Bridging the Gap Between General and Down-Closed Convex Sets in Submodular Maximization

Loay Mualem[1] , **Murad Tukan**[2] and **Moran Feldman**[1]
[1]University of Haifa
[2]DataHeroes
loaymua@gmail.com, murad@dataheroes.ai, moranfe@cs.haifa.ac.il

Abstract

Optimization of DR-submodular functions has experienced a notable surge in significance in recent times, marking a pivotal development within the domain of non-convex optimization. Motivated by real-world scenarios, some recent works have delved into the maximization of non-monotone DR-submodular functions over general (not necessarily down-closed) convex set constraints. Up to this point, these works have all used the minimum L-infinity norm of any feasible solution as a parameter. Unfortunately, a recent hardness result due to Mualem and Feldman shows that this approach cannot yield a smooth interpolation between down-closed and non-down-closed constraints. In this work, we suggest novel offline and online algorithms that provably provide such an interpolation based on a natural decomposition of the convex body constraint into two distinct convex bodies: a down-closed convex body and a general convex body. We also empirically demonstrate the superiority of our proposed algorithms across three offline and two online applications.

1 Introduction

Optimization of continuous DR-submodular functions (and the strongly related discrete submodular set functions) has experienced a notable surge in significance in recent times, marking a pivotal development within the domain of non-convex optimization. The tools developed in this context adaptively tackle challenges related to real-world applications at the forefront of various fields such as data summarization [Mualem *et al.*, 2023; Hassani *et al.*, 2017; Bian *et al.*, 2019; Mitra *et al.*, 2021; Soma and Yoshida, 2017], robotics [Shi *et al.*, 2021; Tukan *et al.*, 2023], and human brain mapping [Salehi *et al.*, 2017], among many others. Most of the works in the literature focused either on DR-submodular optimization for monotone objective functions, or subject to a down-closed convex set constraint.[1] However, real-world problems are often naturally captured as optimization of a non-monotone DR-submodular function over a constraint convex set that is not down-closed. For example, imagine an online shopping platform optimizing its product recommendations within strict interface size constraints. The challenge faced by the store involves designing concise summaries that respect upper and lower bounds on product inclusion (these bounds are imposed by the user interface, and they form a non-down-closed constraint), and are good with respect to an objective function balancing between diversity and relevance in the displayed recommendations (which naturally yields a non-monotone objective).

Motivated by scenarios similar to the one given above, and optimization under fairness constraints [El Halabi *et al.*, 2023; Halabi *et al.*, 2023; Yuan and Tang, 2023], some recent works have delved into the maximization of non-monotone DR-submodular functions over general (not necessarily down-closed) convex set constraints. Unfortunately, in general, no constant approximation can be obtained for this problem in sub-exponential time due to a hardness result by Vondrák [2013]. However, Durr et al. [2021] observed that Vondrák's proof of the hardness result was based on a class of instances whose convex sets $\mathcal{K}$ include no point whose ℓ_∞-norm is less than 1.[2] In light of this observation, Durr et al. [2021] considered a parametrization of the problem based on the minimum ℓ_∞ norm of any vector in $\mathcal{K}$ (this minimum is usually denoted by m), and were able to provide a sub-exponential time $\frac{1}{3\sqrt{3}}(1-m)$-approximation for the problem of optimizing a non-monotone DR-submodular function subject to a general convex set $\mathcal{K}$.

The work Durr et al. [2021] has inspired a new line of research. Thắng & Srivastav [2021] showed how to obtain a similar result in an online (regret minimization) setting, and Du et al. [2022] improved the approximation ratio in the of-

[*]A full version of this paper, including all details and proofs omitted here, is available in [Mualem *et al.*, 2024].

[1]A set $\mathcal{K}$ is *down-closed* with respect to a domain $\mathcal{X}$ if, for every two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\mathbf{x} \in \mathcal{K}$ whenever $\mathbf{y} \in \mathcal{K}$ and $\mathbf{y}$ coordinate-wise dominates $\mathbf{x}$. As is standard, we usually omit the domain when talking about down-closed sets, and implicitly assume it to be the domain of the objective function.

[2]For simplicity, we implicitly assume that the domain of the objective function and the convex set constraint is $[0,1]^n$. This assumption is without loss of generality (see Section 2).

fline setting to $\frac{1}{4}(1-m)$ (while still using sub-exponential time). More recently, the first polynomial time algorithm for the problem, guaranteeing the same approximation ratio as [Du *et al.*, 2022], has been provided by Du [2022]; and subsequently, Mualem & Feldman [2023] obtained the same result in the online setting. In their work, Mualem & Feldman also proved a hardness result, showing that $\frac{1}{4}(1-m)$-approximation is the best approximation ratio that can be obtained in sub-exponential time, which shows that the last algorithms are optimal, and settles the approximability of the problem with respect to this parametrization.

Our contribution. Every down-closed convex body has $m = 0$. The reverse is not true, but one could hope that convex bodies having $m = 0$ admit as good approximation as down-closed convex bodies. Unfortunately, the hardness result of Mualem & Feldman disproves this hope since the state-of-the-art approximation ratio for down-closed bodies is 0.401 [Buchbinder and Feldman, 2023]. This prompts the question of whether there is a different way to look at the problem (beyond parametrization by m) that will provide a smooth interpolation between the approximability obtainable for down-closed and general convex bodies. In this work, we suggest such an interpolation based on a decomposition of the convex body constraint into two distinct convex bodies: a down-closed convex body and a general convex body. Our key results based on this decomposition are as follows.

- We provide a novel polynomial time (offline) algorithm for maximizing DR-submodular functions over convex sets given as a composition of a down-closed convex body $\mathcal{K}_{\mathrm{D}}$ and a general convex body $\mathcal{K}_{\mathrm{N}}$. Our algorithm always recovers at least the $\frac{1}{4}(1-m)$-approximation of Du [2022]. The approximation guarantee smoothly improves when a significant fraction of the value of the optimal solution belongs to the down-closed part $\mathcal{K}_{\mathrm{D}}$ of the decomposition. In particular, when the convex body happens to be entirely down-closed, our algorithm guarantees $e^{-1} \approx 0.367$-approximation, which recovers the approximation ratio for down-closed convex bodies obtained by the Measured Continuous Greedy technique [Bian *et al.*, 2017a; Feldman *et al.*, 2011].[3]
- We provide a novel online (regret minimization) algorithm for the same problem that replicates the guarantees of our offline algorithm, up to a regret term that is proportional to the square root of the number of time steps.
- We demonstrate that a decomposition of the kind that we use can be naturally obtained for various machine-learning applications, and use this fact to empirically demonstrate the superiority of our proposed algorithms compared to existing methods across various offline and online applications, namely, offline and online revenue maximization, quadratic programming and location summarization. In the full version of this paper [Mualem *et al.*, 2024], we also provide theoretical implications for fairness settings.

1.1 Related Work

DR-submodular maximization has recently emerged as a key tool in numerous applications within the realms of machine learning and statistics. This surge in relevance has prompted a growing number of studies in this field. In what follows, we provide a brief overview of the main results in this field.

