18 of [2] ensures that $y(t)$ always remains within $[0, 1]^{\mathcal{N}}$. We repeat the lemma here for convenience as we are going to use it again later.

Lemma A.3. [Lemma 1.18 of [2]] For every time $t \in [0, \beta]$ and element $u \in \mathcal{N}$, $y_u \leq 1 - e^{-t} \leq 1$.

The rest of the approximation ratio analysis of Algorithm 4 is similar to the analysis of the Distorted Continuous Greedy algorithm in [8]. However, our analysis is based on the function $\Phi = e^{t-\beta} \cdot G(y(t)) + \ell(y(t))$, which is a generalization of the function used in [8]. The following lemma calculate the derivative of Φ .

Lemma A.4. $\frac{d\Phi(t)}{dt} = e^{t-\beta} \cdot G(y(t)) + z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell)$.

Proof.

$$\begin{aligned}
\frac{d\Phi(t)}{dt} &= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \frac{dG(y(t))}{dt} + \frac{d\ell(y(t))}{dt} \\
&= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \sum_{u \in \mathcal{N}} \frac{dy_u(t)}{dt} \cdot \left. \frac{\partial G(x)}{\partial x_u} \right|_{x=y(t)} + \sum_{u \in \mathcal{N}} \frac{dy_u(t)}{dt} \cdot \left. \frac{\partial \ell(x)}{\partial x_u} \right|_{x=y(t)} \\
&= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \sum_{u \in \mathcal{N}} (1 - y_u(t)) \cdot z_u(t) \cdot \left. \frac{\partial G(x)}{\partial x_u} \right|_{x=y(t)} + \sum_{u \in \mathcal{N}} \frac{dy_u(t)}{dt} \cdot \left. \frac{\partial \ell(x)}{\partial x_u} \right|_{x=y(t)}.
\end{aligned}$$

As G is multilinear, its partial derivative with respect to a single coordinate is the difference between the value of the function for two different values of this coordinate over the difference between the values. Plugging this observation into the previous inequality yields

$$\begin{aligned}
\frac{d\Phi(t)}{dt} &= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \sum_{u \in \mathcal{N}} (1 - y_u(t)) \cdot z_u(t) \cdot \left. \frac{\partial G(x)}{\partial x_u} \right|_{x=y(t)} + \sum_{u \in \mathcal{N}} \frac{dy_u(t)}{dt} \cdot \left. \frac{\partial \ell(x)}{\partial x_u} \right|_{x=y(t)} \\
&= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \sum_{u \in \mathcal{N}} (1 - y_u(t)) \cdot z_u(t) \cdot \frac{G(y(t) \vee \mathbf{1}_{\{u\}}) - G(y(t))}{1 - y_u(t)} + \sum_{u \in \mathcal{N}} \frac{dy_u(t)}{dt} \cdot \left. \frac{\partial \ell(x)}{\partial x_u} \right|_{x=y(t)} \\
&= e^{t-\beta} \cdot G(y(t)) + e^{t-\beta} \cdot \sum_{u \in \mathcal{N}} z_u(t) \cdot w_u(t) + \sum_{u \in \mathcal{N}} z_u(t) \cdot \ell_u \\
&= e^{t-\beta} \cdot G(y(t)) + z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell). \quad \square
\end{aligned}$$

In our next lemma, that lower bounds the last expression, we use Corollary 1.15 of [2] for the inequality between the Lovász and the multilinear extensions. We repeat it here for convenience.

Corollary A.5. [Corollary 1.15 of [2]] Let $f: 2^{\mathcal{N}} \rightarrow \mathbb{R}$ be a submodular function, and let \hat{f} and F be its Lovász and multilinear extensions, respectively. Then, for every $x \in [0, 1]^{\mathcal{N}}$, $F(x) \geq \hat{f}(x)$.

