The Diminishing Returns (DR) Property and Its Applications in Machine Learning

Omid Sadeghi

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Maryam Fazel, Chair

Jeffrey A. Bilmes

Kevin Jamieson

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

The Diminishing Returns (DR) Property and Its Applications in Machine Learning

Omid Sadeghi

Chair of the Supervisory Committee:
Maryam Fazel
Department of Electrical and Computer Engineering

Numerous tasks in machine learning involve objective functions that exhibit a *Diminishing Returns (DR)* property, i.e., the marginal gain of increasing the input gets smaller as the input gets larger. For instance, in document summarization, the goal is to select a small subset of sentences that represent the entirety of the document. As the summary gets larger, the additional benefit of adding a sentence to the summary diminishes. In this dissertation, we focus on the class of set functions and continuous functions that exhibit the DR property. These functions are called submodular set functions and continuous DR-submodular functions respectively. This dissertation presents several contributions to various online and offline maximization problems in machine learning where the utility functions satisfy the DR property, with the main themes organized into three parts: (i) study of online DR-submodular maximization under online budget constraints and designing primal-dual algorithms that not only perform well in terms of the utility, but they also satisfy the online constraints; (ii) characterization of the class of strongly DR-submodular functions and providing techniques for offline and online maximization of these functions that utilize the additional structure to obtain improved convergence rates and regret guarantees respectively; and (iii) study of offline and online submodular set function maximization problems under social and economic considerations (e.g., privacy and strategic behavior) and designing differentially private and incentive-compatible algorithms for these problems.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Maryam Fazel, for her unwavering support and guidance throughout my doctoral journey. She allowed me to explore and choose projects that I am truly passionate about. Her encouragement and constructive feedback have been invaluable in shaping my research and professional development. I am grateful for her patience and understanding during the challenging times and for always believing in me. I would also like to thank all my committee members for carefully examining my work.

I am thankful to Professors Jeffrey A. Bilmes and Lillian J. Ratliff whom I collaborated with during my Ph.D. studies. I am also grateful to fellow UW graduate students Prasanna Raut, Mitas Ray, and Adhyyan Narang whom I worked with and mentored.

Playing sports helped me maintain my physical and mental health throughout my doctoral journey. I would like to thank everyone in Seattle Green Lake Running Group and all my UW soccer teammates for providing a much-needed break from my academic work.

I am grateful to all my friends here in Seattle and around the world. In particular, I want to thank Moein, Navid, and Mohsen for being there for me during the tough times.

Last but not least, I would like to thank my family (my brother Ali and my parents) back home in Iran; I could not have done this without your support.

DEDICATION

to my family.

Chapter 1

INTRODUCTION

The Diminishing Returns (DR) property is mostly associated with submodular set functions. A set function F over the ground set V is called submodular if for all sets $A \subset B \subset V$ and $j \in V$, we have:

$$F(A \cup \{j\}) - F(A) \geq F(B \cup \{j\}) - F(B).$$

i.e., as the set gets larger, the marginal gain of adding an item j diminishes. Submodular set functions are often used to quantify coverage, relevance, and diversity in various machine-learning applications. In particular, it is well-known that the rank function of matroids, cut functions of graphs and hypergraphs, entropy, mutual information, and covering functions are all examples of submodular set functions. The DR property enables efficient minimization and approximate maximization of submodular set functions.

Moving beyond set functions, it is interesting to study continuous functions that exhibit the DR property for two reasons: i) It is an important modeling ingredient for many real-world applications, and ii) it captures a class of well-behaved non-convex optimization problems that admits optimization algorithms with polynomial running time. Continuous functions with the DR property are called continuous DR-submodular.

In this dissertation, we focus on maximizing discrete and continuous functions with the DR property. In particular, we study offline and online optimization problems where in addition to maximizing a utility function with the DR property, we are constrained by various other considerations that arise in machine learning applications.

