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

Kobi Bodek
Department of Mathematics and Computer Science, Open University of Israel, Ra'anana, Israel

Moran Feldman
Computer Science Department, University of Haifa, Israel

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 paper 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).

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Mathematics of computing → Combinatorial optimization

Keywords and phrases Unconstrained submodular maximization, regularization, double greedy, non-oblivious local search, inapproximability

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.23

Related Version *Full Version*: `https://arxiv.org/pdf/2204.03412.pdf` [1]

Funding *Moran Feldman*: Research supported in part by Israel Science Foundation (ISF) grant number 459/20.

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 [4] 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. [17] 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. [17] 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 [7, 10, 11, 14]. 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 [12].[2]

Given the rarity of results so far for optimizing $g + \ell$ with a function g that is non-monotone, this paper 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 $g\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ and a linear function $\ell\colon 2^{\mathcal{N}} \to \mathbb{R}$ 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`.[3] Therefore, we follow previous works (starting from [7, 17]), 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.[4]

It is instructive to begin the study of `RegularizedUSM` with the special case in which the objective 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 [7] 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.** *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.*

---

[1] Technically, Sviridenko et al. [17] 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 [7].

[2] Very recently, another work of this kind appeared as a pre-print [16]. However, the main result of [16] is identical to the result of [12]. In particular, it is important to note that the result of [16] applies only to non-positive ℓ functions, like the result of [12], although this is not explicitly stated in [16].

[3] Formally, this is implied, e.g., by Theorem 1. Intuitively, the hurdle is that the combination of a positive g and a negative ℓ can lead to an optimal value that is very close to 0 compared to the values taken by the individual functions g and ℓ. When this happens, any algorithm with a positive multiplicative guarantee must output a solution that is close to optimal in terms of g and ℓ.

[4] Some 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 paper. This distinction is usually of little consequence.

We would like to draw attention to two properties of Theorem 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 is that it applies to any $\beta \geq 0$, while the algorithmic result of Feldman [7] 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 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. [17] designed two algorithms for maximizing $g + \ell$ sums, one of which was based on the continuous greedy algorithm of Călinescu et al. [3]. It is possible to modify this algorithm to be based instead on a related algorithm called "measured continuous greedy" due to [8]. In general, this does not lead to any result for maximizing $g + \ell$ sums. However, Lu et al. [12] 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. [12] using the technique discussed above. Our main algorithmic contribution is the first algorithm with a non-trivial approximation guarantee for general `RegularizedUSM`.

▶ **Theorem 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. [12] for `RegularizedUSM` with a non-positive ℓ can be extended (using the ideas of Feldman [7]) to get $(\beta e^{-\beta}, \beta)$-approximation for the same special case for any $\beta \in [0, 1]$.[5] 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 (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 holds even in the special case 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 3, we numerically draw it in Figure 1, which demonstrates that Theorem 3 closes much of the gap left with regard to `RegularizedUSM` with non-positive linear function ℓ.

[5] Technically, 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 $\beta \geq 1$.

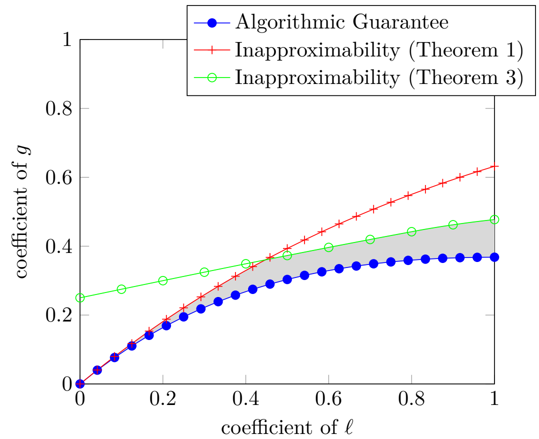

Figure 1 Graphical presentation of the existing results for RegularizedUSM with a non-positive linear function ℓ. The x and y axes represent the coefficients of ℓ and g, respectively. The algorithmic guarantee drawn is the $(\beta e^{-\beta}, \beta)$-approximation obtainable by generalizing Lu et al. [12]. The shaded area represents the gap that still exists between the best known approximation guarantee and inapproximability results.