Lemma A.6. For every $t \in [0, \beta)$,

$$e^{t-\beta} \cdot G(y(t)) + z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell) \geq e^{-\beta} \cdot g(OPT) + \ell(OPT).$$

Proof. One possible candidate to be $z(t)$ is $\mathbf{1}_{OPT}$ since P contains the characteristic vectors of all the feasible sets. Hence,

$$\begin{aligned}
z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell) &\geq \mathbf{1}_{OPT} \cdot (e^{t-\beta} \cdot w(t) + \ell) = e^{t-\beta} \cdot \sum_{u \in OPT} w(t) + \ell(OPT) \\
&= e^{t-\beta} \cdot \sum_{u \in OPT} [G(y(t) \vee \mathbf{1}_{\{u\}}) - G(y(t))] + \ell(OPT) \\
&\geq e^{t-\beta} \cdot [G(y(t) \vee \mathbf{1}_{OPT}) - G(y(t))] + \ell(OPT) \\
&\geq e^{t-\beta} \cdot [\hat{g}(y(t) \vee \mathbf{1}_{OPT}) - G(y(t))] + \ell(OPT),
\end{aligned}$$

where the penultimate inequality holds by the submodularity of G , and the last inequality holds by Corollary A.5.

Recall that Lemma A.3 ensures that every coordinate of $y(t)$ is upper bounded by $1 - e^{-t}$. Hence,

$$\hat{g}(y(t) \vee \mathbf{1}_{OPT}) = \int_{\lambda=0}^1 g(T_\lambda(y(t) \vee \mathbf{1}_{OPT})) d\lambda \geq \int_{\lambda=1-e^{-t}}^1 g(T_\lambda(y(t) \vee \mathbf{1}_{OPT})) d\lambda = e^{-t} \cdot g(OPT).$$

Plugging the last inequality into the previous one gives

$$z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell) \geq e^{-\beta} \cdot g(OPT) - e^{t-\beta} \cdot G(y(t)) + \ell(OPT),$$

and the lemma now follows by adding $e^{t-\beta} \cdot G(y(t))$ to both sides of this inequality. \square

We are now ready to prove the approximation guarantee of Theorem A.2.

Proof of Theorem A.2. Lemmata A.4 and A.6 prove together that

$$\frac{d\Phi(t)}{dt} = e^{t-\beta} \cdot G(y(t)) + z(t) \cdot (e^{t-\beta} \cdot w(t) + \ell) \geq e^{-\beta} \cdot g(OPT) + \ell(OPT).$$

Integrating both sides of this inequality from $t = 0$ to $t = \beta$, we get

$$\Phi(\beta) - \Phi(0) \geq \beta e^{-\beta} \cdot g(OPT) + \beta \cdot \ell(OPT).$$

Recall that g is non-negative and $\ell(y(0)) = \ell(\mathbf{1}_\emptyset) = 0$, hence

$$\Phi(0) = e^{-\beta} G(y(0)) + \ell(y(0)) \geq 0.$$

Since $\Phi(\beta) = G(y(\beta)) + \ell(y(\beta))$, we finally get

$$G(y(\beta)) + \ell(y(\beta)) \geq \beta e^{-\beta} \cdot g(OPT) + \beta \cdot \ell(OPT). \quad \square$$

The formal proof of Theorem A.2 is analogous to the corresponding part in the proof of Theorem 1 in [8], and is, thus, omitted.

B Proof of Theorem 3.1

In this section we prove Theorem 3.1, which we repeat here for convenience.

Theorem 3.1. *Consider an instance (g, ℓ) of **RegularizedUSM** consisting of a non-negative submodular function $g: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ and a linear function $\ell: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$, and assume that there exists a group \mathcal{G} of permutations over \mathcal{N} such that the equalities $g(S) = g(\sigma(S))$ and $\ell(S) = \ell(\sigma(S))$ hold for all sets $S \subseteq \mathcal{N}$ and permutations $\sigma \in \mathcal{G}$. Let G and L be the multilinear extensions of g and ℓ respectively, and for every vector $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, let us denote $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\mathbf{x}]$, i.e., $\bar{\mathbf{x}}$ is the expected vector $\sigma(\mathbf{x})$ when σ is picked uniformly at random out of \mathcal{G} . For any two constants $\alpha, \beta \geq 0$, if $\max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive and*

$$\max_{\mathbf{x} \in [0, 1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] ,$$

*then no polynomial time algorithm for **RegularizedUSM** can guarantee $((1 + \varepsilon)\alpha, (1 + \varepsilon)\beta)$ -approximation for any positive constant ε . Furthermore, this inapproximability guarantee holds also when we restrict attention to instances (g', ℓ') of **RegularizedUSM** having the following additional properties.*