1.1 Outline

This dissertation is based on most of the projects completed during my Ph.D. studies, all related to the central theme of (DR) submodular maximization. In Chapter 2, we set up the notation and introduce the necessary background materials. Each of the subsequent chapters is based on one of my first-author publications with minor revisions to increase the cohesion of the dissertation. The chapters are categorized into three parts described below.

Online budget-constrained DR-submodular maximization. We first study the online DR-submodular maximization problem subject to linear packing constraints in Chapter 3 which is a generalization of online packing problems by allowing the utility functions to be non-convex and DR-submodular. We design an algorithm for this problem and provide its competitive ratio guarantees. Next, in chapters 4-6, we focus on a class of online DR-submodular maximization problems where in addition to maximizing the utility, the sequence of decisions must satisfy some linear or convex constraints on average. We study this problem under an adversarial and a stochastic model for the online linear constraints in Chapter 4 and Chapter 5 respectively, and we design algorithms with sublinear regret as well as sublinear constraint violation guarantees for each setting. In Chapter 6, we consider the more general setting with online convex constraints and provide a single best-of-both-worlds algorithm with improved performance guarantees for both settings without prior knowledge of the regime.

Strongly DR-submodular maximization. In Chapter 7, we first introduce and characterize the class of strongly DR-submodular functions. Then, we focus on offline strongly DR-submodular maximization problems and show how this additional structure in the objective function could be utilized to obtain faster convergence rates for the problem. Next, in Chapter 8, we turn our focus on online strongly DR-submodular maximization problems, and design algorithms for various stochastic and adversarial online input models with improved regret guarantees compared to the setting with DR-submodular utilities.

Submodular maximization under social and economic considerations. Finally, we turn our attention to offline and online submodular set function maximization problems under additional social and economic considerations. In Chapter 9, we consider submodular set function maximization problems in machine learning involving individuals with sensitive data in the dataset and design offline and online differentially private algorithms for the problem. Next, in Chapter 10, we study a generalization of the well-known online binary prediction with expert advice framework where at each round, the learner has to pick a subset of experts (instead of just one expert) and the utility is a (sub)modular function of the selected experts. We assume that experts act strategically and may misreport their true beliefs to maximize their chances of being picked by the learner. We design (approximately) incentive-compatible algorithms for the problem.

1.2 Publications

This dissertation is based on multiple first-author conference and workshop publications listed below:

- Omid Sadeghi, Reza Eghbali, and Maryam Fazel. Online Algorithms for Budget-Constrained DR-Submodular Maximization. In *International Conference on Machine Learning (ICML) Workshop on Negative Dependence and Submodularity for ML (NDSML)*, 2020.

- Omid Sadeghi and Maryam Fazel. Online Continuous DR-Submodular Maximization with Long-Term Budget Constraints. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

- Omid Sadeghi, Prasanna Raut, and Maryam Fazel. A Single Recipe for Online Submodular Maximization with Adversarial or Stochastic Constraints. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

- Omid Sadeghi, Prasanna Raut, and Maryam Fazel. Online DR-Submodular Maximization: Minimizing Regret and Constraint Violation. In *AAAI Conference on Artificial Intelligence*, 2021.

- Omid Sadeghi and Maryam Fazel. Differentially Private Monotone Submodular Maximization Under Matroid and Knapsack Constraints. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

- Omid Sadeghi and Maryam Fazel. Improved Regret Bounds for Online Submodular Maximization. In *International Conference on Machine Learning (ICML) Workshop on Subset Selection in Machine Learning: From Theory to Applications*, 2021.

- Omid Sadeghi and Maryam Fazel. Fast First-Order Methods for Monotone Strongly DR-Submodular Maximization. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA)*, 2023.

