Online Non-Monotone DR-Submodular Maximization

Nguyễn Kim Thắng, Abhinav Srivastav

IBISC, Univ. Evry, University Paris-Saclay, France
kimthang.nguyen@univ-evry.fr, abhinavsriva@gmail.com

Abstract

In this paper, we study fundamental problems of maximizing DR-submodular continuous functions that have real-world applications in the domain of machine learning, economics, operations research and communication systems. It captures a subclass of non-convex optimization that provides both theoretical and practical guarantees. Here, we focus on minimizing regret for online arriving non-monotone DR-submodular functions over down-closed and general convex sets.

First, we present an online algorithm that achieves a 1/e-approximation ratio with the regret of $O(T^{3/4})$ for maximizing DR-submodular functions over any down-closed convex set. Note that, the approximation ratio of 1/e matches the best-known guarantee for the offline version of the problem. Next, we give an online algorithm that achieves an approximation guarantee (depending on the search space) for the problem of maximizing non-monotone continuous DR-submodular functions over a general convex set (not necessarily down-closed). To best of our knowledge, no prior algorithm with approximation guarantee was known for non-monotone DR-submodular maximization in the online setting. Finally we run experiments to verify the performance of our algorithms on problems arising in machine learning domain with the real-world datasets.

Introduction

Continuous DR-submodular optimization is a subclass of non-convex optimization that is an upcoming frontier in machine learning. Roughly speaking, a differentiable non-negative bounded function $F : [0,1]^n \rightarrow [0,1]$ is *DR-submodular* if $\nabla F(\boldsymbol{x}) \geq \nabla F(\boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^n$ where $x_i \leq y_i$ for every $1 \leq i \leq n$. (Note that, w.l.o.g. after normalization, assume that F has values in $[0,1]$.) Intuitively, continuous DR-submodular functions represent the diminishing returns property or the economy of scale in continuous domains. DR-submodularity has been of great interest (Bian, Buhmann, and Krause 2019; Bian et al. 2017a; Chen, Hassani, and Karbasi 2018; Hassani, Soltanolkotabi, and Karbasi 2017; Niazadeh, Roughgarden, and Wang 2018; Staib and Jegelka 2017). Many problems arising in machine learning and statistics, such as Non-definite Quadratic Programming (Ito and Fujimaki 2016), Determinantal Point Processes

(Kulesza, Taskar et al. 2012), log-submodular models (Djolonga and Krause 2014), to name a few, have been modelled using the notion of continuous DR-submodular functions.

In the past decade, online computational framework has been quite successful for tackling a wide variety of challenging problems and capturing many real-world problems with uncertainty. In this computational framework, we focus on the model of online learning. In online learning, at any time step, given a history of actions and a set of associated reward functions, online algorithm first chooses an action from a set of feasible actions; then, an adversary subsequently selects a reward function. The objective is to perform as good as the best fixed action in hindsight. This setting have been extensively explored in the literature, especially in context of convex functions (Hazan 2016).

In fact, several algorithms with theoretical approximation guarantees are known for maximizing (offline and online) DR-submodular functions. However, these guarantees hold under the assumptions that are based on the monotonicity of functions coupled with the structure of convex sets such as unconstrained hypercube and down-closed. Though, a majority of real-world problems can be formulated as *non-monotone* DR-submodular functions over convex sets that might not be necessarily down-closed. For example, the problems of Determinantal Point Processes, log-submodular models, etc can be viewed as non-monotone DR-submodular maximization problems. Although several algorithms for non-monotone (set) submodular maximization are known and they are built on maximizing the corresponding multilinear extensions (a particular class of DR-submodular functions), they do not extend to general DR-submodular functions since the optimal solutions of the former are integer points while the ones of the latter are not. Besides, general convex sets include conic convex sets, up-closed convex sets, mixture of covering and packing linear constraints, etc. which appear in many applications. Among others, conic convex sets play a crucial role in convex optimization. Conic programming (Boyd and Vandenberghe 2004) — an important subfield of convex optimization — consists of optimizing an objective function over a conic convex set. Conic programming reduces to linear programming and semi-definite programming when the objective function is linear and the convex cones are the positive orthant $\mathbb{R}^n_+$ and positive semidefinite matrices $\mathbb{S}^n_+$, respectively. Optimizing non-monotone DR-submodular

functions over a (bounded) conic convex set (not necessarily downward-closed) is an interesting and important problem both in theory and in practice. To best of our knowledge, no prior work has been done on maximizing DR-submodular functions over conic sets in online setting. The limit of current theory (Bian, Buhmann, and Krause 2019; Bian et al. 2017a; Chen, Hassani, and Karbasi 2018; Hassani, Soltanolkotabi, and Karbasi 2017; Niazadeh, Roughgarden, and Wang 2018; Zhang et al. 2019) motivates us to develop online algorithms for non-monotone functions.

In this work, we explore the *online* problem of maximizing *non-monotone* DR-submodular functions over a hypercube and down-closed[1] and over a *general* convex sets. Formally, we consider the following setting for DR-submodular maximization: given a convex domain $\mathcal{K} \subseteq [0,1]^n$ in advance, at each time step $t = 1, 2, \ldots, T$, the online algorithm first selects a vector $\boldsymbol{x}^t \in \mathcal{K}$. Subsequently, the adversary reveals a non-monotone continuous DR-submodular function F^t and the algorithm receives a reward of $F^t(\boldsymbol{x}^t)$. The objective is also to maximize the total reward. We say that an algorithm achieves a $(r, R(T))$-*regret* if

$$\sum_{t=1}^{T} F^t(\boldsymbol{x}^t) \geq r \cdot \max_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} F^t(\boldsymbol{x}) - R(T)$$

In other words, r is the approximation ratio that measures the quality of the online algorithm compared to the best fixed solution in hindsight and $R(T)$ represents the regret in the classic terms. Equivalently, we say that the algorithm has r-*regret* at most $R(T)$. Our goal is to design online algorithms with $(r, R(T))$-regret where $0 < r \leq 1$ is as large as possible, and $R(T)$ is sub-linear in T, i.e., $R(T) = o(T)$.

Our Contributions and Techniques

In this paper, we provide algorithms with performance guarantees for the problem over down-closed and general convex sets. Our contributions and techniques are summarized as follows (also see Table 1, the entries in the red correspond to our contribution).

Maximizing Online Non-monotone DR-submodular Functions Over Down-closed Convex Sets The natures of online DR-submodular maximization for monotone and non-monotone functions are quite different. In the setting of monotone functions, the best approximation guarantee is $(1 - 1/e)$ (Chen, Hassani, and Karbasi 2018) but one can prove that the natural gradient descent algorithm attains 1/2-approximation (Chen, Hassani, and Karbasi 2018). However, for online non-monotone DR-submodular maximization, no constant guarantee is known and it is likely that natural algorithms such as gradient descent, mirror descent, etc do not guarantee to achieve any constant approximation, even small one. An illustrative example of poor local optima (for coordinate ascent algorithm) can be found in (Bian, Buhmann, and Krause 2019, Appendix A).