- *If ℓ is non-negative or non-positive, then we can assume that ℓ' also has the same property.*
- *If g is monotone, then we can assume that g' is monotone as well.*

The proof of Theorem 3.1 is based on the symmetry gap framework of Vondrák [19]. In this proof we assume $\varepsilon < 1/2$. Note that this assumption is without loss of generality since, if Theorem 3.1 applies to some constant $\varepsilon > 0$, then it trivially holds for every larger ε value. Let us now restate two central lemmata of [19].⁹

Lemma B.1 (Lemma 3.1 of [19]). *Let n be a positive integer, and let $F: [0, 1]^{\mathcal{N}} \rightarrow \mathbb{R}$ and $X = [n]$. If we define $f: 2^{\mathcal{N} \times X} \rightarrow \mathbb{R}_{\geq 0}$ as $f(S) = F(\mathbf{x}(S))$, where the vector $\mathbf{x}(S)$ is defined by $x_u(S) = \frac{1}{n}|S \cap (\{u\} \times X)|$ for every $u \in \mathcal{N}$. Then,*

1. *if $\frac{\partial F}{\partial x_u} \geq 0$ everywhere for each element $u \in \mathcal{N}$, then f is monotone,*
2. *and if the first partial derivatives of F are absolutely continuous and $\frac{\partial^2 F}{\partial x_u \partial x_v} \leq 0$ almost everywhere for all elements $u, v \in \mathcal{N}$, then f is submodular.*

⁹Some of the notation was modified in this restatement (compared to the original statement in [19]) to make it easier to use these lemmata for our purposes.

Lemma B.2 (Lemma 3.2 of [19]). *Consider a function $g: 2^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ invariant under a group of permutations \mathcal{G} on the ground set \mathcal{N} . Let $G(\mathbf{x})$ be the multilinear extension of G , define $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\mathbf{1}_{\sigma(\mathbf{x})}]$ and fix any $\varepsilon' > 0$. Then, there is $\delta > 0$ and functions $\hat{G}, \hat{H}: [0, 1]^{\mathcal{N}} \rightarrow \mathbb{R}_{\geq 0}$ (which are also symmetric with respect to \mathcal{G}), satisfying the following:*

1. *For all $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, $\hat{H}(\mathbf{x}) = \hat{G}(\bar{\mathbf{x}})$.*
2. *For all $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, $|\hat{G}(\mathbf{x}) - G(\mathbf{x})| \leq \varepsilon'$.*
3. *Whenever $\|\mathbf{x} - \bar{\mathbf{x}}\|_2 \leq \delta$, $\hat{G}(\mathbf{x}) = \hat{H}(\mathbf{x})$ and the value depends only on $\bar{\mathbf{x}}$.*
4. *The first partial derivatives of \hat{G} and \hat{H} are absolutely continuous.*
5. *If f is monotone, then, for every element $u \in \mathcal{N}$, $\frac{\partial \hat{G}}{\partial x_u} \geq 0$ and $\frac{\partial \hat{H}}{\partial x_u} \geq 0$ everywhere.*
6. *If f is submodular then, for every two elements $u, v \in \mathcal{N}$, $\frac{\partial^2 \hat{G}}{\partial x_u \partial x_v} \leq 0$ and $\frac{\partial^2 \hat{H}}{\partial x_u \partial x_v} \leq 0$ almost everywhere.*

In our use of Lemma B.2 we have to carefully choose a value for the ε' parameter of the lemma. Specifically, we choose $\varepsilon' = (\varepsilon/3) \cdot \max_{S \subseteq \mathcal{N}}[\alpha \cdot g(S) + \beta \cdot \ell(S)]$. Notice that the conditions of Theorem 3.1 guarantee that this value is (strictly) positive.