▶ **Theorem 3.** *Given a value $\beta \geq 0$, let us define*

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

Then, for every $\varepsilon > 0$, no polynomial time algorithm guarantees $(\alpha(\beta) + \varepsilon, \beta)$-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 3 matches the state-of-the-art inapproximability result of Oveis Gharan and Vondrák [15] 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. [6], and the first algorithm to obtain this approximation ratio was the "Double Greedy" algorithm of Buchbinder et al. [2]. Specifically, Buchbinder et al. [2] 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 expected based on the guarantees of these algorithms for general `USM`.

▶ **Theorem 4.** *When ℓ is non-negative, the algorithm DeterministicDG guarantees $(\alpha, 1-\alpha)$-approximation for RegularizedUSM for all $\alpha \in [0, 1/3]$ at the same time (the algorithm is oblivious to the value of α).*

▶ **Theorem 5.** *When ℓ is non-negative, the algorithm* `RandomizedDG` *guarantees* $(\alpha, 1-\alpha/2)$*-approximation 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 $(1 - 1/e, 1)$ [17], which is natural since $1 - 1/e$ is the optimal approximation ratio for maximizing such a function g subject to a matroid constraint [13]. Thus, one might expect to get $(1/2, 1)$-approximation for `RegularizedUSM` with a non-negative ℓ. However, both Theorems 4 and 5 fail to prove such a guarantee, and we are able to show that this is not a coincidence.

▶ **Theorem 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`*.*

Paper Structure. In Section 2 we give a few formal definitions and explain the notation used throughout the paper. Then, we prove our inapproximability result for monotone `RegularizedUSM` (Theorem 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 $f : 2^{\mathcal{N}} \to \mathbb{R}$, 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 $f(u \mid S) \triangleq f(S \cup \{u\}) - f(S)$. A set function $f : 2^{\mathcal{N}} \to \mathbb{R}$ is called *submodular* if it satisfies the intuitive property of diminishing returns. More formally, f is submodular if $f(u \mid S) \geq f(u \mid T)$ 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 $S \subseteq \mathcal{N}$.[6] 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.

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 [3]). Formally, given a set function $f : 2^{\mathcal{N}} \to \mathbb{R}$, its multilinear extension is the function $F : [0, 1]^{\mathcal{N}} \to \mathbb{R}$ defined, for every vector $\mathbf{x} \in [0, 1]^{\mathcal{N}}$, by $F(\mathbf{x}) = \mathbb{E}[f(\mathtt{R}(\mathbf{x}))]$, where $\mathtt{R}(\mathbf{x})$ 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).

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

Value Oracle. As is standard in the submodular optimization literature, we assume in this paper 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 [5].

3 Inapproximability for Monotone Functions

In this section we show an inapproximability for monotone `RegularizedUSM` (Theorem 1). All our inapproximability results in this paper are proved using Theorem 7. Since the proof of this theorem is a relatively straightforward adaptation of the symmetry gap framework of Vondrák [18], we defer it to the full version of this paper [1].

▶ **Theorem 7.** *Consider an instance (g, ℓ) of* `RegularizedUSM` *consisting of a non-negative submodular function $g\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ and a linear function $\ell\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$, 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.*

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 7.

▶ **Observation 8.** *If ℓ is non-positive and $\alpha > 0$, then one can replace the term "$((1+\varepsilon)\alpha, (1+\varepsilon)\beta)$-approximation" in Theorem 7 with the term "$(\alpha+\varepsilon, \beta)$-approximation".*

Proof. Theorem 7 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. ◀

To prove Theorem 1 using Theorem 7, 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

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

▶ **Lemma 9.** *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 7 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 7. 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,

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

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,

$$\lim_{n\to\infty}(n - 1)[1 - r^{1/(n-1)}] = \lim_{y\to0}\frac{1 - r^y}{y} = \lim_{y\to0}\frac{-r^y \ln r}{1} = -\ln r \enspace ,$$

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 7 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,

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

Theorem 1, which we repeat here for convenience, now follows by combining Theorem 7, Observation 8 and Lemma 9 since g is a non-negative monotone submodular function and ℓ is a non-positive linear function.

▶ **Theorem 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 2, which we repeat here for convenience.