The main result of our paper is the *first* online algorithm that achieves a constant approximation ratio with the regret

of $O(T^{3/4})$ for maximizing non-monotone DR-submodular functions over any down-closed convex set where T is number of time steps. Moreover, the approximation is $1/e$ which matches to the best guarantee in the offline setting.

Our algorithm is built on the Meta-Frank-Wolfe algorithm introduced by (Chen, Hassani, and Karbasi 2018) for *monotone* DR-submodular maximization. Their meta Frank-Wolfe algorithm combines the framework of meta-actions proposed in (Streeter and Golovin 2009) with a variant of the Frank-Wolfe proposed in (Bian et al. 2017b) for maximizing monotone DR-submodular functions. Informally, at every time t, our algorithm starts from the origin $\boldsymbol{0}$ ($\boldsymbol{0} \in \mathcal{K}$ since $\mathcal{K}$ is down-closed) and executes L steps of the Frank-Wolfe algorithm where every update vector at iteration $1 \leq \ell \leq L$ is constructed by combining the output of an optimization oracle $\mathcal{E}_\ell$ and the current vector so that we can exploit the *concavity property in positive directions* of DR-submodular functions. The solution $\boldsymbol{x}^t$ is produced at the end of the L-th step. Subsequently, after observing the function F^t, the algorithm subtly defines a vector $\boldsymbol{d}_\ell^t$ and feedbacks $\langle \cdot, \boldsymbol{d}_\ell^t \rangle$ as the reward function at time t to the oracle $\mathcal{E}_\ell$ for $1 \leq \ell \leq L$. The most important distinguishing point of our algorithm compared to that in (Chen, Hassani, and Karbasi 2018) relies on the oracles. In their algorithm, they used linear oracles which are appropriate for maximizing monotone DR-submodular functions. However, it is not clear whether *linear or even convex oracles* are strong enough to achieve a constant approximation for non-monotone functions.

While aiming for a $1/e$-approximation — the best known approximation in the offline setting — we consider the following online non-convex problem, referred throughout this paper as *online vee learning* problem. At each time t, the online algorithm knows $\boldsymbol{c}^t \in \mathcal{K}$ in advance and selects a vector $\boldsymbol{x}^t \in \mathcal{K}$. Subsequently, the adversary reveals a vector $\boldsymbol{a}^t \in \mathbb{R}^n$ and the algorithm receives a reward $\langle \boldsymbol{a}^t, \boldsymbol{c}^t \vee \boldsymbol{x}^t \rangle$. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the vector $\boldsymbol{x} \vee \boldsymbol{y}$ has the ith coordinate $(x \vee y)_i = \max\{x(i), y(i)\}$. The goal is to maximize the total reward over a time horizon T.

In order to bypass the non-convexity of the online vee learning problem, we propose the following methodology. Unlike dimension reduction techniques that aim at reducing the dimension of the search space, we *lift* the search space and the functions to a higher dimension so as to benefit from their properties in this new space. Concretely, at a high level, given a convex set $\mathcal{K}$, we define a "sufficiently dense" lattice $\mathcal{L}$ in $[0,1]^n$ such that every point in $\mathcal{K}$ can be approximated by a point in the lattice. The term "sufficiently dense" corresponds to the fact that the values of any Lipschitz function can be approximated by the function values at lattice points. As the next step, we lift all lattice points in $\mathcal{K}$ to a high dimension space so that they form a subset of vertices (corners points) $\widetilde{\mathcal{K}}$ of the unit hypercube in the newly defined space. Interestingly, the reward function $\langle \boldsymbol{a}^t, \boldsymbol{c}^t \vee \boldsymbol{x}^t \rangle$ can be transformed to a linear function in the new space. Hence, our algorithm for the online vee problem consists of updating at every time step a solution in the high-dimension space using a linear oracle and projecting the solution back to the original space.

Once the solution is projected back to original space, we construct appropriate update vectors for the original DR-

---

[1]$\mathcal{K}$ is *down-closed* if for every $\boldsymbol{z} \in \mathcal{K}$ and $\boldsymbol{y} \leq \boldsymbol{z}$ then $\boldsymbol{y} \in \mathcal{K}$

		Monotone	Non-monotone
Offline	hypercube		$\frac{1}{2}$-approx (Bian, Buhmann, and Krause 2019) (Niazadeh, Roughgarden, and Wang 2018)
	down-closed	$\left(1 - \frac{1}{e}\right)$-approx (Bian et al. 2017b)	$\left(\frac{1}{e}\right)$-approx (Bian et al. 2017a)
	general	$\left(1 - \frac{1}{e}\right)$-approx (Mokhtari, Hassani, and Karbasi 2018)	$\left(\frac{1-\min_{\boldsymbol{x}\in\mathcal{K}}\|\boldsymbol{x}\|_\infty}{3\sqrt{3}}\right)$-approx (Dürr et al. 2020)
Online	hypercube		$\left(1/2, O(\sqrt{T})\right)$
	down-closed		$\left(\frac{1}{e}, O(T^{3/4})\right)$
	general	$\left(1 - \frac{1}{e}, O(\sqrt{T})\right)$ (Chen, Hassani, and Karbasi 2018)	$\left(\frac{1-\min_{\boldsymbol{x}\in\mathcal{K}}\|\boldsymbol{x}\|_\infty}{3\sqrt{3}}, O(\frac{T}{\ln T})\right)$

Table 1: Summary of results on DR-submodular maximization. The entries represent the best-known $(r, R(T))$-regret; our results are shown in red. The entry in blue is not explicitly stated in literature but can be deduced from (Ene and Nguyen 2016) or from our technique. The guarantees in blank cells can be deduced from a more general one.

submodular maximization and feedback rewards for the online vee oracle. Exploiting the underlying properties of DR-submodularity, we show that our algorithm achieves the regret bound of $(1/e, O(T^{3/4}))$. A useful feature of our algorithm is that it is projection-free if using appropriate projection-free oracles.

Over The Hypercube. Restricting on the DR-submodular maximization problem over the hypercube $[0, 1]^n$, a particular down-closed convex set that has been widely studied, our approach leads to an $1/2$-approximation algorithm with regret bound of $O(\sqrt{T}\log T)$. Note that this result can be deduced from (Ene and Nguyen 2016) but as it is not explicitly stated in literature, for completeness, we present the proof in the appendix of the full version.