Applying Lemma B.2 with the above chosen parameter value to the function g and the group \mathcal{G} whose existence is guaranteed by the statement of Theorem 3.1, we get two functions \hat{G} and \hat{H} with the properties stated by Lemma B.2. Now, for every permutation $\sigma \in \mathcal{G}$ and integer $n \geq 1$, we can define functions $g_1^{\sigma, n}$ and $g_2^{\sigma, n}$ as follows. For every set $S \subseteq \mathcal{N} \times [n]$, let $\mathbf{y}^\sigma(S)$ be the vector defined as $y_u^\sigma(S) = \frac{1}{n} |\{i \in [n] \mid (\sigma(u), i) \in S\}|$ for every $u \in \mathcal{N}$. Then,

$$g_1^{\sigma, n}(S) = \hat{G}(\mathbf{y}^\sigma(S)) \quad \text{and} \quad g_2^{\sigma, n}(S) = \hat{H}(\mathbf{y}^\sigma(S)) .$$

Observe that, by Lemma B.1, the functions $g_1^{\sigma, n}$ and $g_2^{\sigma, n}$ are always non-negative and submodular.

We now need to invoke Lemma 3.3 of [19]. Unfortunately, the statement of this lemma is quite involved as it is designed to handle also constrained settings. Therefore, we give here only a simplified version of this lemma that suffices for our purposes.

Lemma B.3 (Special case of Lemma 3.3 of [19]). *Consider any deterministic sub-exponential time algorithm ALG that gets access to a function $g': 2^{\mathcal{N} \times [n]} \rightarrow \mathbb{R}_{\geq 0}$, and let σ be a uniformly random permutation from \mathcal{G} . Then, with probability $1 - e^{-\Omega(n)}$, ALG outputs a set of the same value when it gets either $g_1^{\sigma, n}$ or $g_2^{\sigma, n}$ as input.*

Let us now construct a family of instances of **RegularizedUSM**. For every permutation $\sigma \in \mathcal{G}$, we denote by $\mathcal{I}(\sigma, n)$ an instance of **RegularizedUSM** over the ground set $\mathcal{N} \times [n]$ whose submodular and linear objective functions are $g_1^{\sigma, n}$ and $L(\mathbf{y}^\sigma(S))$, respectively. We would like to prove that, when σ is chosen uniformly at random out of \mathcal{G} , the random instance $\mathcal{I}(\sigma, n)$ is hard in expectation for every deterministic algorithm, and therefore, by Yao's principle, it is hard also for randomized algorithms. However, before doing this, let us observe that the objective functions of $\mathcal{I}(\sigma, n)$ have all the necessary properties.

Observation B.4. *The function $g_1^{\sigma, n}$ is monotone whenever g is, and the function $L(\mathbf{y}^\sigma(S))$ is non-negative or non-positive whenever ℓ is non-negative or non-positive, respectively.*

Proof. The first part of the observation follows from Lemmata B.1 and B.2, and the second part of the observation holds since L is the multilinear extension of ℓ (which implies that it only takes values that are equal to some convex combination of values taken by ℓ). \square

We now prove, as promised, that $\mathcal{I}(\sigma, n)$ is a hard in expectation instance when the permutation σ is chosen uniformly at random out of \mathcal{G} .

Lemma B.5. *Consider any deterministic sub-exponential time algorithm ALG that gets the instance $\mathcal{I}(\sigma, n)$ for a uniformly random $\sigma \in \mathcal{G}$. Then, for a large enough n (independent of ALG), the output set T of ALG obeys*

$$\begin{aligned} \mathbb{E}[g_1^{\sigma, n}(T) + L(\mathbf{y}^\sigma(T))] &\leq \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] + (\varepsilon/2) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \\ &\leq (1 + \varepsilon/2) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] . \end{aligned}$$

Proof. The second inequality of the lemma is an immediate consequence of the inequality assumed by Theorem 3.1. Therefore, we concentrate on proving the first inequality.