▶ **Theorem 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. [6] and Filmus and Ward [9]. The auxiliary function used by our algorithm is a function $h\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ defined as follows. For every set $S \subseteq \mathcal{N}$,

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

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 2. Setting a lower value for ε decreases Δ, which increases our algorithm's time complexity, 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. Due to space constraints, the proof of this reduction and some other proofs from this section are deferred to the full version of this paper [1]. In a nutshell, the first part of Reduction 10 is proved by arguing that elements violating this part cannot belong to the set we need to compete with, and thus, can be ignored; and the second part of the reduction is proved by showing that a simple algorithm outputting the best solution of size at most 1 achieves the guarantee of Theorem 2 when this part of the reduction is violated.

▶ **Reduction 10.** *While proving Theorem 2, we may assume that every element $u \in \mathcal{N}$ obeys*

$$\alpha(\beta) \cdot g(u) + \beta \cdot \ell(u) \geq 0 \qquad \textit{and} \qquad \max\{g(u) + \ell(u), g(\varnothing)\} \leq \beta \cdot [g(\hat{S}) + \ell(\hat{S})] \enspace .$$

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 10 was applied, that n is large enough and that $\max\{g(\varnothing), \max_{u \in \mathcal{N}} g(u)\} > 0$.[7] The algorithm maintains a solution T, which it updates in

[7] Let 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 2. Additionally, if $\max\{g(\varnothing), \max_{u \in \mathcal{N}} g(u)\} = 0$, then the submodularity of g guarantees that g is the zero function, which means that we can get the guarantee of Theorem 2 by outputting the set $\{u \in \mathcal{N} \mid \ell(u) > 0\}$.

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 6 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 8 looks for an element $u \in T$ which will increase $h(T)$ by Δ if removed from T (again, based on the estimate ω_T). If such an element u is found, then the algorithm removes it from T and continues to the next iteration. However, if both Lines 6 and 8 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 $\varnothing$). 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.

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

1 Let $\Delta \leftarrow \frac{\varepsilon}{2n} \cdot \max\{g(\varnothing), \max_{u \in \mathcal{N}} g(u)\}$ and $T \leftarrow \{u \in \mathcal{N} \mid \ell(u) > 0\}$.
2 **for** $i = 1$ **to** $\lceil 4n^2/\varepsilon \rceil + 1$ **do**
3   **for** *every* $u \in \mathcal{N}$ **do**
4   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 $k = \lceil 128 n^4 \varepsilon^{-2} \beta^2 \cdot \ln(10n^4/\varepsilon) \rceil$ 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**
6   Update $T \leftarrow T + u$.
7   **else if** *there exists* $u \in T$ *such that* $\omega_u + \beta(1+\beta) \cdot \ell(u) \leq -\Delta$ **then**
8   Update $T \leftarrow T - u$.
9   **else** Exit the "for" loop.
10 Let $\hat{T}$ be a sample of $T(\beta)$.
11 **if** $g(\hat{T}) + \ell(\hat{T}) \geq 0$ **then return** $\hat{T}$.
12 **else return** $\varnothing$.

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*. The following lemma is proved by showing that every estimate strongly concentrates due to the independent samples used to compute it, and then lower bounding the probability that all the estimates are good via the union bound.

▶ **Lemma 11.** *With high probability (a probability approaching* 1 *when* n *tends to infinity), all the estimates made by Algorithm 1 are good.*

Using the previous lemma, we can now prove that, with high probability, Algorithm 1 terminates with T being an approximate local maximum. Specifically, in the proof of the next lemma we show that if all the estimates are good (which is a high probability event by the previous lemma), then Algorithm 1 must encounter an approximate local maximum because otherwise the value of its solution grows to an impossibly high value.

▶ **Lemma 12.** *With high probability, when Algorithm 1 terminates we have*

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

The last lemma shows that with high probability the final set T is an approximate local maximum with respect to h. Lemma 14 shows that this implies that $T(\beta)$ is a good solution in expectation. To prove Lemma 14, we need the following known lemma.

▶ **Lemma 13** (Lemma 2.2 of [6]). *Let $f\colon 2^X \to \mathbb{R}_{\geq 0}$ 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, $\mathbb{E}[f(A_p)] \geq (1-p) \cdot f(\varnothing) + p \cdot f(A)$.*

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

▶ **Lemma 14.** *If the set T obeys*

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

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 12, 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,

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

where the equality holds because the law of total expectation implies that, for $u \notin T$,

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

Similarly, since the second part of Lemma 12 implies that for every $u \in T$ we have $h(u \mid T - u) \geq -3\Delta/2$, the submodularity of g gives us

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

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

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

We can now use Lemma 13 to lower bound the first two terms on the leftmost side of the last inequality as follows.