Maximizing Online Non-monotone DR-submodular Functions over General Convex Sets We go beyond the down-closed structure by considering general convex sets. Building upon our salient ideas from previous algorithm, we prove that for general convex sets, the Meta-Frank-Wolfe algorithm with adapted step-sizes guarantees $\left(\frac{1-\min_{\boldsymbol{x}\in\mathcal{K}}\|\boldsymbol{x}\|_\infty}{3\sqrt{3}}, O(\frac{T}{\ln T})\right)$-regret in expectation. Notably, if the set $\mathcal{K}$ contains $\mathbf{0}$ (for example, $\mathcal{K}$ is the intersection of the semi-definite positive matrix cone and the hypercube $[0, 1]^n$) then the algorithm guarantees in expectation a $(\frac{1}{3\sqrt{3}}, O(\frac{T}{\log T}))$-regret. To the best of our knowledge, this is the *first* constant-approximation algorithm for the problem of online non-monotone DR-submodular maximization over a non-trivial convex set that goes beyond down-closed sets. Note that, any algorithm for the problem over a non-down-closed convex set (in particular arbitrary convex set containing the origin) that guarantees a constant approximation must require in general exponentially many value queries to the function (Vondrák 2013). Remark that the quality of the solution, specifically the approximation

ratio, depends on the initial solution $\boldsymbol{x}_0$. This confirms an observation in various contexts that initialization plays an important role in non-convex optimization.

Related Work

In this section, we give a summary on best-known results on DR-submodular maximization. The domain has been investigated more extensively in recent years due to its numerous applications in the field of statistics and machine learning, for example active learning (Golovin and Krause 2011), viral makerting (Kempe, Kleinberg, and Tardos 2003), network monitoring (Gomez Rodriguez, Leskovec, and Krause 2010), document summarization (Lin and Bilmes 2011), crowd teaching (Singla et al. 2014), feature selection (Elenberg et al. 2018), deep neural networks (Elenberg et al. 2017), diversity models (Djolonga, Tschiatschek, and Krause 2016) and recommender systems (Guillory and Bilmes 2011).

Offline Setting (Bian et al. 2017b) considered the problem of maximizing monotone DR-functions subject to a down-closed convex set and showed that the greedy method proposed by (Calinescu et al. 2011), a variant of well-known Frank-Wolfe algorithm in convex optimization, guarantees a $(1 - 1/e)$-approximation. However, it has been observed by (Hassani, Soltanolkotabi, and Karbasi 2017) that the greedy method is not robust in stochastic settings (where only unbiased estimates of gradients are available). Subsequently, (Mokhtari, Hassani, and Karbasi 2018) proposed a $(1 - 1/e)$-approximation algorithm for maximizing monotone DR-submodular functions over general convex sets in stochastic settings by a new variance reduction technique.

The problem of maximizing non-monotone DR-submodular functions is much harder. (Bian, Buhmann, and Krause 2019) and (Niazadeh, Roughgarden, and Wang 2018) have independently presented algorithms with the same approximation guarantee of $(1/2)$ for the

problem of non-monotone DR-submodular maximization over the hypercube ($\mathcal{K} = [0, 1]^n$). These algorithms are inspired by the bi-greedy algorithms in (Buchbinder et al. 2015; Buchbinder and Feldman 2018). (Bian et al. 2017a) made a further step by providing a $(1/e)$-approximation algorithm where the convex sets are down-closed. (Mokhtari, Hassani, and Karbasi 2018) also presented an algorithm that achieve $(1/e)$ for non-monotone continuous DR-submodular function over a down-closed convex domain that uses only stochastic gradient estimates. Remark that when aiming for approximation algorithms (in polynomial time), the restriction to down-closed polytopes is unavoidable. Specifically, (Vondrák 2013) proved that any algorithm for the problem over a non-down-closed set that guarantees a constant approximation must require in general exponentially many value queries to the function. Recently, (Dürr et al. 2020) presented an algorithm that achieves $((1 - \min_{\boldsymbol{x}\in\mathcal{K}}\|\boldsymbol{x}\|_\infty)/3\sqrt{3})$-approximation for non-monotone DR-submodular function over general convex sets that are not necessarily down-closed.

Online Setting The results for online DR-submodular maximization are known *only* for monotone functions. (Chen, Hassani, and Karbasi 2018) considered the online problem of maximizing monotone DR-submodular functions over general convex sets and provided an algorithm that achieves a $(1 - 1/e, O(\sqrt{T}))$-regret. Subsequently, (Zhang et al. 2019) presented an algorithm that reduces the number of per-function gradient evaluations from $T^{3/2}$ in (Chen, Hassani, and Karbasi 2018) to 1 and achieves the same approximation ratio of $(1 - 1/e)$. Leveraging the idea of one gradient per iteration, they presented a bandit algorithm for maximizing monotone DR-submodular function over a general convex set that achieves in expectation $(1 - 1/e)$ approximation ratio with regret $O(T^{8/9})$. Note that in the discrete setting, (Roughgarden and Wang 2018) studied the non-monotone (discrete) submodular maximization over the hypercube and gave an algorithm which guarantees the tight $(1/2)$-approximation and $O(\sqrt{T})$ regret. Recently and independently, (Niazadeh et al. 2020) have provided an $(1/2, O(\sqrt{T}\log T))$-regret algorithm for DR-submodular maximization over the hypercube. Their algorithm is built on an interesting framework that transform offline greedy algorithms to online ones using Blackwell approachability. Our approach, based on the lifting procedure, is completely different to theirs.

Preliminaries and Notations

Throughout the paper, we use bold face letters, e.g., $\boldsymbol{x}, \boldsymbol{y}$ to represent vectors. For every $S \subseteq \{1, 2, \ldots, n\}$, vector $\mathbf{1}_S$ is the n-dim vector whose i^{th} coordinate equals to 1 if $i \in S$ and 0 otherwise. Given two n-dimensional vectors $\boldsymbol{x}, \boldsymbol{y}$, we say that $\boldsymbol{x} \le \boldsymbol{y}$ iff $x_i \le y_i$ for all $1 \le i \le n$. Additionally, we denote by $\boldsymbol{x} \vee \boldsymbol{y}$, their coordinate-wise maximum vector such that $(x \vee y)_i = \max\{x_i, y_i\}$. Moreover, the symbol $\circ$ represents the element-wise multiplication that is, given two vectors $\boldsymbol{x}, \boldsymbol{y}$, vector $\boldsymbol{x} \circ \boldsymbol{y}$ is the such that the i-th coordinate

$(\boldsymbol{x} \circ \boldsymbol{y})_i = x_i y_i$. The scalar product $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \sum_i x_i y_i$ and the norm $\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x}\rangle^{1/2}$. In the paper, we assume that $\mathcal{K} \subseteq [0, 1]^n$. We say that $\mathcal{K}$ is the *hypercube* if $\mathcal{K} = [0, 1]^n$; $\mathcal{K}$ is *down-closed* if for every $\boldsymbol{z} \in \mathcal{K}$ and $\boldsymbol{y} \le \boldsymbol{z}$ then $\boldsymbol{y} \in \mathcal{K}$; and $\mathcal{K}$ is *general* if $\mathcal{K}$ is a convex subset of $[0, 1]^n$ without any special property.