Since ℓ is a linear function, for every set $S \subseteq \mathcal{N} \times [n]$,

$$\begin{aligned} L(\mathbf{y}^\sigma(S)) &= \sum_{u \in \mathcal{N}} \ell(u) \cdot y_u^\sigma(S) = \frac{1}{n} \sum_{u \in \mathcal{N}} \ell(u) \cdot |\{i \in [n] \mid (\sigma(u), i) \in S\}| \\ &= \frac{1}{n} \sum_{u \in \mathcal{N}} \ell(\sigma(u)) \cdot |\{i \in [n] \mid (\sigma(u), i) \in S\}| = \frac{1}{n} \sum_{u \in \mathcal{N}} \ell(u) \cdot |\{i \in [n] \mid (u, i) \in S\}| , \end{aligned}$$

where the penultimate equality holds since ℓ is invariant under σ , and the last equality holds since σ is a permutation. This implies that the linear objective function of $\mathcal{I}(\sigma, n)$ is independent of σ , and can be efficiently evaluated given ℓ and n alone. Therefore, when ALG is applied to $\mathcal{I}(\sigma, n)$, we can treat the linear objective function $L(\mathbf{y}^\sigma(S))$ as part of ALG , which makes ALG an algorithm over the submodular objective function of $\mathcal{I}(\sigma, n)$. Hence, by Lemma B.3, with probability $1 - e^{-\Omega(n)}$ the output set T of ALG obeys $g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)$.

When the equality $g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)$ holds, we can upper bound $g_1^{\sigma, n}(T)$ as follows.

$$\begin{aligned} g_1^{\sigma, n}(T) &= g_2^{\sigma, n}(T) = \hat{H}(y^\sigma(T)) = \hat{G}(\overline{y^\sigma(T)}) \leq G(\overline{y^\sigma(T)}) + \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \\ &\leq \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] - L(\overline{y^\sigma(T)}) + \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] , \end{aligned}$$

where the third equality and the first inequality both follow from Lemma B.2 and the value we chose for the ε' parameter of this lemma. When the equality $g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)$ does not hold, we can still observe that, since $\max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive by the assumptions of Theorem 3.1, there must exist a value d independent of n such that

$$\begin{aligned} g_1^{\sigma, n}(T) &= \hat{G}(y^\sigma(T)) \leq G(y^\sigma(T)) + \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \\ &\leq \max_{S \subseteq \mathcal{N}} g(S) + \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \leq d \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] , \end{aligned}$$

where the second inequality holds since G , as the multilinear extension of g , cannot produce values larger than the maximum value of g .

At this point we would like to use the law of total expectation to combine the two upper bounds on $g_1^{\sigma, n}(S)$ proved above. This leads to

$$\begin{aligned} \mathbb{E}[g_1^{\sigma, n}(T)] &= \Pr[g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)] \cdot \mathbb{E}[g_1^{\sigma, n}(T) \mid g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)] \\ &\quad + \Pr[g_1^{\sigma, n}(T) \neq g_2^{\sigma, n}(T)] \cdot \mathbb{E}[g_1^{\sigma, n}(T) \mid g_1^{\sigma, n}(T) \neq g_2^{\sigma, n}(T)] \\ &\leq \Pr[g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)] \cdot \left\{ \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] - \mathbb{E}[L(\overline{y^\sigma(T)}) \mid g_1^{\sigma, n}(T) = g_2^{\sigma, n}(T)] \right\} \\ &\quad + \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] + (1 - e^{-\Omega(n)}) \cdot d \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] . \end{aligned}$$

To simplify the last inequality, we make two observations. First, that for a large enough n it is guaranteed that $d(1 - e^{-\Omega(n)}) \leq \varepsilon/6$ because both d and ε are independent of n , and second, that the non-negativity of g implies that

$$\mathbb{E}[L(\overline{y^\sigma(T)}) \mid g_1^{\sigma,n}(T) \neq g_2^{\sigma,n}(T)] \leq \max_{x \in [0,1]^{\mathcal{N}}} L(\bar{x}) \leq \max_{x \in [0,1]^{\mathcal{N}}} [G(\bar{x}) + L(\bar{x})] .$$

Using these two observations and the law of total expectation (again), the previous inequality yields

$$\mathbb{E}[g_1^{\sigma,n}(T)] \leq \max_{\mathbf{x} \in [0,1]^{\mathcal{N}}} [G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] - \mathbb{E}[L(\overline{y^\sigma(T)})] + \frac{\varepsilon}{2} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] .$$