Offline DR-submodular optimization. The study of maximization of DR-submodular functions over down-closed convex sets was initiated by Bian et al. [2017a]. Their work showcased the effectiveness of a modified Frank-Wolfe algorithm, grounded in the greedy method introduced by [Calinescu *et al.*, 2011] in the context of set functions, ensuring $(1-1/e)$-approximation for the problem when the objective function is guaranteed to be monotone (this is optimal due to [Nemhauser and Wolsey, 1978]). Hassani et al. [2017] subsequently identified a limitation in the algorithm proposed by [Bian *et al.*, 2017a], revealing its lack of robustness in stochastic settings where only an unbiased estimator of the gradient is available. To mitigate this limitation, Hassani et al. [2017] demonstrated that gradient methods exhibit robustness in such scenarios, achieving $1/2$-approximation.

When the DR-submodular function is not guaranteed to be monotone, the approximation task becomes notably more challenging. In separate works, Bian et al. [2019] and Niazadeh et al. [2020] introduced distinct algorithms, both ensuring $1/2$-approximation for the maximization of non-monotone DR-submodular functions over a hypercube constraint, which is the best possible [Feige *et al.*, 2011]. It is noteworthy that (one version of) the algorithm proposed by [Niazadeh *et al.*, 2020] applies also to (non-DR) submodular functions. As mentioned above, the state-of-the-art approximation for maximizing non-monotone DR-submdoular functions subject to a down-closed convex body constraint is 0.401 [Buchbinder and Feldman, 2023].

Recently, Pedramfar et al. [2023] introduced a unified approach for maximization of continuous DR-submodular functions that encompasses a range of settings and oracle access types, while Mualem & Feldman [2022] suggested a parameter termed *monotonicity-ratio* allowing for a smooth interpolation between e^{-1}-approximation for non-monotone objectives and $(1-1/e)$-approximation for monotone objectives.

Online (regret minimization) DR-submodular optimization. Chen et al. [2018] were the first to address online maximization of monotone DR-submodular functions over a convex set (for monotone objective functions the distinction between down-closed and general convex sets is irrelevant). They introduced two algorithms: one ensuring $(1-1/e)$-approximation with approximately $O(\sqrt{T})$-regret, and another algorithm that is resilient to stochastic settings, but guarantees only $1/2$-approximation up to the same regret. Later, Chen et al. [2019] proposed an algorithm that combines $(1-1/e)$-approximation with roughly $O(\sqrt{T})$-regret and robustness, and Zhang et al. [2019] demonstrated how to reduce

[3]As mentioned above, the state-of-the-art approximation ratio for down-closed convex body constraints is 0.401. However, all current algorithms for such convex bodies with ratios better than e^{-1} are based on the Measured Continuous Greedy technique (with additional, often impractical, components). To keep our algorithms simple and our message clear, we only aim to recover e^{-1}-approximation for down-closed convex bodies.

the number of gradient calculations per time step to one at the expense of increasing the regret to roughly $O(T^{4/5})$. The last reduction is particularly relevant for bandit versions of the problem (we refer the reader to [Pedramfar *et al.*, 2023] for a detailed overview of such versions).

For online optimization of DR-submodular functions that are not guaranteed to be monotone, Thắng & Srivastav [2021] introduced three algorithms. One of these algorithms is applicable to general convex set constraints, and was later improved over by Mualem and Feldman [2023], as discussed above. Another was designed for maximization over the entire hypercube, achieving $1/2$-approximation with approximately $O(\sqrt{T})$-regret. The last algorithm of [Thắng and Srivastav, 2021] addresses down-closed convex set constraints, and attains e^{-1}-approximation with roughly $O(T^{2/3})$-regret.

1.2 Paper Organization

In Section 2, we formally describe the problem we consider. Then, in Section 3, we discuss the technique underlying our results. Our offline and online algorithms, which are based on this technique, can be found in Sections 4 and 5, respectively. Finally, Section 6 compares the empirical performance of our algorithms on multiple machine learning applications with the performance of previously suggested algorithms from the literature.

2 Preliminaries

In this section, we formally present the problem we consider in this paper and the notation that we use. Let us begin with the definition of DR-submodular functions, which are continuous analogs of submodular set functions first defined by Bian et al. [2017b]. Formally, given a domain $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$, where $\mathcal{X}_i$ is a closed range in $\mathbb{R}$ for every $i \in [n]$, a function $F\colon \mathcal{X} \to \mathbb{R}$ is called *DR-submodular* if the inequality

$$F(\mathbf{a} + k\mathbf{e}_i) - F(\mathbf{a}) \geq F(\mathbf{b} + k\mathbf{e}_i) - F(\mathbf{b})$$

holds for every two vectors $\mathbf{a}, \mathbf{b} \in \mathcal{X}$, positive value k and coordinate $i \in [n]$ obeying $\mathbf{a} \leq \mathbf{b}$ and $\mathbf{b} + k\mathbf{e}_i \in \mathcal{X}$ (here and throughout the paper, $\mathbf{e}_i$ denotes the standard i-th basis vector, and comparison between two vectors should be understood to hold coordinate-wise). Bian et al. [2017b] observed that for continuously differentiable functions F, the above definition of DR-submodulrity is equivalent to

$$\nabla F(\mathbf{x}) \leq \nabla F(\mathbf{y}) \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{x} \geq \mathbf{y} \enspace ,$$

and for twice differentiable functions F, it is equivalent to the Hessian being non-positive at every vector $\mathbf{x} \in \mathcal{X}$.

In this work, we study the problem of maximizing a non-negative DR-submodular function $F\colon \mathcal{X} \to \mathbb{R}_{\geq 0}$ subject to a convex body $\mathcal{K} \subseteq \mathcal{X}$ constraint. We are interested in the approximation guarantee that can be obtained for this problem based on a particular decomposition of $\mathcal{K}$ into two other convex bodies: a convex body $\mathcal{K}_{\mathrm{D}}$ that is down-closed with respect to $\mathcal{X}$ and a (not necessary down-closed) convex body $\mathcal{K}_{\mathrm{N}}$. Formally, by saying that $\mathcal{K}_{\mathrm{D}}$ and $\mathcal{K}_{\mathrm{N}}$ are a *decomposition* of $\mathcal{K}$, we mean that $\mathcal{K} = (\mathcal{K}_{\mathrm{N}} + \mathcal{K}_{\mathrm{D}}) \cap \mathcal{X}$, where $\mathcal{K}_{\mathrm{N}} + \mathcal{K}_{\mathrm{D}} \triangleq \{\mathbf{y} + \mathbf{z} \mid \mathbf{y} \in \mathcal{K}_{\mathrm{N}}, \mathbf{z} \in \mathcal{K}_{\mathrm{D}}\}$.