Definition 1. *A function $F : [0, 1]^n \to \mathbb{R}^+ \cup \{0\}$ is* diminishing-return (DR) submodular *if for all vector $\boldsymbol{x} \ge \boldsymbol{y} \in [0, 1]^n$, any basis vector $\boldsymbol{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ and any constant $\alpha > 0$ such that $\boldsymbol{x} + \alpha\boldsymbol{e}_i \in [0, 1]^n$, $\boldsymbol{y} + \alpha\boldsymbol{e}_i \in [0, 1]^n$, it holds that $F(\boldsymbol{x} + \alpha\boldsymbol{e}_i) - F(\boldsymbol{x}) \le F(\boldsymbol{y} + \alpha\boldsymbol{e}_i) - F(\boldsymbol{y})$. Note that if function F is differentiable then the diminishing-return (DR) inequality is equivalent to $\nabla F(\boldsymbol{x}) \le \nabla F(\boldsymbol{y}) \forall \boldsymbol{x} \ge \boldsymbol{y} \in [0, 1]^n$. Moreover, if F is twice-differentiable then the DR property is equivalent to all of the entries of its Hessian being non-positive, i.e., $\frac{\partial^2 F}{\partial x_i \partial x_j} \le 0$ for all $1 \le i, j \le n$.*

Besides, a differentiable function $F : \mathcal{K} \subseteq [0, 1]^n \to \mathbb{R}^+ \cup \{0\}$ is said to be β-*smooth* if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$, the following holds, $F(\boldsymbol{y}) \le F(\boldsymbol{x}) + \langle \nabla F(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\beta}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$; or equivalently, the gradient is β-Lipschitz, i.e., $\|\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})\| \le \beta\|\boldsymbol{x} - \boldsymbol{y}\|$.

Online DR-Submodular Maximization over Down-Closed Convex Sets

Online Vee Learning

In this section, we give an algorithm for an online problem that will be the main building block in the design of algorithms for online DR-submodular maximization over down-closed convex sets. In the online vee learning problem, given a down-closed convex set $\mathcal{K}$, at every time t, the online algorithm receives $\boldsymbol{c}^t \in \mathcal{K}$ at the beginning of the step and needs to choose a vector $\boldsymbol{x}^t \in \mathcal{K}$. Subsequently, the adversary reveals a vector $\boldsymbol{a}^t \in \mathbb{R}^n$ and the algorithm receives a reward $\langle \boldsymbol{a}^t, \boldsymbol{c}^t \vee \boldsymbol{x}^t\rangle$. The goal is to maximize the total reward $\sum_{t=1}^{T}\langle \boldsymbol{a}^t, \boldsymbol{c}^t \vee \boldsymbol{x}^t\rangle$ over a time horizon T.

The main issue in the online vee problem is that the reward functions are non-concave. In order to overcome this obstacle, we consider a novel approach that consists of discretizing and lifting the corresponding functons to a higher dimensional space.

Discretization and Lifting Let $\mathcal{L}$ be a lattice such that $\mathcal{L} = \{0, \frac{1}{M}, \frac{2}{M}, \ldots, \frac{\ell}{M}, \ldots, 1\}^n$ where $0 \le \ell \le M$ for some parameter M (to be defined later). Note that each $x_i = \frac{\ell}{M}$ for $0 \le \ell \le M$ can be uniquely represented by a vector $\boldsymbol{y}_i \in \{0, 1\}^{M+1}$ such that $y_{i0} = \ldots = y_{i\ell} = 1$ and $y_{i,\ell+1} = \ldots = y_{i,M} = 0$. Based on this observation, we lift the discrete set $\mathcal{K} \cap \mathcal{L}$ to the $(n \times (M+1))$-dim space. Specifically, define a *lifting* map $m : \mathcal{K} \cap \mathcal{L} \to \{0, 1\}^{n \times (M+1)}$ such that each point $(x_1, \ldots, x_n) \in \mathcal{K} \cap \mathcal{L}$ is mapped to an unique point $(y_{10}, \ldots, y_{1M}, \ldots, y_{n0}, \ldots, y_{nM}) \in \{0, 1\}^{n \times M}$ where $y_{i0} = \ldots = y_{i\ell} = 1$ and $y_{i,\ell+1} = \ldots = y_{i,M} = 0$ iff $x_i = \frac{\ell}{M}$ for $0 \le \ell \le M$. Define $\widetilde{\mathcal{K}}$ be the set $\{\mathbf{1}_X \in \{0, 1\}^{n \times (M+1)} : \mathbf{1}_X = m(\boldsymbol{x})$ for some $\boldsymbol{x} \in$

Algorithm 1 Online algorithm for vee learning

Input: A convex set $\mathcal{K}$, a time horizon T.

1: Let $\mathcal{L}$ be a lattice in $[0,1]^n$ and denote the polytope $\mathcal{C} := \texttt{conv}(\widetilde{\mathcal{K}})$.
2: Initialize arbitrarily $\boldsymbol{x}^1 \in \mathcal{K} \cap \mathcal{L}$ and $\boldsymbol{y}^1 = \boldsymbol{1}_{X^1} = m(\boldsymbol{x}^1) \in \widetilde{\mathcal{K}}$.
3: **for** $t = 1$ to T **do**
4:    Observe $\boldsymbol{c}^t$, compute $\boldsymbol{1}_{C^t} = m(\bar{\boldsymbol{c}}^t)$.
5:    Play $\boldsymbol{x}^t$.
6:    Observe $\boldsymbol{a}^t$ (so compute $\widetilde{\boldsymbol{a}}^t$) and receive the reward $\langle \boldsymbol{a}^t, \boldsymbol{c}^t \vee \boldsymbol{x}^t \rangle$.
7:    Set $\boldsymbol{y}^{t+1} \leftarrow \texttt{update}_{\mathcal{C}}(\boldsymbol{y}^t; \widetilde{\boldsymbol{a}}^{t'} \circ (\boldsymbol{1} - \boldsymbol{1}_{C^{t'}}) : t' \leq t)$ where $\circ$ denotes the element-wise multiplication.
8:    Round: $\boldsymbol{1}_{X^{t+1}} \leftarrow \texttt{round}(\boldsymbol{y}^{t+1})$.
9:    Compute $\boldsymbol{x}^{t+1} = m^{-1}(\boldsymbol{1}_{X^{t+1}})$.
10: **end for**

$\mathcal{K} \cap \mathcal{L}$}. Observe that $\widetilde{\mathcal{K}}$ is a subset of discrete points in $\{0,1\}^{n \times (M+1)}$. Let $\mathcal{C} := \texttt{conv}(\widetilde{\mathcal{K}})$ be the convex hull of $\widetilde{\mathcal{K}}$.