The last inequality is identical to the one that we need to prove, except that the term $\mathbb{E}[L(\overline{y^\sigma(T)})]$ in the last inequality should be replaced with $\mathbb{E}[L(y^\sigma(T))]$. However, these two terms are identical, and therefore, the lemma follows. To see that these two terms are indeed identical, observe that, since ℓ is linear and invariant under the permutations of the group \mathcal{G} ,

$$L(\overline{y^\sigma(S)}) = L(\mathbb{E}_{\sigma' \in \mathcal{G}}[\sigma'(y^\sigma(S))]) = \mathbb{E}_{\sigma' \in \mathcal{G}}[L(\sigma'(y^\sigma(S)))] = \mathbb{E}_{\sigma' \in \mathcal{G}}[L(y^\sigma(S))] = L(y^\sigma(S)) . \quad \square$$

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Consider any (possibly randomized) sub-exponential time algorithm ALG . By Lemma B.5 and Yao's theorem, there must exist an instance $\mathcal{I}(\sigma, n)$ such that ALG produces a set of expected value at most $(1 + \varepsilon/2) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)]$ given this instance.

Let us now lower bound the value of the optimal solution for $\mathcal{I}(\sigma, n)$. Let T be the set maximizing $\max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)]$. Then, the set $T' = \{(\sigma^{-1}(u), i) \mid u \in T, i \in [n]\}$ is a valid solution for $\mathcal{I}(\sigma, n)$ such that

$$\begin{aligned} \alpha \cdot g_1^{\sigma,n}(T') + \beta \cdot L(\mathbf{y}^\sigma(T')) &= \alpha \cdot \hat{G}(\mathbf{y}^\sigma(T')) + \beta \cdot L(\mathbf{y}^\sigma(T')) = \alpha \cdot \hat{G}(\mathbf{1}_T) + \beta \cdot L(\mathbf{1}_T) \\ &\geq \alpha \cdot G(\mathbf{1}_T) + \beta \cdot L(\mathbf{1}_T) - \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \\ &= \alpha \cdot g(T) + \beta \cdot \ell(T) - \frac{\varepsilon}{3} \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] \\ &= \left(1 - \frac{\varepsilon}{3}\right) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] , \end{aligned}$$

where the second equality holds by the definition of \mathbf{y}^σ , the inequality follows from Lemma B.2 and the last equality holds by the definition of T .

Assume now towards a contradiction that ALG is a $((1+\varepsilon)\alpha, (1+\varepsilon)\beta)$ -approximation algorithm. Given this assumption, the above results imply together

$$\begin{aligned} \left(1 + \frac{\varepsilon}{2}\right) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] &\geq \max_{S \subseteq \mathcal{N}} [(1 + \varepsilon)\alpha \cdot g_1^{\sigma,n}(S) + (1 + \varepsilon)\beta \cdot L(\mathbf{y}^\sigma(S))] \\ &= (1 + \varepsilon)[\alpha \cdot g_1^{\sigma,n}(T') + \beta \cdot L(\mathbf{y}^\sigma(T'))] \\ &\geq (1 + \varepsilon) \cdot \left(1 - \frac{\varepsilon}{3}\right) \cdot \max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)] . \end{aligned}$$

Since one of the conditions of Theorem 3.1 is that $\max_{S \subseteq \mathcal{N}} [\alpha \cdot g(S) + \beta \cdot \ell(S)]$ is strictly positive, the above inequality is equivalent to

$$1 + \frac{\varepsilon}{2} \geq (1 + \varepsilon) \cdot \left(1 - \frac{\varepsilon}{3}\right) = 1 + \frac{2\varepsilon}{3} - \frac{\varepsilon^2}{3} .$$

However, this inequality does not hold for any $\varepsilon \in (0, 1/2)$, and thus, our assumption that ALG is a $((1 + \varepsilon)\alpha, (1 + \varepsilon)\beta)$ -approximation algorithm leads to a contradiction. \square