For simplicity, we assume (throughout the paper) that the domain $\mathcal{X}$ of our objective functions is $[0,1]^n$. This assumption is without loss of generality since there is a natural linear mapping from $\mathcal{X}$ to $[0,1]^n$ preserving the above discussed properties of F and $\mathcal{K}$. Additionally, as is standard in the field, we assume that F is β-smooth for some $\beta > 0$. A function $F\colon [0,1]^n \to \mathbb{R}$ is call *β-smooth* if it is continuously differentiable, and obeys

$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|_2 \leq \beta \|\mathbf{x} - \mathbf{y}\|_2 \quad \forall\, \mathbf{x}, \mathbf{y} \in [0,1]^n \enspace .$$

Another standard assumption in the field is that the relevant convex-bodies ($\mathcal{K}_{\mathrm{N}}$ and $\mathcal{K}_{\mathrm{D}}$ in our case) are solvable, i.e., that one can efficiently optimize linear functions over them. We take a step further, and assume the ability to optimize linear functions over any convex body defined by the intersection of a polynomial number of linear constraints and constraints requiring particular vectors to belong either to $\mathcal{K}_{\mathrm{N}}$ or to $\mathcal{K}_{\mathrm{D}}$. This assumption appeared (often implicitly) in many previous works (see, for example, [Buchbinder and Feldman, 2023; Ene and Nguyen, 2016; Mualem and Feldman, 2023]), and is theoretically justified by the well-known equivalence between separability and solvability.

We often refer below to the *diameter* D of $\mathcal{K}$. This diameter is defined as $D \triangleq \max_{\mathbf{x},\mathbf{y} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}\|_2$.

2.1 Additional Vector Operations

Following Buchbinder and Feldman [2023], we use the following coordinate-wise vector operations. To reduce the number of parentheses necessary, we assume that both these operations have a higher precedence compared to vector addition and subtraction.

Definition 2.1. *Given two vectors $\mathbf{x}, \mathbf{y} \in [0,1]^n$,*

- *we denote by $\mathbf{x} \odot \mathbf{y}$ their coordinate-wise multiplication (also known as the Hadamard product).*
- *we denote by $\mathbf{x} \oplus \mathbf{y}$ their coordinate-wise probabilistic sum. In other words, for every $i \in [n]$, $(\mathbf{x} \oplus \mathbf{y})_i \triangleq x_i + y_i - x_i y_i = 1 - (1 - x_i)(1 - y_i)$.*

As was noted by [Buchbinder and Feldman, 2023], the operation $\oplus$ is symmetric and associative. We also use $\bar{0}$ and $\bar{1}$ to represent the all-zeros and all-ones vectors, respectively.

2.2 Online Optimization

In the online (regret minimization) version of the problem we consider in this work, there are L time steps.[4] In every time step $\ell \in [L]$, the adversary selects a non-negative β-smooth DR-submodular function F_t, and then the algorithm should select a distribution P_ℓ of points in $\mathcal{K} = (\mathcal{K}_{\mathrm{N}} + \mathcal{K}_{\mathrm{D}}) \cap [0,1]^n$ without knowing F_t (the function F_t is revealed to the algorithm only after P_ℓ is selected). The objective of the algorithm is to maximize $\sum_{\ell=1}^{L} \mathbb{E}_{\mathbf{x} \sim P_\ell}[F_\ell(\mathbf{x})]$, and its success in doing so is measured compared to the best fixed solution (i.e., any two vectors $\mathbf{o}_{\mathrm{N}} \in \mathcal{K}_{\mathrm{N}}$ and $\mathbf{o}_{\mathrm{D}} \in \mathcal{K}_{\mathrm{D}}$ such that $\mathbf{o}_{\mathrm{N}} + \mathbf{o}_{\mathrm{D}} \in [0,1]^n$).

[4] The number of time steps is usually denoted by T in the literature. However, we use L in this paper to avoid confusion with the parameter T traditionally used by continuous submodular maximization algorithms (including our own algorithms).

Let us elaborate a bit on the last point. If the functions $F_1, F_2, \dots, F_L$ were all known upfront, one could execute the offline algorithm we develop to get a set of solutions $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(L)}$ such that $\sum_{\ell=1}^{L} F_\ell(\mathbf{x}^{(\ell)}) \geq \psi(\sum_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{N} + \mathbf{o}_\mathrm{D}), \sum_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{D}))$ for some function ψ (the structure of the function ψ is determined by the guarantee of Theorem 4.1 below). Since an online algorithm has to select the output distribution P_ℓ before seeing the function F_ℓ, it can only guarantee

$$\sum_{\ell=1}^{L} \mathbb{E}_{\mathbf{x}\sim P_\ell}[F_\ell(\mathbf{x})] \geq \psi\Big(\sum_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{N}+\mathbf{o}_\mathrm{D}), \sum_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{D})\Big) - \mathcal{R}(L)$$

for some regret function $\mathcal{R}(L)$. Asymptotically, for our online algorithm, $\mathcal{R}(L)$ grows proportionally to $\sqrt{L}$, and therefore, for large L values, the average guarantee of our online algorithm per function F_ℓ approaches the one of our offline algorithm. As usual for online settings, we assume that the range of the functions $F_1, F_2, \dots, F_L$ is $[0, 1]$.

3 Our Technique

Our algorithms maintain vectors $\mathbf{y} \in \mathcal{K}_\mathrm{N}$ and $\mathbf{z} \in \mathcal{K}_\mathrm{D}$. Intuitively, the vector $\mathbf{y}$ is maintained by the Frank-Wolfe variant developed by Mualem and Feldman [2023] for non-down-closed polytopes, and the vector $\mathbf{z}$ is maintained by the continuous-greedy-like variant of Frank-Wolfe developed by Bian [2017a] for down-closed polytopes. Combining the two algorithms requires us to solve some technical issues. For example, it is necessary to run the two algorithms in parallel since they both depend on the coordinates of the solution growing at a bounded rate, and it is necessary to create a correlation between the algorithms to guarantee that $\mathbf{y} + \mathbf{z}$ remains within $[0, 1]^n$. However, it turns out that the more interesting question is regarding the best way to combine the two vectors $\mathbf{y}$ and $\mathbf{z}$ into the output solution of the algorithm.

The most natural approach is to consider the sum $\mathbf{y} + \mathbf{z}$ as the output solution. Unfortunately, this does not work well since it results in coordinates of the solution growing too fast. To make this more concrete, we note that our algorithms, as well as the algorithms of [Bian *et al.*, 2017a] and [Mualem and Feldman, 2023], simulate continuous algorithms working from time $t = 0$ until time $t = 1$. Consider now a particular coordinate $j \in [n]$. Up until time $t \in [0, 1]$, our algorithms spend (up to) t units of "energy" on this coordinate. A fraction $x \in [0, t]$ of this "energy" is invested in growing y_j, and the remaining $t - x$ "energy" is invested in growing z_j. By the properties of the algorithms of [Bian *et al.*, 2017a] and [Mualem and Feldman, 2023], this investment of "energy" leads to $y_i = 1 - e^{-x}$ and $z_i = 1 - e^{x-t}$, which in the worst case can make $(\mathbf{y} + \mathbf{z})_i$ as large as $2(1 - e^{-t/2})$. To get a better upper bound on the coordinates of the solution, we have to use $\mathbf{y} \oplus \mathbf{z}$ as the output solution. Note that this choice guarantees that $(\mathbf{y} \oplus \mathbf{z})_j \leq 1 - [1 - (1 - e^{-x})] \cdot [1 - (1 - e^{x-t})] = 1 - e^{-t}$, which is always better (for $t > 0$) compared to the bound of $2(1 - e^{-t/2})$ obtained above.