Algorithm Description In our algorithm, at every time t, we will output $\boldsymbol{x}^t \in \mathcal{L} \cap \mathcal{K}$. In fact, we will reduce the online vee problem to a online linear optimization problem in the $(n \times (M+1))$-dim space. Given $\boldsymbol{c}^t \in \mathcal{K}$, we round every coordinate c_i^t for $1 \leq i \leq n$ to the largest multiple of $\frac{1}{M}$ which is smaller than c_i^t. In other words, the rounded vector $\bar{\boldsymbol{c}}^t$ has the ith-coordinate $\bar{c}_i^t = \frac{\ell}{M}$ where $\frac{\ell}{M} \leq c_i^t < \frac{\ell+1}{M}$. Vector $\bar{\boldsymbol{c}}^t \in \mathcal{L}$ and also $\bar{\boldsymbol{c}}^t \in \mathcal{K}$ (since $\bar{\boldsymbol{c}}^t \leq \boldsymbol{c}^t$ and $\mathcal{K}$ is down-closed). Denote $\boldsymbol{1}_{C^t} = m(\bar{\boldsymbol{c}}^t)$. Besides, for each vector $\boldsymbol{a}^t \in \mathbb{R}^n$, define its correspondence $\widetilde{\boldsymbol{a}}^t \in \mathbb{R}^{n \times (M+1)}$ such that $\widetilde{a}_{i,j}^t = \frac{1}{M} a_i^t$ for all $1 \leq i \leq n$ and $0 \leq j \leq M$. Observe that $\langle \boldsymbol{a}^t, \bar{\boldsymbol{c}}^t \rangle = \langle \widetilde{\boldsymbol{a}}^t, \boldsymbol{1}_{C^t} \rangle$ (where the second scalar product is taken in the space of dimension $n \times (M+1)$).

The formal description is given in Algorithm 1. In the algorithm, we use a procedure $\texttt{update}$. This procedure takes arguments as the polytope $\mathcal{C}$, the current vector $\boldsymbol{y}^t$, the gradients of previous time steps $\widetilde{\boldsymbol{a}}^t \circ (\boldsymbol{1} - \boldsymbol{1}_{C^t})$ and outputs the next vector $\boldsymbol{y}^{t+1}$. One can use different update strategies, in particular the gradient descent or the follow-the-perturbed-leader algorithm (if aiming for a projection-free algorithm).

$$\boldsymbol{y}^{t+1} \leftarrow \texttt{Proj}_{\mathcal{C}}\left(\boldsymbol{y}^t - \eta \cdot \widetilde{\boldsymbol{a}}^t \circ (\boldsymbol{1} - \boldsymbol{1}_{C^t})\right)$$
(Gradient Descent)

$$\boldsymbol{y}^{t+1} \leftarrow \arg\min_{\mathcal{C}} \left\{ \eta \sum_{t'=1}^{t} \langle \widetilde{\boldsymbol{a}}^{t'} \circ (\boldsymbol{1} - \boldsymbol{1}_{C^{t'}}), \boldsymbol{y} \rangle + \boldsymbol{n}^T \cdot \boldsymbol{y} \right\}$$
(Follow-the-perturbed-leader)

Besides, in the algorithm, we use additionally a procedure $\texttt{round}$ in order to transform a solution $\boldsymbol{y}^t$ in the polytope $\mathcal{C}$ of dimension $n \times (M+1)$ to an integer point in $\widetilde{\mathcal{K}}$. Specifically, given $\boldsymbol{y}^t \in \texttt{conv}(\widetilde{\mathcal{K}})$, we round $\boldsymbol{y}^t$ to $\boldsymbol{1}_{X^t} \in \widetilde{\mathcal{K}}$ using an efficient polynomial-time algorithm given in (Mirrokni et al. 2017) for the approximate Carathéodory's theorem

w.r.t the ℓ_2-norm. Specifically, given $\boldsymbol{y}^t \in \mathcal{C}$ and an arbitrary $\epsilon > 0$, the algorithm in (Mirrokni et al. 2017) returns a set of $k = O(1/\epsilon^2)$ integer points $\boldsymbol{1}_{X_1^t}, \ldots, \boldsymbol{1}_{X_k^t} \in \widetilde{\mathcal{K}}$ in $O(1/\epsilon^2)$-time such that $\|\boldsymbol{y}^t - \frac{1}{k} \sum_{j=1}^{k} \boldsymbol{1}_{X_j^t}\| \leq \epsilon$. Hence, given $\boldsymbol{1}_{X_1^t}, \ldots, \boldsymbol{1}_{X_k^t}$ which have been computed, the procedure $\texttt{round}$ simply consists of rounding $\boldsymbol{y}^t$ to $\boldsymbol{1}_{X_j^t}$ with probability $1/k$. Finally, the solution $\boldsymbol{x}^t$ in the original space is computed as $m^{-1}(\boldsymbol{1}_{X^{t+1}})$.

Lemma 1. *Assume that $\|\boldsymbol{a}^t\| \leq G$ for all $1 \leq t \leq T$ and the update scheme is chosen either as the gradient ascent update or other procedure with regret of $O(\sqrt{T})$. Then, using the lattice $\mathcal{L}$ with parameter $M = (T/n)^{1/4}$ and $\epsilon = 1/\sqrt{T}$ (in the $\texttt{round}$ procedure), Algorithm 1 achieves a regret bound of $O(G(nT)^{3/4})$. The total running time of the algorithm is $O(T^2)$.*

Algorithm for Online Non-monotone DR-submodular Maximization

In this section, we will provide an algorithm for the problem of online non-monotone DR-submodular maximization. The algorithm maintains L online vee learning oracles $\mathcal{E}_1, \ldots, \mathcal{E}_L$. At each time step t, the algorithm starts from $\boldsymbol{0}$ (origin) and executes L steps of the Frank-Wolfe algorithm where the update vector $\boldsymbol{v}_\ell^t$ is constructed by combining the output $\boldsymbol{u}_\ell^t$ of an online vee optimization oracle $\mathcal{E}_\ell$ with the current vector $\boldsymbol{x}_\ell^t$. In particular, $\boldsymbol{v}_\ell^t$ is set to be $\boldsymbol{x}_\ell^t \vee \boldsymbol{u}_\ell^t - \boldsymbol{x}_\ell^t$. Then, the solution $\boldsymbol{x}^t$ is produced at the end of the L-th step. Subsequently, after observing the DR-submodular function F^t, the algorithm defines a vector $\boldsymbol{d}_\ell^t$ and feedbacks $\langle \boldsymbol{d}_\ell^t, \boldsymbol{x}_\ell^t \vee \boldsymbol{u}_\ell^t \rangle$ as the reward function at time t to the oracle $\mathcal{E}_\ell$ for $1 \leq \ell \leq L$. The pseudocode in presented in Algorithm 2.