- Omid Sadeghi and Maryam Fazel. No-Regret Online Prediction with Strategic Experts. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

There are two other papers that I worked on during my Ph.D. studies that I do not discuss here. They are listed below:

- Mitas Ray*, Omid Sadeghi*, Lillian J. Ratliff, and Maryam Fazel. Function Design for Improved Competitive Ratio in Online Resource Allocation with Procurement Costs. *Under Review.*

- Adhyyan Narang, Omid Sadeghi, Lillian J. Ratliff, Maryam Fazel, and Jeff Bilmes. Online SuBmodular + SuPermodular (BP) Maximization with Bandit Feedback. *Under Submission.*

Chapter 2

PRELIMINARIES

2.1 Notation

We use $[m]$ to denote the set $\{1,2,\dots,m\}$. For a matrix $A \in \mathbb{R}^{n\times m}$, we denote its (i,j)-th entry by $A_{i,j}$, its i-th row by $\hat{a}_i^T$ for all $i \in [n]$, and its j-th column by a_j for all $j \in [m]$. A^T denotes the transpose of A. The inner product of two vectors $x, y \in \mathbb{R}^m$ is denoted by $\langle x, y\rangle$ or $x^T y$. For two vectors $x, y \in \mathbb{R}^m$, $x \preceq y$ means $x_i \le y_i \ \forall i \in [m]$. A continuous function $f : \mathbb{R}^m \to \mathbb{R}$ is called monotone if for all x, y such that $x \preceq y$, $f(x) \le f(y)$ holds. We use f^* to denote the concave conjugate of a function $f : \mathbb{R}^m \to \mathbb{R}$, defined as $f^*(y) = \inf_x \big(\langle x, y\rangle - f(x)\big)$. For a vector $x \in \mathbb{R}^n$, we use $\|x\|$ to denote the Euclidean norm of x. For a convex set $\mathcal{X}$, we will use $\mathrm{Proj}_{\mathcal{X}}(y)$ or $\mathcal{P}_{\mathcal{X}}(y)$ to denote the projection of y onto set $\mathcal{X}$, i.e., $\arg\min_{x\in\mathcal{X}} \|x - y\|$. For a convex set $\mathcal{P}$, the support function of $\mathcal{P}$ is defined as $\sigma_{\mathcal{P}}(x) = \sup_{y\in\mathcal{P}} \langle x, y\rangle$. For a set function F, we use $F(j|A)$ to denote $F(A \cup \{j\}) - F(A)$. A set function $F : 2^V \to \mathbb{R}$ is called monotone if for all S, S' such that $S \subseteq S'$, $F(S) \le F(S')$ holds. The dual norm $\|\cdot\|_*$ of a norm $\|\cdot\|$ is defined as $\|y\|_* = \max_{x:\|x\|\le 1} \langle y, x\rangle$. A differentiable function $f : \mathcal{X} \to \mathbb{R}$ is β-Lipschitz if for all $x, y \in \mathcal{X}$, we have $|f(y) - f(x)| \le \beta\|y - x\|$, or equivalently, $\|\nabla f(x)\| \le \beta$ holds. The diameter of a set $\mathcal{X}$ is defined as $\max_{x,y\in\mathcal{X}} \|y - x\|$. We use $x \vee y = \max\{x, y\}$ and $x \wedge y = \min\{x, y\}$ to denote the component-wise maximum and minimum of $x, y \in \mathbb{R}^n$, i.e., for all $i \in [n]$, $[x \vee y]_i = \max\{x_i, y_i\}$ and $[x \wedge y]_i = \min\{x_i, y_i\}$ holds. For $u \in \mathbb{R}$, we define $[u]_+ := \max\{u, 0\}$.

2.2 Submodular set functions

Definition 2.2.1 *A set function F over the ground set V is submodular if for all $j \in V$ and $A \subseteq B \subseteq V \setminus \{j\}$, the following holds:*

$$F(A \cup \{j\}) - F(A) \geq F(B \cup \{j\}) - F(B).$$

In other words, the gain of adding a particular element j to a set decreases as the set gets larger. This property is known as the discrete Diminishing Returns (DR) property.