While the use of $\mathbf{y} \oplus \mathbf{z}$ is useful, it does not come without a cost. As mentioned above, our algorithms simulate continuous algorithms, which is a common practice in the literature about submodular maximization. To discretize these algorithms, one has to split time into steps, and then do in each step a single modification of the vectors $\mathbf{y}$ and $\mathbf{z}$ simulating all the modifications done by the continuous algorithm throughout the step. The standard way in which this is done is as follows. Assume that, at the beginning of the step, the continuous algorithm increases $\mathbf{y}$ at a rate of $\mathbf{y}'$ and $\mathbf{z}$ at a rate of $\mathbf{z}'$, then the discrete algorithm should increase $\mathbf{y}$ by $\varepsilon\mathbf{y}'$ and $\mathbf{z}$ by $\varepsilon\mathbf{z}'$, where ε is the size of the step. Unfortunately, this standard practice results in $\mathbf{y} \oplus \mathbf{z}$ changing by $\varepsilon\mathbf{y}' \odot (\bar{1} - \mathbf{z}) + \varepsilon\mathbf{z}' \odot (\bar{1} - \mathbf{y}) - \varepsilon^2 \cdot (\mathbf{y}' \odot \mathbf{z}')$. To see why this is problematic, note that in the continuous algorithm, when $\mathbf{y}$ and $\mathbf{z}$ increase at rates of $\mathbf{y}'$ and $\mathbf{z}'$, respectively, $\mathbf{y} \oplus \mathbf{z}$ increases at a rate of $\mathbf{y}' \odot (\bar{1} - \mathbf{z}) + \mathbf{z}' \odot (\bar{1} - \mathbf{y})$. Thus, the term $-\varepsilon^2 \cdot (\mathbf{y}' \odot \mathbf{z}')$ from the previous expression represents a new kind of discretization error that we need to handle.

Another hurdle worth mentioning is that the vectors $\mathbf{y}$ and $\mathbf{z}$ are updated using two different update rules inherited from the algorithms of [Bian *et al.*, 2017a] and [Mualem and Feldman, 2023], and the interaction between these update rules results in a guarantee on the output of the algorithm that depends also on the value of $F(\mathbf{z})$. Thus, it is necessary to make sure that our algorithms maintain $\mathbf{z}$ in a way that also guarantees that $F(\mathbf{z})$ has a good value. In the first version of our offline algorithm, we do that by assuming that we know the value v of the part of the optimal solution that belongs to $\mathcal{K}_\mathrm{D}$. This knowledge allows us to force the algorithm to increase $\mathbf{z}$ in a way guaranteed to make $F(\mathbf{z})$ competitive with v. In the other versions of our offline algorithm and in our online algorithm, we use a potential function argument to avoid the need to know v. This potential function argument is similar to an argument used by Feldman [2021] in a different submodular maximization setting.

4 Offline Maximization

In this section, we present and analyze our offline algorithm, whose guarantee is given by the next theorem.

Theorem 4.1. *Let $\mathcal{K}_\mathrm{N} \subseteq [0, 1]^n$ be a general solvable convex set, $\mathcal{K}_\mathrm{D} \subseteq [0, 1]^n$ be a down-closed solvable convex set, and $F\colon [0, 1]^n \to \mathbb{R}_{\geq 0}$ be a non-negative β-smooth DR-submodular function. Then, there exists a polynomial time algorithm that, given an error parameter $\varepsilon \in (0, 1)$, outputs vectors $\mathbf{w} \in (\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0, 1]^n$ such that*

$$\begin{aligned} F(\mathbf{w}) \geq (1 - m)\cdot &\max_{t_s \in [0,1]} \max_{T \in [t_s, 1]} \Big\{ ((T - t_s)e^{-T} - O(\varepsilon)) \cdot F(\mathbf{o}_\mathrm{D}^{(2)}) \\ &+ \Big(\frac{t_s^2 \cdot e^{-t_s - T}}{2} - O(\varepsilon)\Big) \cdot F(\mathbf{o}_\mathrm{D}^{(1)}) + (e^{-T} - e^{-t_s - T} \\ &- O(\varepsilon)) \cdot F(\mathbf{o}_\mathrm{N} + \mathbf{o}_\mathrm{D}^{(1)}) \Big\} - O(\tfrac{\varepsilon\beta D^2}{1-m}) \ , \end{aligned}$$

where $m = \min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}} \|\mathbf{x}\|_\infty$, D is the diameter of $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0, 1]^n$, $\mathbf{o}_\mathrm{N} \in \mathcal{K}_\mathrm{N}$ and $\mathbf{o}_\mathrm{D}^{(1)} \in \mathcal{K}_\mathrm{D}$ are any vectors whose sum belongs to $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0, 1]^n$, and $\mathbf{o}_\mathrm{D}^{(2)}$ is any vector in $\mathcal{K}_\mathrm{D}$.[5]

[5]The vectors $\mathbf{o}_\mathrm{D}^{(1)}$ and $\mathbf{o}_\mathrm{D}^{(2)}$ can be identical.

It is interesting to note that Theorem 4.1 recovers two guarantees of previous works. Specifically, by setting $T = 1, t_s = 0$ and $\mathcal{K}_\mathrm{N} = \{\bar{0}\}$, the theorem implies e^{-1}-approximation for maximizing a DR-submodular function subject to a down-closed polytope $\mathcal{K}_\mathrm{D}$, recovering the result of [Bian *et al.*, 2017a]. Similarly, by setting $T = t_s = \ln 2$ and $\mathcal{K}_\mathrm{D} = \{\bar{0}\}$, Theorem 4.1 implies $1/4(1-m)$-approximation for maximizing a DR-submodular function subject to a general polytope $\mathcal{K}_\mathrm{N}$, recovering the result of [Mualem and Feldman, 2023].

For ease of the presentation, we have three versions of our offline algorithm. The first version, appearing below as Algorithm 1, proves Theorem 4.1 under the assumption that $F(\mathbf{o}_\mathrm{D}^{(1)})$ is known. In the full version of this paper [Mualem *et al.*, 2024], we present the two other versions of our offline algorithm that prove Theorem 4.1 without making this assumption. One version is theoretically natural, and is the base for our online algorithm described in Section 5. However, to get the best results in practice, it is natural to make some modifications to this natural version (including ones motivated by the work of [Bian *et al.*, 2017a]), which do not improve the theoretical guarantee of the algorithm and cannot be extended to the online version of the algorithm. Both the natural and the modified version of our algorithm are studied in the offline experiments described in Section 6.

Recall that Algorithm 1 proves Theorem 4.1 under the assumption that $F(\mathbf{o}_\mathrm{D}^{(1)})$ is known. In its description, we assume for simplicity that ε^{-1} is integral. If this is not the case, ε can be replaced with $\lceil\varepsilon^{-1}\rceil^{-1}$, which is smaller than ε by at most a factor of 2.