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

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 10. Using this inequality and the non-negativity of g, we can get

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

where the penultimate inequality follows from the second part of Reduction 10, and the last inequality uses the observation that the second part of Reduction 10 and the non-negativity of g imply together that $g(\hat{S})+\ell(\hat{S})$ is non-negative. ◀

We are now ready to prove Theorem 2.

Proof of Theorem 2. Recall that $\hat{T}$ is a sample of $T(\beta)$ for the value of the set T when Algorithm 1 terminates. Lemmata 12 and 14 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 $g(\varnothing)+\ell(\varnothing)=g(\varnothing)\geq 0$. 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

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

where the last inequality here holds for large enough values of n because the second part of Reduction 10 and the non-negativity of g imply together that $g(\hat{S}) + \ell(\hat{S})$ is non-negative. ◀

5 Inapproximability for Negative Linear Functions

In this section we prove Theorem 3, which we repeat here for convenience.

▶ **Theorem 3.** *Given a value $\beta \geq 0$, let us define*

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

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

The proof of Theorem 3 is based on Theorem 7, 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 [15] 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

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

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 idenities of a_i and b_i for every $i \in [n]$.

Let $\mathcal{G}$ be the group of permutations obtained 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 $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{x})]$. Theorem 3 follows by combining this lemma with Theorem 7 and Observation 8. The (quite technical) proof of Lemma 15 can be found in the full version of this paper [1].

▶ **Lemma 15.** *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 $\max_{\mathbf{x} \in [0,1]^{\mathcal{N}}}[G(\bar{\mathbf{x}}) + L(\bar{\mathbf{x}})] \leq \max_{S \subseteq \mathcal{N}}[\alpha(\beta) \cdot g(S) + \beta \cdot \ell(S)]$.*

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 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. [2]. Buchbinder et al. [2] 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 [2] proves slightly stronger results than the above stated approximation ratios. Specifically, their analysis shows that DeterministicDG always outputs a set of value at least $\frac{1}{3}[f(S) + f(\varnothing) + f(\mathcal{N})] \geq \frac{1}{3}g(S) + \frac{2}{3}\ell(S)$ for any set S, where the inequality holds because ℓ is non-negative; which implies that DeterministicDG is a $(1/3, 2/3)$-approximation algorithm. Similarly, the analysis of Buchbinder et al. [2] shows that RandomizedDG outputs a set whose expected value is at least $\frac{1}{4}[2f(S) + f(\varnothing) + f(\mathcal{N})] \geq \frac{1}{2}g(S) + \frac{3}{4}\ell(S)$, which implies that RandomizedDG is a $(1/2, 3/4)$-approximation algorithm.

Theorems 4 and 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 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.*

The proof of Theorem 6 is based on Theorem 7, 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

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

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 $\bar{\mathbf{x}} = \mathbb{E}_{\sigma \in \mathcal{G}}[\sigma(\mathbf{x})]$. By combining this lemma with Theorem 7, 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 6 follows by choosing $\varepsilon = 0.0003/n$. Due to space constraints, (the quite technical) proof of Lemma 16 has been deferred to the full version of this paper [1].

▶ **Lemma 16.** *For a large enough n,*

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

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

6.2 Reanalysis of Deterministic Double Greedy

In this section we prove Theorem 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 4.** *When ℓ is non-negative, the algorithm `DeterministicDG` guarantees $(\alpha, 1-\alpha)$-approximation 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`.

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

The heart of the proof of Theorem 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 17.** *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 17, let us show why it implies Theorem 4.

Proof of Theorem 4. Fix some $\alpha \in [0,1/3]$ and set $S \subseteq \mathcal{N}$. Summing up Lemma 17 over all integer $1 \leq i \leq n$, we get

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

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

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

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

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

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 $S \subseteq \mathcal{N}$. ◀

Let us now prove Lemma 17.