References

- [1] Noga Alon and Joel H. Spencer. *The Probabilistic Method, Second Edition*. John Wiley, 2000.
- [2] Niv Buchbinder and Moran Feldman. *Submodular Functions Maximization Problems*. Chapman and Hall/CRC, 2 edition, 2018.
- [3] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time $(1/2)$ -approximation for unconstrained submodular maximization. *SIAM J. Comput.*, 44(5):1384–1402, 2015.
- [4] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.
- [5] Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discret. Appl. Math.*, 7(3):251–274, 1984.
- [6] Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC)*, pages 1107–1116. ACM, 2012.
- [7] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011.
- [8] Moran Feldman. Guess free maximization of submodular and linear sums. *Algorithmica*, 83(3):853–878, 2021.
- [9] Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 570–579. IEEE Computer Society, 2011.
- [10] Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM J. Comput.*, 43(2):514–542, 2014.
- [11] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 2634–2643. PMLR, 2019.
- [12] Ehsan Kazemi, Shervin Minaee, Moran Feldman, and Amin Karbasi. Regularized submodular maximization at scale. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 5356–5366. PMLR, 2021.
- [13] Cheng Lu, Wenguo Yang, and Suixiang Gao. Regularized non-monotone submodular maximization. *CoRR*, abs/2103.10008, 2021.
- [14] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

- [15] Sofia Maria Nikolakaki, Alina Ene, and Evimaria Terzi. An efficient framework for balancing submodularity and cost. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1256–1266. ACM, 2021.
- [16] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In Dana Randall, editor, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1098–1116. SIAM, 2011.
- [17] Xin Sun, Dachuan Xu, Yang Zhou, and Chenchen Wu. Maximizing modular plus non-monotone submodular functions. *CoRR*, abs/2203.07711, 2022.
- [18] Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Math. Oper. Res.*, 42(4):1197–1218, 2017.
- [19] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013.

תוכן עניינים

1	1.	הקדמה
4	I.	מבנה התיזה
4	2.	סקירת ידע נדרש
4	I.	פונקציות קבוצות וסימונים
5	II.	הרחבה רב לינארית
5	III.	אוב ערך
5	3.	קושי קירוב עבור פונקציות מונוטוניות
7	4.	אלגוריתם למקרה הכללי
14	5.	קושי קירוב עבור פונקציות לינאריות שליליות
17	6.	תוצאות עבור פונקציות לינאריות חיוביות
17	I.	הוכחת אי-היתכנות עבור הקירוב הטבעי הצפוי
20	II.	ניתוח מחודש עבור האלגוריתם החמדן הכפול הדטרמיניסטי
21	III.	ניתוח מחודש עבור האלגוריתם החמדן הכפול האקראי
23	7.	נספח
23	I.	תוצאה מורחבת עבור פונקציות לינאריות שליליות
26	II.	הוכחה של משפט 3.1

תקציר

באופטימיזציה קומבינטורית, במיוחד ביישומיים מעשיים, לעיתים קרובות ישנו צורך להשתמש באילוצים "רכים" אשר, להבדיל מאילוצים "קשים" בהם אסור כלל להפר את האילוץ, הפרה שלהם מותרת, אך תעלה ב"קנס". במקרים רבים, ניתן לייצג בעיות עם אילוצים כאלו כמירוב נטול אילוצים של סכום של פונקציה תת-מודולרית (מייצגת את פונקציית המטרה) ופונקציה מודולרית (מייצגת את הקנס).

לדוגמה, בעולם למידת המכונה ישנה פעולה חשובה ומרכזית בתחום הקרויה "סיכום מידע". מטרת הפעולה היא לבחור תת-קבוצה של דגימות המייצגת (במובן כלשהו) את כל הדגימות. לשם כך, לכל תת-קבוצה של דגימות נותנים ערך שייכמת את מידת הייצוג שלה. לרוב, הפונקציה הנבחרת להשמת ערכים אלו מקיימת את תכונת התת-מודולריות – כאשר מוסיפים דגימה כלשהיא, התוספת במידת הייצוג קטנה יותר כאשר קבוצת הדגימות גדלה.

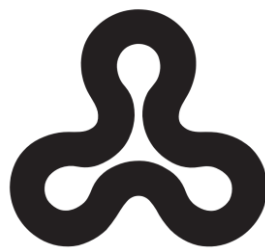
אחת השיטות בתחום למניעת התאמת יתר בין המודל לדגימות הנתונות היא להעניק לכל דגימה "מחיר", כאשר מחיר קבוצה של דגימות הינו סכום מחירי הדגימות. לכן, משימת בחירת תת-קבוצה אופטימלית של דגימות הופכת להיות מירוב ערך כולל אשר ניתן להצגה כסכום של פונקציה תת-מודולרית ופונקציה מודולרית. החלק המודולרי בהקשר זה ידוע בתור "ווסת".