Algorithm 1 `Frank-Wolfe/Continuous-Greedy Hybrid for Known` $F(\mathbf{o}_\mathrm{D}^{(1)})$

1: Let $\mathbf{y}^{(0)} \leftarrow \arg\min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}} \|\mathbf{x}\|_\infty$ and $\mathbf{z}^{(0)} \leftarrow \bar{0}$.
2: **for** $i = 1$ to ε^{-1} **do**
3: Solve the following linear program:

$$\begin{array}{ll} \max & \langle \nabla F(\mathbf{y}^{(i-1)} \oplus \mathbf{z}^{(i-1)}) \odot (\bar{1} - \mathbf{z}^{(i-1)}), \\ & \quad \mathbf{a}^{(i)} + \mathbf{b}^{(i)} \odot (\bar{1} - \mathbf{y}^{(i-1)})\rangle \\ \text{s.t.} & \mathbf{a}^{(i)} \in \mathcal{K}_\mathrm{N}, \mathbf{b}^{(i)} \in \mathcal{K}_\mathrm{D} \\ & \langle \mathbf{b}^{(i)} \odot (\mathbf{1} - \mathbf{z}^{(i-1)}), \nabla F(\mathbf{z}^{(i-1)})\rangle \\ & \quad \geq (1-\varepsilon)^{i-1} \cdot F(\mathbf{o}_\mathrm{D}^{(1)}) - F(\mathbf{z}^{(i-1)}) \\ & \mathbf{a}^{(i)} + \mathbf{b}^{(i)} \odot (\bar{1} - \mathbf{y}^{(i-1)}) \leq \bar{1} \end{array}$$

4: Let $\mathbf{y}^{(i)} \leftarrow (1-\varepsilon)\cdot \mathbf{y}^{(i-1)} + \varepsilon \cdot \mathbf{a}^{(i)}$.
5: Let $\mathbf{z}^{(i)} \leftarrow \mathbf{z}^{(i-1)} + \varepsilon \cdot (1 - \mathbf{z}^{(i-1)}) \odot \mathbf{b}^{(i)}$.
6: **end for**
7: Return a vector maximizing F among all the vectors in $\{\mathbf{y}^{(i)} \oplus \mathbf{z}^{(i)} \mid i \in \mathbb{Z}, 0 \leq i \leq \varepsilon^{-1}\}$.

5 Online Maximization

In this section, we consider the online (regret minimization) setting described in Section 2.2. The algorithm we present and analyze has the guarantee given by the next theorem.

Theorem 5.1. *Let $\mathcal{K}_\mathrm{N} \subseteq [0,1]^n$ be a general solvable convex set, and $\mathcal{K}_\mathrm{D} \subseteq [0,1]^n$ be a down-closed solvable convex set. If the adversary chooses only non-negative β-smooth DR-submodular functions $F_\ell\colon [0,1]^n \to [0,1]$, then there exists a polynomial time online algorithm that, given value $\varepsilon \in (0,1)$, outputs at every time step $\ell \in [L]$ a distribution P_ℓ over vectors in $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$ guaranteeing that*

$$\begin{aligned} &\sum\nolimits_{\ell=1}^{L} \mathbb{E}_{\mathbf{x}\sim P_\ell}[F_\ell(\mathbf{x})] \geq (1-m)\cdot \max_{t_s\in[0,1]} \max_{T\in[t_s,1]} \\ &\quad \Big\{\Big((T-t_s)e^{-T} + \frac{t_s^2 \cdot e^{-t_s-T}}{2} - O(\varepsilon)\Big)\cdot \sum\nolimits_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{D}) \\ &\quad + (e^{-T} - e^{-t_s-T} - O(\varepsilon))\cdot \sum\nolimits_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{N}+\mathbf{o}_\mathrm{D})\Big\} \\ &\quad - O(\varepsilon\beta L D^2) - O(DG\sqrt{L}) - O(\sqrt{L\ln\varepsilon^{-1}})\ , \end{aligned}$$

where $m = \min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}}\|\mathbf{x}\|_\infty$, D is the diameter of $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$, $G = \max\{\max_{\mathbf{x}\in(\mathcal{K}_\mathrm{N}+\mathcal{K}_\mathrm{D})\cap[0,1]^n} \|\nabla F_t(\mathbf{x})\|_2, \max_{\mathbf{x}\in\mathcal{K}_\mathrm{D}} \|\nabla F_t(\mathbf{x})\|_2,\}$, and $\mathbf{o}_\mathrm{N} \in \mathcal{K}_\mathrm{N}$ and $\mathbf{o}_\mathrm{D} \in \mathcal{K}_\mathrm{D}$ are any vectors whose sum belongs to $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$.

Note the following two remarks about Theorem 5.1.

- Section 2.2 states that the regret of our online algorithm (compared to the offline algorithm) asymptotically grows as $\sqrt{L}$. This might seem to contradict the presence of the error term $O(\varepsilon\beta L D^2)$ in the guarantee of Theorem 5.1. However, this is not the case since this error term is part of the ψ functions mentioned in Section 2.2 (because the term $O(\frac{\varepsilon\beta D^2}{1-m})$ appears in the guarantee of Theorem 4.1).
- Theorem 4.1 considers two vectors $\mathbf{o}_\mathrm{D}^{(1)}, \mathbf{o}_\mathrm{D}^{(2)} \in \mathcal{K}_\mathrm{D}$. A similar result could be proved in Theorem 5.1. However, in the online setting, the algorithm is required to be competitive against any fixed solution. Thus, we felt that it is more natural to state Theorem 5.1 for the case in which $\mathbf{o}_\mathrm{D}^{(1)}$ and $\mathbf{o}_\mathrm{D}^{(2)}$ are the same vector (denoted by $\mathbf{o}_\mathrm{D}$).

The central component in the proof of Theorem 5.1 is the following proposition, which is a variant of Theorem 5.1 in which (i) t_s is a parameter of the algorithm, and (ii) the algorithm outputs in each time step a single vector rather than a distribution over vectors.

Proposition 5.2. *Let $\mathcal{K}_\mathrm{N} \subseteq [0,1]^n$ be a general solvable convex set, and $\mathcal{K}_\mathrm{D} \subseteq [0,1]^n$ be a down-closed solvable convex set. If the adversary chooses only non-negative β-smooth DR-submodular functions $F_\ell\colon [0,1]^n \to [0,1]$, then there exists a polynomial time online algorithm that, given parameters $t_s \in [0,1]$ and $\varepsilon \in (0,1)$, outputs at every time step $\ell \in [L]$ a vector $\mathbf{w}^{(\ell)} \in (\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$ guaranteeing that*

$$\begin{aligned} &\sum\nolimits_{\ell=1}^{L} F_\ell(\mathbf{w}^{(\ell)}) \geq (1-m)\cdot \max_{T\in[t_s,1]} \\ &\quad \Big\{\Big((T-t_s)e^{-T} + \frac{t_s^2 \cdot e^{-t_s-T}}{2} - O(\varepsilon)\Big)\cdot \sum\nolimits_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{D}) \\ &\quad + (e^{-T} - e^{-t_s-T} - O(\varepsilon))\cdot \sum\nolimits_{\ell=1}^{L} F_\ell(\mathbf{o}_\mathrm{N}+\mathbf{o}_\mathrm{D})\Big\} \\ &\quad - O(\varepsilon\beta L D^2) - O(DG\sqrt{L})\ , \end{aligned}$$

where $m = \min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}}\|\mathbf{x}\|_\infty$, D is the diameter of $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$, $G = \max\{\max_{\mathbf{x}\in(\mathcal{K}_\mathrm{N}+\mathcal{K}_\mathrm{D})\cap[0,1]^n} \|\nabla F_t(\mathbf{x})\|_2,$

$\max_{\mathbf{x}\in\mathcal{K}_\mathrm{D}} \|\nabla F_t(\mathbf{x})\|_2\}$*, and* $\mathbf{o}_\mathrm{N} \in \mathcal{K}_\mathrm{N}$ *and* $\mathbf{o}_\mathrm{D} \in \mathcal{K}_\mathrm{D}$ *are any vectors whose sum belongs to* $(\mathcal{K}_\mathrm{N} + \mathcal{K}_\mathrm{D}) \cap [0,1]^n$.