Proof of Lemma 17. Buchbinder et al. [2] showed that Algorithm 2 guarantees[8]

$$[f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})] \geq f(S^{(i-1)}) - f(S^{(i)}) \ . \tag{4}$$

Furthermore, we prove below that we also have the inequality

$$f(Y_i) - f(Y_{i-1}) \geq 0 \ . \tag{5}$$

These two inequalities imply the lemma together since the inequality guaranteed by the lemma is equal to $\alpha \cdot (4) + (1 - 3\alpha) \cdot (5)$ – 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 (5). If $Y_i = Y_{i-1}$, then Inequality (5) 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. [2] also showed that $a_i + b_i \geq 0$. ◀

6.3 Reanalysis of Randomized Double Greedy

In this section we prove Theorem 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 5.** *When ℓ is non-negative, the algorithm* `RandomizedDG` *guarantees $(\alpha, 1 - \alpha/2)$-approximation 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`.

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

The heart of the proof of Theorem 5 is the next 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 18.** *For every integer $1 \leq i \leq n$, value $\alpha \in [0, 1/2]$ and set $S \subseteq \mathcal{N}$, $(\alpha/2) \cdot \mathbb{E}[f(X_i) - f(X_{i-1})] + (1 - 3\alpha/2) \cdot \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq \alpha \cdot \mathbb{E}[f(S^{(i-1)}) - f(S^{(i)})]$.*

[8] Technically, Buchbinder et al. [2] proved Inequality (4) only for the special case in which S is a set maximizing f. However, their analysis does not use this property.

Before we get the to the proof of Lemma 18, let us show why it implies Theorem 5.

Proof of Theorem 5. Fix some $\alpha \in [0, 1/2]$ and set $S \subseteq \mathcal{N}$. Summing up Lemma 18 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 4, we have $X_n = Y_n = S^{(n)}$, $f(X_0) = g(\varnothing) \geq 0$, $f(Y_0) = g(\mathcal{N}) + \ell(\mathcal{N}) \geq \ell(S)$ 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 $S \subseteq \mathcal{N}$. ◀

Let us now prove Lemma 18.

Proof of Lemma 18. Buchbinder et al. [2] showed that Algorithm 3 guarantees[9]

$$\mathbb{E}[f(X_i) - f(X_{i-1})] + \mathbb{E}[f(Y_i) - f(Y_{i-1})] \geq 2\mathbb{E}[f(S^{(i-1)}) - f(S^{(i)})] \ . \tag{6}$$

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 (6) 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, $f(Y_i) = f(Y_{i-1} - u_i) = f(Y_{i-1}) + b_i > f(Y_{i-1})$. ◀

References

1 Kobi Bodek and Moran Feldman. Maximizing sums of non-monotone submodular and linear functions: Understanding the unconstrained case. *CoRR*, abs/2204.03412, 2022. doi:10.48550/arXiv.2204.03412.

2 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. doi:10.1137/130929205.

3 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. doi:10.1137/080733991.

[9] Again, the proof of [2] was technically stated only for the case in which S is a set maximizing f, but it extends without modification to any set $S \subseteq \mathcal{N}$.

4 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. doi:10.1016/0166-218X(84)90003-9.

5 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. doi:10.1145/2213977.2214076.

6 Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011. doi:10.1137/090779346.

7 Moran Feldman. Guess free maximization of submodular and linear sums. *Algorithmica*, 83(3):853–878, 2021. doi:10.1007/s00453-020-00757-9.

8 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. doi:10.1109/FOCS.2011.46.

9 Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM J. Comput.*, 43(2):514–542, 2014. doi:10.1137/130920277.

10 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. URL: http://proceedings.mlr.press/v97/harshaw19a.html.

11 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. URL: http://proceedings.mlr.press/v139/kazemi21a.html.

12 Cheng Lu, Wenguo Yang, and Suixiang Gao. Regularized non-monotone submodular maximization. *CoRR*, abs/2103.10008, 2021. arXiv:2103.10008.

13 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.

14 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. doi:10.1145/3447548.3467367.

15 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. doi:10.1137/1.9781611973082.83.

16 Xin Sun, Dachuan Xu, Yang Zhou, and Chenchen Wu. Maximizing modular plus non-monotone submodular functions. *CoRR*, abs/2203.07711, 2022. arXiv:2203.07711.

17 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. doi:10.1287/moor.2016.0842.

18 Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013. doi:10.1137/110832318.