Theorem 1. *Let $\mathcal{K} \subseteq [0,1]^n$ be a down-closed convex set with diameter D. Assume that functions F^t's are G-Lipschitz, β-smooth and $\mathbb{E}\left[\left\|\tilde{\boldsymbol{g}}_\ell^t - \nabla F^t(\boldsymbol{x}_\ell^t)\right\|^2\right] \leq \sigma^2$ for every t and ℓ. Then, choose $L = T^{3/4}$ and the step-sizes $\eta_\ell = 1/L$ and $\rho_\ell = 2/(\ell+3)^{2/3}$ for all $1 \leq \ell \leq L$, Algorithm 2 achieves the following guarantee:*

$$\sum_{t=1}^{T} \mathbb{E}[F^t(\boldsymbol{x}^t)] \geq \left(\frac{1}{e} - O\left(\frac{1}{L}\right)\right) \max_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} F^t(\boldsymbol{x})$$
$$- O(\beta D + n^{3/4}G + \sigma)DT^{3/4}$$

Maximizing Non-Monotone DR-Submodular Functions over a General Convex Domain

In this section, we consider the problem of maximizing non-monotone continuous DR-submodular functions over a general convex domain. We show that beyond down-closed convex domain, the Meta-Frank-Wolfe algorithm, introduced by (Chen, Hassani, and Karbasi 2018) for monotone DR-submodular functions, provides indeed meaningful guarantees for the problem of online non-monotone DR-submodular maximization over a general convex set. Note that no algorithm with performance guarantee was known in the online setting.

Algorithm 2 Online algorithm for down-closed convex sets

Input: A convex set $\mathcal{K}$, a time horizon T, online vee optimization oracles $\mathcal{E}_1, \ldots, \mathcal{E}_L$, step sizes $\rho_\ell \in (0, 1)$, and $\eta_\ell = 1/L$ for all $\ell \in [L]$

1: Initialize vee optimizing oracle $\mathcal{E}_\ell$ for all $\ell \in \{1, \ldots L\}$.
2: Initialize $\boldsymbol{x}_1^t \leftarrow \boldsymbol{0}$ and $\boldsymbol{d}_0^t \leftarrow \boldsymbol{0}$ for every $1 \leq t \leq T$.
3: **for** $t = 1$ to T **do**
4:     Compute $\boldsymbol{u}_\ell^t \leftarrow$ output of oracle $\mathcal{E}_\ell$ in round t for all $\ell \in \{1, \ldots, L\}$.
5:     **for** $1 \leq \ell \leq L$ **do**
6:       Set $\boldsymbol{v}_\ell^t \leftarrow \boldsymbol{x}_\ell^t \vee \boldsymbol{u}_\ell^t - \boldsymbol{x}_\ell^t$
7:       Set $\boldsymbol{x}_{\ell+1}^t \leftarrow \boldsymbol{x}_\ell^t + \eta_\ell \boldsymbol{v}_\ell^t$.
8:     **end for**
9:     Set $\boldsymbol{x}^t \leftarrow \boldsymbol{x}_{L+1}^t$.
10:    Play $\boldsymbol{x}^t$, observe the function F^t and get the reward of $F^t(\boldsymbol{x}^t)$.
11:    Compute a gradient estimate $\boldsymbol{g}_\ell^t$ such that $\mathbb{E}[\boldsymbol{g}_\ell^t | \boldsymbol{x}_\ell^t] = \nabla F^t(\boldsymbol{x}_\ell^t)$ for all $\ell \in \{1, 2, \ldots, L\}$.
12:    Compute $\boldsymbol{d}_\ell^t = (1 - \rho_\ell)\boldsymbol{d}_{\ell-1}^t + \rho_\ell \boldsymbol{g}_\ell^t$, for every $\ell \in \{1, \ldots, L\}$
13:    Feedback $\langle \boldsymbol{d}_\ell^t, \boldsymbol{x}_\ell^t \vee \boldsymbol{u}_\ell^t \rangle$ to oracle $\mathcal{E}_\ell$ for all $\ell \in \{1, 2, \ldots, L\}$,
14: **end for**

Online Linear Optimization Oracles   In our algorithms, we use multiple online linear optimization oracles to estimate the gradient of online arriving functions. This idea was originally developed for maximizing monotone DR-submodular functions (Chen et al. 2018). Before presenting algorithms, we recall the online linear optimization problems and corresponding oracles. In the online linear optimization problem, at every time $1 \leq t \leq T$, the oracle selects $\boldsymbol{u}^t \in \mathcal{K}$. Subsequently, the adversary reveals a vector $\boldsymbol{d}^t$ and feedbacks the function $\langle \cdot, \boldsymbol{d}^t \rangle$ (and a reward $\langle \boldsymbol{u}^t, \boldsymbol{d}^t \rangle$) to the oracle. The objective is to minimize the regret. There exists several oracles that guarantee sublinear regret, for example the gradient descent algorithm has the regret of $O(n\sqrt{T})$ (see for example (Hazan 2016)).

Algorithm Description   At a high level, at every time t, our algorithm produces a solution $\boldsymbol{x}^t$ by running L steps of the Frank-Wolfe algorithm that uses the outputs of L linear optimization oracles as update vectors. After the algorithm plays $\boldsymbol{x}^t$, it observes stochastic gradient estimates at L points in the convex domain. Subsequently, these estimates are averaged with the estimates from the previous round and are fed to the reward functions of L online linear oracles. The pseudocode is presented in the Algorithm 3.

Theorem 2. *Let $\mathcal{K} \subseteq [0,1]^n$ be a general convex domain with the diameter D. Assume that for every $1 \leq t \leq T$,*
1. *F^t is β-smooth DR-submodular function and the norm of the gradient ∇F^t is bounded by G, i.e., $\|\nabla F(\boldsymbol{x})\| \leq G, \forall \boldsymbol{x} \in \mathcal{K}$,*
2. *the variance of the unbiased stochastic gradients is bounded by σ^2, i.e., $\mathbb{E}\left[\left\|\boldsymbol{g}_\ell^t - \nabla F^t(\boldsymbol{x}_\ell^t)\right\|^2\right] \leq \sigma^2$ for every $1 \leq \ell \leq L$; and*

Algorithm 3 Meta-Frank-Wolfe for general convex domains

Input: A convex set $\mathcal{K}$, a time horizon T, online linear optimization oracles $\mathcal{E}_1, \ldots, \mathcal{E}_L$, step sizes $\rho_\ell \in (0, 1)$, and $\eta_\ell \in (0, 1)$