Algorithm 2 `Online Frank-Wolfe/Continuous-Greedy Hybrid`

for $i = 1$ to ε^{-1} **do**
  Initialize an instance $\mathcal{E}_i$ of Regularized-Follow-the-Leader for the convex body $\{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \in \mathcal{K}_\mathrm{N}, \mathbf{b} \in \mathcal{K}_\mathrm{D}, \mathbf{a} + \mathbf{b} \in [0,1]^n\}$.
end for
Let $m \leftarrow \min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}} \|\mathbf{x}\|_\infty$.
for $\ell = 1$ to L **do**
  Let $\mathbf{y}^{(0,\ell)} \leftarrow \arg\min_{\mathbf{x}\in\mathcal{K}_\mathrm{N}} \|\mathbf{x}\|_\infty$, $\mathbf{z}^{(0,\ell)} \leftarrow \bar{0}$.
  for $i = 1$ to $\varepsilon^{-1} t_s$ **do**
    Let $(\mathbf{a}^{(i,\ell)}, \mathbf{b}^{(i,\ell)})$ be the vector picked by $\mathcal{E}_i$ in time step ℓ.
    Let $\mathbf{y}^{(i,\ell)} \leftarrow (1-\varepsilon)\cdot \mathbf{y}^{(i-1,\ell)} + \varepsilon \cdot \mathbf{a}^{(i,\ell)}$.
    Let $\mathbf{z}^{(i,\ell)} \leftarrow \mathbf{z}^{(i-1,\ell)} + \varepsilon \cdot (1 - \mathbf{z}^{(i-1,\ell)}) \odot \mathbf{b}^{(i,\ell)}$.
  end for
  for $i = \varepsilon^{-1} t_s + 1$ to ε^{-1} **do**
    Let $\mathbf{a}^{(i,\ell)} \leftarrow \mathbf{y}^{(i-1,\ell)}$.
    Let $\mathbf{b}^{(i,\ell)}$ be a vector consisting of the last n coordinates of the vector picked by $\mathcal{E}_i$ in time step ℓ.
    Let $\mathbf{y}^{(i,\ell)} \leftarrow \mathbf{y}^{(i-1,\ell)}$.
    Let $\mathbf{z}^{(i,\ell)} \leftarrow \mathbf{z}^{(i-1,\ell)} + \varepsilon \cdot (1 - \mathbf{z}^{(i-1,\ell)}) \odot \mathbf{b}^{(i,\ell)}$.
  end for
  Set the vector $\mathbf{y}^{(\varepsilon^{-1},\ell)} \oplus \mathbf{z}^{(\varepsilon^{-1},\ell)}$ as the output for time step ℓ.
  for $i = 1$ to $\varepsilon^{-1} t_s$ **do**
    Let $\mathbf{g}^{(i,\ell)}$ be the vector in $\mathbb{R}^{2n}$ obtained as follows. The first n coordinates of this vector are given by $e^{2\varepsilon i}\nabla F_\ell(\mathbf{y}^{(i-1,\ell)} \oplus \mathbf{z}^{(i-1,\ell)}) \odot (\bar{1} - \mathbf{z}^{(i-1,\ell)})$, and the other n coordinates of $\mathbf{g}^{(i,\ell)}$ are equal to $e^{2\varepsilon i}\nabla F_\ell(\mathbf{y}^{(i-1,\ell)} \oplus \mathbf{z}^{(i-1,\ell)}) \odot (\bar{1} - \mathbf{z}^{(i-1,\ell)}) \odot (\bar{1} - \mathbf{y}^{(i-1,\ell)}) + (1-m)\cdot e^{\varepsilon i}(t_s - \varepsilon i)\cdot \nabla F_\ell(\mathbf{z}^{(i-1,\ell)}) \odot (1 - \mathbf{z}^{(i-1,\ell)})$.
    Pass $\mathbf{g}^{(i,\ell)}$ as the adversarially chosen vector $\mathbf{d}^{(\ell)}$ for $\mathcal{E}_i$.
  end for
  for $i = \varepsilon^{-1} t_s + 1$ to ε^{-1} **do**
    Let $\mathbf{g}^{(i,\ell)}$ be a vector in $\mathbb{R}^{2n}$ obtained as follows. The first n coordinates of this vector are all zeros, and the other n coordinates of $\mathbf{g}^{(i,\ell)}$ are equal to $\nabla F_\ell(\mathbf{y}^{(i-1,\ell)} \oplus \mathbf{z}^{(i-1,\ell)}) \odot (\bar{1} - \mathbf{z}^{(i-1,\ell)}) \odot (\bar{1} - \mathbf{y}^{(i-1,\ell)})$.
    Pass $\mathbf{g}^{(i,\ell)}$ as the adversarially chosen vector $\mathbf{d}^{(\ell)}$ for $\mathcal{E}_i$.
  end for
end for

Let us explain why Proposition 5.2 implies Theorem 5.1. The guarantee given in Theorem 5.1 remains unchanged (up to the constants hidden by the big O notation) if the maximum over $t_s \in [0,1]$ is restricted to the set of $O(\varepsilon^{-1})$ values that are integer multiples of ε between 0 and 1. If we knew upfront what value from this set leads to the best guarantee for Proposition 5.2, then we could use the algorithm of this proposition to get the guarantee of Theorem 5.1.

Unfortunately, in reality, we do not usually know upfront the best value for t_s. Nevertheless, since there are only $O(\varepsilon^{-1})$ such values that need to be considered, we can use a regret minimization algorithm (such as the one of [Cesa-Bianchi *et al.*, 2007]) to get in each time step a distribution over solutions whose expected value is at least as good as the guarantee of Proposition 5.2 for the best value of t_s, up to an error term of $O(\sqrt{L \ln \varepsilon^{-1}})$. Thus, Theorem 5.1 indeed follows from Proposition 5.2.

At this point, we would like to describe the algorithm that we use to prove Proposition 5.2, which is given as Algorithm 2. Similarly to the Meta-Frank-Wolfe algorithm of [Chen *et al.*, 2018], Algorithm 2 uses multiple instances of an online algorithm for linear optimization. Specifically, we use the algorithm Regularized-Follow-the-Leader due to [Abernethy *et al.*, 2008], which has the following behavior. There are L time steps. In every time step $\ell \in [L]$, the algorithm selects a vector $\mathbf{u}^{(\ell)} \in \mathcal{K}$ for some given convex body $\mathcal{K}$, and then an adversary reveals to the algorithm a vector $\mathbf{d}^{(\ell)}$ that was chosen independently of $\mathbf{u}^{(\ell)}$. Regularized-Follow-the-Leader guarantees that

$$\sum_{t=1}^{L} \langle \mathbf{u}^{(\ell)}, \mathbf{d}^{(\ell)} \rangle \geq \max_{\mathbf{x}\in\mathcal{K}} \sum_{\ell=1}^{L} \langle \mathbf{x}, \mathbf{d}^{(\ell)} \rangle - D'G'\sqrt{2L} \ ,$$

where $G' = \max_{1\leq \ell \leq L} \|\mathbf{d}^{(\ell)}\|_2$ and D' is the diameter of $\mathcal{K}$.