בניגוד למזעור, מירוב פונקציה תת-מודולרית הינו בעיה קשה [6] (כאשר הפונקציה נתונה באמצעות אוב נדרש מספר מעריכי של גישות לאוב כדי לפתור את הבעיה, ואפילו כאשר הפונקציה נתונה בצורה מפורשת הבעיה עדיין נשארת NP-קשה). הפונקציות בהן אנו עוסקים בעבודה זו נתונות כחיבור של שתי פונקציות – מודולרית ותת-מודולרית (אי-שלילית), אי לכך הקירוב (שבצורה סטנדרטית מיוצג ע"י מקדם אחד) ירשם באמצעות שני מקדמים שייצגו את איכות הקירוב עבור כל פונקציה.

נציג עתה את התוצאות שהשגנו בעבודה זו. לשם כך, עלינו להבדיל בין מספר מקרים שונים. עבור המקרה בו מובטח כי הפונקציה המודולרית היא אי-חיובית, אנחנו מציגים שני חסמים מלמעלה על יחס הקירוב שניתן להשיג. החסם הראשון הוא עבור המקרה בו הפונקציה התת-מודולרית הינה מונוטונית. חסם זה תואם לתוצאת קירוב שהוצגה לאחרונה [12]. החסם השני הינו עבור המקרה הכללי בו לא מובטח כי הפונקציה התת-מודולרית היא מונוטונית. חסם זה משפר את תוצאת החסם הראשון עבור מקדמים של הפונקציה המודולרית הקרובים יותר ל-1, ובכך מצמצם את הפער בהשוואה לתוצאת הקירוב שניתן להשיג.

במקרה בו מובטח כי הפונקציה המודולרית היא אי-שלילית, וממילא הפונקציה התת-מודולרית במקרה בו אנו עוסקים הינה אי-שלילית, אזי ניתן להתייחס לסכום הפונקציות כפונקציה תת-מודולרית כללית (לאו דווקא מונוטונית) אי-שלילית. לכן, בתנאים הללו, נוכל ליישם בצורה ישירה את האלגוריתם "החמדן הכפול" הדטרמיניסטי ואת האלגוריתם "החמדן הכפול" האקראי ולקבל מקדם קירוב $1/3$ ו- $1/2$ בהתאמה עבור הסכום [2]. כלומר, מקדם קירוב זהה $1/3$ ו- $1/2$ בהתאמה עבור הפונקציה המודולרית ועבור הפונקציה התת-מודולרית. בעבודה זו, אנו מנתחים מחדש את תוצאת האלגוריתמים הנ"ל, ומראים כי הפעלתם בצורתם הנוכחית משיגה למעשה תוצאות טובות אף יותר עבור מקדמי הפונקציה המודולרית. יחד עם זאת, אנו גם מוכיחים כי, בניגוד למקרה המונוטוני, לא ניתן להשיג תוצאות אופטימליות בו זמנית הן לפונקציה המודולרית והן לפונקציה התת-מודולרית.

לבסוף, עבור המקרה הכללי בו לא מובטח כי הפונקציה המודולרית היא אי-שלילית או אי-חיובית, אנו מציגים את האלגוריתם הראשון עם מקדמי קירוב קבועים לא טריוויאליים עבור שתי הפונקציות (המודולרית והתת-מודולרית). האלגוריתם מבוסס על חיפוש מקומי, כאשר פונקציית המטרה בחיפוש המקומי היא פונקציית עזר שהוגדרה בהתאם למקדמי הקירוב המבוקשים.



האוניברסיטה הפתוחה
המחלקה למתמטיקה ולמדעי המחשב

**מירוב סכום של פונקצייה תת-מודולרית לא מונוטונית ושל פונקצייה
לינארית: הבנת המקרה נטול האילוצים**

מוגש על-ידי
קובי בודק

עבודה תזה זו מוגשת כחלק מהדרישות לקבלת תואר
"מוסמך למדעים" M.Sc. במדעי המחשב
באוניברסיטה הפתוחה
המחלקה למתמטיקה ומדעי המחשב

בהנחיית פרופסור מורן פלדמן ופרופסור זאב נוטוב

דצמבר 2022