1: Initialize online linear optimization oracle $\mathcal{E}_\ell$ for all $\ell \in \{1, \ldots L\}$.
2: Initialize $\boldsymbol{x}_1^t \leftarrow \boldsymbol{x}_0$ for some $\boldsymbol{x}_0 \in \mathcal{K}$ and $\boldsymbol{d}_0^t \leftarrow 0$ for every $1 \leq t \leq T$. Let $\boldsymbol{x}_0 \leftarrow \arg\min_{\boldsymbol{x} \in \mathcal{K}} \|\boldsymbol{x}\|_\infty$.
3: **for** $t = 1$ to T **do**
4:     Set $\boldsymbol{v}_\ell^t \leftarrow$ output of oracle $\mathcal{E}_\ell$ in round $t - 1$ for all $\ell \in \{1, \ldots, L\}$.
5:     Set $\boldsymbol{x}_{\ell+1}^t \leftarrow (1 - \eta_\ell)\boldsymbol{x}_\ell^t + \eta_\ell \boldsymbol{v}_\ell^t$.
6:     Play $\boldsymbol{x}^t = \boldsymbol{x}_{L+1}^t$.
7:     Observe $\boldsymbol{g}_\ell^t$ such that $\mathbb{E}\left[\boldsymbol{g}_\ell^t | \boldsymbol{x}_\ell^t\right] = \nabla F^t(\boldsymbol{x}_\ell^t)$.
8:     Set $\boldsymbol{d}_\ell^t \leftarrow (1 - \rho_\ell) \cdot \boldsymbol{d}_{\ell-1}^t + \rho_\ell \cdot \boldsymbol{g}_\ell^t$ for $\ell = \{1, \ldots, L\}$.
9:     Feedback the reward $\langle \boldsymbol{v}_\ell^t, \boldsymbol{d}_\ell^t \rangle$ to $\mathcal{E}_\ell$ for $\ell = \{1, \ldots, L\}$.
10: **end for**

Then setting $L = O(T)$, $\kappa = \frac{\ln 3}{2}$, $\rho_\ell = \frac{2}{(\ell+3)^{2/3}}$ and $\eta_\ell = \frac{\kappa}{\ell H_L}$ for $1 \leq \ell \leq L$ and H_L is the Lth harmonic number, the following inequality holds true for Algorithm 3:

$$\sum_{t=1}^{T} \mathbb{E}\left[F^t(\boldsymbol{x}^t)\right] \geq \left(\frac{1}{3\sqrt{3}}\right)(1 - \|\boldsymbol{x}_0\|_\infty) \max_{\boldsymbol{x}^* \in \mathcal{K}} \sum_{t=1}^{T} F^t(\boldsymbol{x}^*)$$
$$- O\left(\frac{(\beta D + G + \sigma)DT}{\ln L}\right) - O(\sqrt{T}).$$

Remarks

- The regret guarantee, in particular the approximation ratio, depends on the initial solution $\boldsymbol{x}_0$. This confirms the observation that initialization plays an important role in non-convex optimization. For particular case where $\mathcal{K}$ is the intersection of a cone and the hypercube $[0, 1]^n$ (so $\boldsymbol{0} \in \mathcal{K}$), Algorithm 3 provides vanishing $1/(3\sqrt{3})$-regret. Note that this is the first constant approximation for non-monotone DR-submodular maximization over a non-trivial convex domain beyond the class of down-closed convex domains.

- Assume that $\boldsymbol{0} \in \mathcal{K}$ and F^t's are identical, i.e., $F^t = F, \forall t$, the algorithm guarantees a convergence rate of $O\left(1/\log T\right)$. It means that to be ϵ-close to a solution which is $1/(3\sqrt{3})$-approximation to the optimal solution of F, the algorithm requires $T = O(2^{1/\epsilon})$ iterations. Note that the exponential complexity is unavoidable as any contant approximation algorithm for the multilinear extension of a submodular function (so DR-submodular) over a non-down-closed convex set requires necessarily an exponential number of value queries (Vondrák 2013).

Experiments

In this section, we validate our online algorithms on a set of real-world dataset. We show the performance of our algorithms over a down-closed polytope and over a general convex
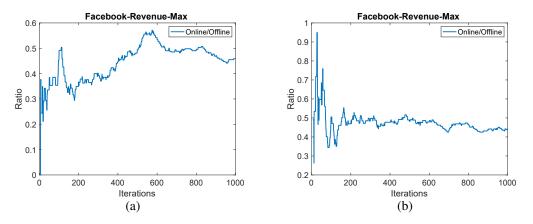
Figure 1: Online revenue maximization over (a) a down-closed polytope; and (b) a general polytope. (Iterations correspond to timesteps $t \in [T]$.)

polytope. All experiments are performed in MATLAB using CPLEX optimization tool on MAC OS version 10.15.

In the revenue maximization problem, the goal of a company is to offer for free or advertise a product to users so that the revenue increases through their "word-of-mouth" effect on others. Here, we are given an undirected social network graph $G = (V, W)$, where $w_{ij} \in W$ represents the weight of the edge between vertex i and vertex j. If the company invests x_i unit of cost on an user $i \in V$ then user i becomes an advocate of the product with probability $1 - (1 - p)^{x_i}$ where $p \in (0, 1)$ is a parameter. Intuitively, this signifies that for investing a unit cost to i, we have an extra chance that the user i becomes an advocate with probability p (Soma and Yoshida 2017).

Let $S \subset V$ be a random set of users who advocate for the product. Then the revenue with respect to S is defined as $\sum_{i \in S} \sum_{j \in V \setminus S} w_{ij}$. Let $F : [0,1]^{|E|} \to \mathbb{R}$ be the expected revenue obtained in this model, that is $F(\boldsymbol{x}) = \mathbb{E}_S[\sum_{i \in S} \sum_{j \in V \setminus S} w_{ij}] = \sum_i \sum_{j:i \neq j} w_{ij}(1 - (1 - p)^{x_i})(1 - p)^{x_j}$. It has been shown that F is a non-monotone DR-submodular function (Soma and Yoshida 2017).

In our setting, we consider the online variant of the revenue maximization on a social network where at time t the weight of an edge is given $w_{ij}^t \in \{0, 1\}$. The experiments are performed on the Facebook dataset that contains 20K users (vertices) and 1M relationships (edges). We choose the number of time steps to be $T = 1000$. At each time $t \in 1, \ldots, T$, we randomly uniformly select 2000 vertices $V^t \subset V$, independently of $V^1, \ldots, V^{t-1}$, and construct a batch B_t with edge-weights $w_{ij}^t = 1$ if and only if $i, j \in V^t$ and edge (i, j) exists in the Facebook dataset. In case if i or j do not belong to V^t, $w_{ij}^t = 0$. We set $p = 0.0001$. In the first experiment, we impose a maximum investment constraint on the problem such that $\sum_{i \in V} x_i \leq 1$. This, in addition to $x_i \geq 0, \forall i \in V$ constitutes a *down-closed* feasible convex set. For the general convex set, we impose an additional minimum investment constraint on the problem such that $\sum_{i \in V} x_i \geq 0.1$.

For comparison purposes, we consider (offline) Frank-Wolfe variants as the benchmark. Variants of Frank-Wolfe algorithm are shown to be competitive for maximizing (offline) non-monotone DR-submodular functions over down-closed convex sets (Bian et al. 2017a) and over general convex sets (Dürr et al. 2020). Moreover, as observed in (Bian et al. 2017a; Dürr et al. 2020), variants of Frank-Wolfe algorithm behave well in practice with approximation ratio better than 0.9 in several experiment settings. For these reasons, we adopt Frank-Wolfe variants as our benchmark.