Algorithm 2 follows the same general structure of our offline algorithm, with two main modifications: the linear programs of the offline algorithm are replaced by instances of the online linear optimization algorithm that we use, and the vector corresponding to $i = T^{-1}$ is used as the output instead of the best vector among multiple options.

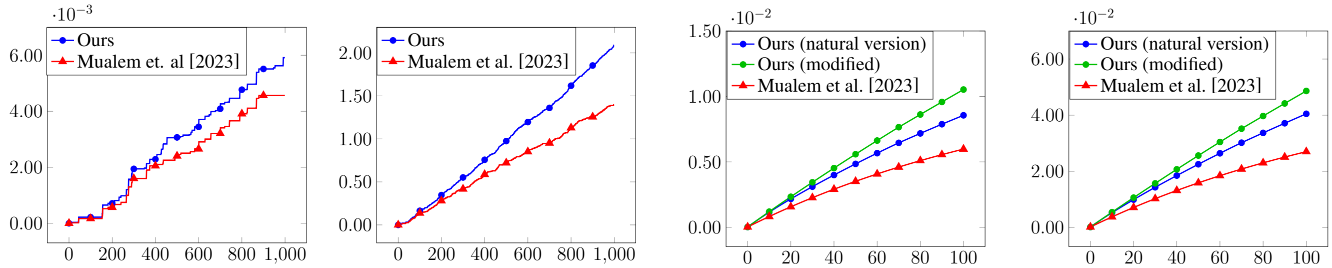
In the pseudocode of Algorithm 2, we implicitly assume that $\varepsilon \leq 1/70$ and εt_s is integral. The first assumption is without loss of generality since we can decrease ε to be $1/70$ if its original value is larger, and the second assumption is without loss of generality because the coefficients in the guarantee of Proposition 5.2 change only by $O(\varepsilon)$ when t_s is modified by up to ε to make $t_s\varepsilon$ integral.

6 Applications and Experimental Results

In this work, we study the empirical performance of the offline and online algorithms described in the previous sections on three machine learning tasks. We present one application here, and defer the rest to the full version of this paper [Mualem *et al.*, 2024]. The full version also includes empirical stability and ablation studies of our offline algorithm.

6.1 Revenue Maximization

As our first experimental setting, we consider the following revenue maximization setting, which was also considered by [Mualem and Feldman, 2023; Soma and Yoshida, 2017; Thắng and Srivastav, 2021]. The objective of some company is to promote a product to users to boost revenue through the "word-of-mouth" effect. The problem of optimizing this objective can be formalized as follows. The input is a weighted undirected graph $G = (V, E)$ representing a social network, where w_{ij} represents the weight of the edge between vertex



(a) Online Algorithms on the Advogato network.

(b) Online Algorithms on the Facebook network.

(c) Offline Algorithms on the Advogato network.

(d) Offline Algorithms on the Facebook network.

Figure 1: Results of the revenue maximization experiments. In each plot, the x-axis is the timestep, and the y-axis is the corresponding function value.

i and vertex j (with $w_{ij} = 0$ if the edge (i, j) is absent from the graph). If the company allocates a cost of x_i units to a user $i \in V$, then that user becomes an advocate of the product with a probability of $1 - (1 - p)^{x_i}$, where $p \in (0, 1)$ is a parameter. Note that this formula implies that each ε unit of cost invested in the user independently contributes to the chance of making the user an advocate. Furthermore, by investing a full unit in the user, the user becomes an advocate with a probability of p [Soma and Yoshida, 2017].

Given a set $S \subseteq V$ of users who have become advocates for the product, the expected revenue is related to the total influence of the users in S on non-advocate users, which is formally expressed as $\sum_{i\in S} \sum_{j\in V\setminus S} w_{ij}$. Hence, the objective function $F\colon [0, 1]^V \to \mathbb{R}_{\geq 0}$ in this setting is defined as the expectation of the aforementioned expression, i.e.,

$$F(\mathbf{x}) = \mathbb{E}_S\left[\sum_{i\in S} \sum_{j\in V\setminus S} w_{ij}\right] = \sum_{i\in V} \sum_{j\in V, i\neq j} w_{ij}(1 - (1 - p)^{x_i})(1 - p)^{x_j} \enspace .$$

It has been demonstrated that F is a non-monotone DR-submodular function [Soma and Yoshida, 2017].

We conducted experiments in both online and offline scenarios based on instances of the aforementioned setting derived from two distinct datasets. The first dataset is sourced from a Facebook network [Viswanath *et al.*, 2009], encompassing $64K$ users (vertices) and $1M$ unweighted relationships (edges). The second dataset is based on the Advogato network [Massa *et al.*, 2009], comprising $6.5K$ users (vertices) and $61K$ weighted relationships (edges).

Online Setting. In our online experiments, inspired by [Mualem and Feldman, 2023], we set the number of time steps to $L = 1000$, with $p = 0.0001$. At each time step ℓ, the objective function is defined by selecting a uniformly random subset $V_\ell \subseteq V$ of a given size, and then retaining only edges connecting two vertices of V_ℓ. For the Advogato network, V_ℓ is of size 200, and for the larger Facebook network, V_ℓ is of size 15,000. The above objective functions are optimized subject to the constraint $0.1 \leq \sum_i x_i \leq 1$, which represents both minimum and maximum investment requirements. Notably, the intersection of this constraint with the implicit box constraint forms a non-down-monotone feasibility polytope. However, this polytope can be decomposed into two polytopes: (i) $\mathcal{K}_\mathrm{N}$, a polytope defined by the equality $\sum_{i=1}^n x_i = 0.1$, and (ii) $\mathcal{K}_\mathrm{D}$, a down-closed polytope defined by $\sum_{i=1}^n x_i \leq 0.9$. Observe that $(\mathcal{K}_\mathrm{D} + \mathcal{K}_\mathrm{N}) \cap [0, 1]^n$ is indeed the original polytope, and thus, this is a valid decomposition of this polytope.

In our experiments, we have compared the performance of our algorithm from Section 5 with the online algorithm of Mualem & Feldman [2023], which is the current state-of-the-art algorithm for the online setting. In both algorithms, we have set the number of online linear optimizers used to be 100 (which corresponding to setting the error parameter ε to 0.01 in our algorithm and to ${}^{\ln 2}/_{100}$ in the algorithm of Mualem & Feldman). The results of our experiments on the Advogato and Facebook networks can be found in Figures 1a and 1b, respectively. In both experiments, our algorithm significantly outperforms the state-of-the-art algorithm.

Offline Setting. Our offline experiments are similar to their online counterparts. However, since there is only one objective function in this setting, we run the experiment on the entire network graph. In this setting, we compared our offline algorithm (see Section 4 for a discussion of the versions of this algorithm used in the experiments) with the current state-of-the-art algorithm from [Mualem and Feldman, 2023] (which is an explicit version of the algorithm of Du [2022]). Both algorithms have been executed for $T = 100$ iterations, and the error parameter ε was set accordingly (which again means 0.01 in our algorithm and ${}^{\ln 2}/_{100}$ in the algorithm of Mualem & Feldman). The results of the offline experiments on the Advogato and Facebook networks can be found in Figures 1c and 1d, respectively. One can observe that our method consistently outperforms the previous state-of-the-art.

7 Conclusion

We have presented novel offline and online algorithms for DR-submodular maximization subject to a general convex body constraint. Our algorithms are able to provide a smooth interpolation between the approximability of general and down-closed convex bodies by considering a decomposition of the convex body constraint into a down-closed convex body and a general convex body. In addition to giving a theoretical analysis of our algorithms, we have demonstrated their empirical superiority (compared to state-of-the-art methods) in various online and offline machine learning applications.

Acknowledgements

The work of Loay Mualem and Moran Feldman has been partially supported by Israel Science Foundation (ISF) grant number 459/20.

References

[Abernethy *et al.*, 2008] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory (COLT)*, pages 263–273, 2008.

[Bian *et al.*, 2017a] An Bian, Kfir Yehuda Levy, Andreas Krause, and Joachim M. Buhmann. Non-monotone continuous DR-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 486–496, 2017.

[Bian *et al.*, 2017b] Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 111–120. PMLR, 2017.

[Bian *et al.*, 2019] Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous DR-submodular maximization and applications to provable mean field inference. In *International Conference on Machine Learning (ICML)*, pages 644–653. PMLR, 2019.

[Buchbinder and Feldman, 2023] Niv Buchbinder and Moran Feldman. Constrained submodular maximization via new bounds for dr-submodular functions. *CoRR*, abs/2311.01129, 2023.

[Calinescu *et al.*, 2011] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[Cesa-Bianchi *et al.*, 2007] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2-3):321–352, 2007.

[Chen *et al.*, 2018] Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1896–1905. PMLR, 2018.

[Chen *et al.*, 2019] Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 2047–2056. PMLR, 2019.

[Du *et al.*, 2022] Donglei Du, Zhicheng Liu, Chenchen Wu, Dachuan Xu, and Yang Zhou. An improved approximation algorithm for maximizing a DR-submodular function over a convex set. *arXiv preprint arXiv:2203.14740*, 2022.

[Du, 2022] Donglei Du. Lyapunov function approach for approximation algorithm design and analysis: with applications in submodular maximization. *CoRR*, abs/2205.12442, 2022.

[Dürr *et al.*, 2021] Christoph Dürr, Nguyên Kim Thǎng, Abhinav Srivastav, and Léo Tible. Non-monotone DR-submodular maximization over general convex sets. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2148–2154, 2021.

[El Halabi *et al.*, 2023] Marwa El Halabi, Federico Fusco, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub Tarnawski. Fairness in streaming submodular maximization over a matroid constraint. In *International Conference on Machine Learning*, pages 9150–9171. PMLR, 2023.

[Ene and Nguyen, 2016] Alina Ene and Huy L. Nguyen. Constrained submodular maximization: Beyond 1/e. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 248–257. IEEE Computer Society, 2016.

[Feige *et al.*, 2011] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011.

[Feldman *et al.*, 2011] Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Foundations of Computer Science (FOCS)*, pages 570–579, 2011.

[Feldman, 2021] Moran Feldman. Guess free maximization of submodular and linear sums. *Algorithmica*, 83(3):853–878, 2021.

[Halabi *et al.*, 2023] Marwa El Halabi, Jakub Tarnawski, Ashkan Norouzi-Fard, and Thuy-Duong Vuong. Fairness in submodular maximization over a matroid constraint. *arXiv preprint arXiv:2312.14299*, 2023.

[Hassani *et al.*, 2017] Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[Massa *et al.*, 2009] Paolo Massa, Martino Salvetti, and Danilo Tomasoni. Bowling alone and trust decline in social network sites. In *IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pages 658–663. IEEE Computer Society, 2009.

[Mitra *et al.*, 2021] Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular + concave. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11577–11591, 2021.

[Mualem and Feldman, 2022] Loay Mualem and Moran Feldman. Using partial monotonicity in submodular maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[Mualem and Feldman, 2023] Loay Mualem and Moran Feldman. Resolving the approximability of offline and online non-monotone DR-submodular maximization over general convex sets. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pages 2542–2564. PMLR, 2023.

[Mualem *et al.*, 2023] Loay Mualem, Ethan R Elenberg, Moran Feldman, and Amin Karbasi. Submodular minimax optimization: Finding effective sets. *arXiv preprint arXiv:2305.16903*, 2023.

[Mualem *et al.*, 2024] Loay Mualem, Murad Tukan, and Moran Fledman. Bridging the gap between general and down-closed convex sets in submodular maximization. *arXiv preprint arXiv:2401.09251*, 2024.

[Nemhauser and Wolsey, 1978] G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Math. Oper. Res.*, 3(3):177–188, 1978.

[Niazadeh *et al.*, 2020] Rad Niazadeh, Tim Roughgarden, and Joshua R Wang. Optimal algorithms for continuous non-monotone submodular and DR-submodular maximization. *Journal of Machine Learning Research*, 21(1):4937–4967, 2020.

[Pedramfar *et al.*, 2023] Mohammad Pedramfar, Christopher John Quinn, and Vaneet Aggarwal. A unified approach for maximizing continuous DR-submodular functions. *arXiv preprint arXiv:2305.16671*, 2023.

[Salehi *et al.*, 2017] Mehraveh Salehi, Amin Karbasi, Dustin Scheinost, and R. Todd Constable. A submodular approach to create individualized parcellations of the human brain. In Maxime Descoteaux, Lena Maier-Hein, Alfred M. Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 10433 of *Lecture Notes in Computer Science*, pages 478–485. Springer, 2017.

[Shi *et al.*, 2021] Guangyao Shi, Ishat E Rabban, Lifeng Zhou, and Pratap Tokekar. Communication-aware multi-robot coordination with submodular maximization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8955–8961. IEEE, 2021.

[Soma and Yoshida, 2017] Tasuku Soma and Yuichi Yoshida. Non-monotone DR-submodular function maximization. In Satinder Singh and Shaul Markovitch, editors, *AAAI Conference on Artificial Intelligence*, pages 898–904. AAAI Press, 2017.

[Thắng and Srivastav, 2021] Nguyễn Kim Thắng and Abhinav Srivastav. Online non-monotone DR-submodular maximization. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9868–9876. AAAI Press, 2021.

[Tukan *et al.*, 2023] Murad Tukan, Fares Fares, Yotam Grufinkle, Ido Talmor, Loay Mualem, Vladimir Braverman, and Dan Feldman. Orbslam3-enhanced autonomous toy drones: Pioneering indoor exploration. *arXiv preprint arXiv:2312.13385*, 2023.

[Viswanath *et al.*, 2009] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *ACM SIGCOMM Workshop on Social Networks (WOSN)*, August 2009.

[Vondrák, 2013] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42(1):265–304, 2013.

[Yuan and Tang, 2023] Jing Yuan and Shaojie Tang. Group fairness in non-monotone submodular maximization. *J. Comb. Optim.*, 45(3):88, 2023.

[Zhang *et al.*, 2019] Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.