In Figure 1 we show the ratio of between the objective value achieved by the online algorithm and that of the benchmark over the down-closed convex set (Figure 1(a)) and the general convex set (Figure 1(b)). More specifically, the offline Frank-Wolfe variant performs almost as well as the optimum solution (approximation ratio larger than 0.92) and our online algorithms attain a ratio larger than 0.44 compared with the offline benchmark. Hence, the approximation ratios of the online algorithms is larger than $0.43 * 0.92 > 0.39 > 1/e$.)

Conclusion

In this paper, we study the regret minimization for maximization online non-monotone DR-submodular functions. We presented the first $1/e$-approximation online algorithm with the regret of $O(T^{3/4})$ over down-closed convex sets. Moreover, we presented an online algorithm that achieves an approximation guarantee (depending the search space) for the problem of maximizing non-monotone DR-submodular functions over a *general* convex set. Finally, we run experiments to verify the performance of our algorithms on a social revenue maximization problem on a Facebook user-relationship dataset. The design of an $(1/e, O(\sqrt{T}))$-regret algorithm would be challenging as at each time step, one has to "guess" the right update vector to obtain the approximation ratio of $1/e$ while maintaining a low regret without being trapped into a poor local minima.

Acknowledgments

References

Bian, A. A.; Levy, K.; Krause, A.; and Buhmann, J. M. 2017a. Non-monotone Continuous DR-submodular Maximization: Structure and Algorithms. In *Neural Information Processing Systems (NIPS)*.

Bian, A. A.; Mirzasoleiman, B.; Buhmann, J.; and Krause, A. 2017b. Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains. In *Artificial Intelligence and Statistics*, 111–120.

Bian, Y.; Buhmann, J.; and Krause, A. 2019. Optimal Continuous DR-Submodular Maximization and Applications to Provable Mean Field Inference. In *Proceedings of the 36th International Conference on Machine Learning*, 644–653.

Boyd, S.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Buchbinder, N.; and Feldman, M. 2018. Deterministic algorithms for submodular maximization problems. *ACM Transactions on Algorithms (TALG)* 14(3): 32.

Buchbinder, N.; Feldman, M.; Seffi, J.; and Schwartz, R. 2015. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing* 44(5): 1384–1402.

Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* 40(6): 1740–1766.

Chen, L.; Harshaw, C.; Hassani, H.; and Karbasi, A. 2018. Projection-Free Online Optimization with Stochastic Gradient: From Convexity to Submodularity. In *Proceedings of the 35th International Conference on Machine Learning*, 814–823.

Chen, L.; Hassani, H.; and Karbasi, A. 2018. Online continuous submodular maximization. In *Proc. 21st International Conference on Artificial Intelligence and Statistics (AISTAT)*.

Djolonga, J.; and Krause, A. 2014. From MAP to marginals: Variational inference in bayesian submodular models. In *Advances in Neural Information Processing Systems*, 244–252.

Djolonga, J.; Tschiatschek, S.; and Krause, A. 2016. Variational inference in mixed probabilistic submodular models. In *Advances in Neural Information Processing Systems*, 1759–1767.

Dürr, C.; Thang, N. K.; Srivastav, A.; and Tible, L. 2020. Non-monotone DR-submodular Maximization: Approximation and Regret Guarantees. In *Proc. 29th International Joint Conference on Artificial Intelligence (IJCAI)*.

Elenberg, E.; Dimakis, A. G.; Feldman, M.; and Karbasi, A. 2017. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems*, 4044–4054.

Elenberg, E. R.; Khanna, R.; Dimakis, A. G.; Negahban, S.; et al. 2018. Restricted strong convexity implies weak submodularity. *The Annals of Statistics* 46(6B): 3539–3568.

Ene, A.; and Nguyen, H. L. 2016. A reduction for optimizing lattice submodular functions with diminishing returns. *arXiv:1606.08362* .

Golovin, D.; and Krause, A. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* 42: 427–486.

Gomez Rodriguez, M.; Leskovec, J.; and Krause, A. 2010. Inferring networks of diffusion and influence. In *Proc. 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1019–1028.

Guillory, A.; and Bilmes, J. A. 2011. Simultaneous Learning and Covering with Adversarial Noise. In *ICML*, volume 11, 369–376.

Hassani, H.; Soltanolkotabi, M.; and Karbasi, A. 2017. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, 5841–5851.

Hazan, E. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2(3-4): 157–325.

Ito, S.; and Fujimaki, R. 2016. Large-scale price optimization via network flow. In *Advances in Neural Information Processing Systems*, 3855–3863.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proc. 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146.

Kulesza, A.; Taskar, B.; et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5(2–3): 123–286.

Lin, H.; and Bilmes, J. 2011. A class of submodular functions for document summarization. In *Proc. 49th Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 510–520.

Mirrokni, V.; Leme, R. P.; Vladu, A.; and Wong, S. C.-w. 2017. Tight bounds for approximate Carathéodory and beyond. In *Proc. 34th International Conference on Machine Learning*, 2440–2448.

Mokhtari, A.; Hassani, H.; and Karbasi, A. 2018. Conditional Gradient Method for Stochastic Submodular Maximization: Closing the Gap. In *Procs of the International Conference on Artificial Intelligence and Statistics*, volume 84, 1886–1895.

Niazadeh, R.; Golrezaei, N.; Wang, J.; Susan, F.; and Badanidiyuru, A. 2020. Online learning via offline greedy: Applications in market design and optimization. *Available at SSRN 3613756* .

Niazadeh, R.; Roughgarden, T.; and Wang, J. R. 2018. Optimal Algorithms for Continuous Non-monotone Submodular and DR-Submodular Maximization. In *Neural Information Processing Systems (NIPS)*.

Roughgarden, T.; and Wang, J. R. 2018. An Optimal Learning Algorithm for Online Unconstrained Submodular Maximization. In *Conference On Learning Theory*, 1307–1325.

Singla, A.; Bogunovic, I.; Bartók, G.; Karbasi, A.; and Krause, A. 2014. Near-Optimally Teaching the Crowd to Classify. In *ICML*, 154–162.

Soma, T.; and Yoshida, Y. 2017. Non-Monotone DR-Submodular Function Maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Staib, M.; and Jegelka, S. 2017. Robust Budget Allocation via Continuous Submodular Functions. In *International Conference on Machine Learning*, 3230–3240.

Streeter, M.; and Golovin, D. 2009. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, 1577–1584.

Vondrák, J. 2013. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing* 42(1): 265–304.

Zhang, M.; Chen, L.; Hassani, H.; and Karbasi, A. 2019. Online Continuous Submodular Maximization: From Full-Information to Bandit Feedback. In *Advances in Neural Information Processing Systems*, 9206–9217.