For a non-negative normalized monotone non-decreasing submodular function F over the ground set V, total curvature is defined as [68]:

$$c_F = 1 - \min_{S \subset V \setminus \{j\}, F(j) \neq 0} \frac{F(j|S)}{F(j)} = 1 - \min_{j : F(j) \neq 0} \frac{F(j|(V \setminus j))}{F(j)}.$$

It is easy to see that $c_F \in [0, 1]$ always holds. $c_F \leq 1$ is due to monotonicity of F and $c_F \geq 0$ follows from F being submodular. Curvature characterizes how submodular the function is. If $c_F = 0$, the function is modular, and larger values of c_F correspond to the function exhibiting a stronger diminishing returns structure.

2.3 Continuous DR-submodular functions

Definition 2.3.1 *A differentiable function $f : K \to \mathbb{R}$, $K \subset \mathbb{R}^m_+$, is called DR-submodular if:*

$$x \succeq y \Rightarrow \nabla f(x) \preceq \nabla f(y).$$

In other words, ∇f is an anti-tone mapping from $\mathbb{R}^m$ to $\mathbb{R}^m$. This property is known as the continuous Diminishing Returns (DR) property.

If f is twice differentiable, DR-submodularity is equivalent to the Hessian matrix being element-wise non-positive. Note that for $m = 1$, DR-submodularity is equivalent to concavity. However, for $m > 1$, concavity implies negative semi-definiteness of the Hessian matrix which is not equivalent to the Hessian matrix being element-wise non-positive. A similar property is introduced in [201] as well and functions satisfying this property are called "smooth submodular". Additionally, [79] defined the DR property for concave functions with respect to a partial ordering induced by a cone and showed that by taking the cone to be $\mathbb{R}^m_+$, Definition 2.3.1 is recovered and if the cone of positive semi-definite matrices is considered, the DR property generalizes to matrix ordering as well [80]. [25] showed that DR-submodular functions are concave along any non-negative direction, and any non-positive direction. In other words, for a DR-submodular function f, if $t \geq 0$ and $v \in \mathbb{R}^m$ satisfies $v \succeq 0$ or $v \preceq 0$, we have:

$$f(x + tv) \leq f(x) + t\langle \nabla f(x), v\rangle.$$

For a DR-submodular function f, we say that f is L-smooth over non-negative directions with respect to the norm $\|\cdot\|$ if

$$f(y) - f(x) \geq \langle \nabla f(x), y - x\rangle - \frac{L}{2}\|y - x\|^2 \quad \forall x, y; x \preceq y.$$

2.4 Examples of continuous DR-submodular functions

- **Multilinear extension of submodular set functions [45].** The multilinear extension $f : [0, 1]^V \to \mathbb{R}$ of a submodular set function F over the ground set V is defined as:

$$f(x) = \sum_{S \subset V} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) = \mathbb{E}_{S \sim x}[F(S)].$$

Multilinear extensions are extensively used for maximizing the corresponding submodular set function and are known to be a special case of non-concave DR-submodular functions. The Hessian matrix of this class of functions has non-positive off-diagonal entries with zeros on its diagonal. [110] showed that for a large class of submodular set functions, their multilinear extension could be efficiently computed. Weighted matroid rank functions, set cover functions, probabilistic coverage functions, graph cut functions, and concave over modular functions are all examples of such submodular functions (see [110, 26] for more examples and details).

- **Indefinite quadratic functions.** Let $f(x) = \frac{1}{2}x^T Ax + a^T x + c$ where A is a symmetric matrix. If A is entry-wise non-positive, f is a DR-submodular function. Such quadratic utility functions have a wide range of applications. In particular, price optimization with continuous prices [109] and computing stability number of graphs [155] are both non-concave DR-submodular quadratic optimization problems.

- **Concave functions with negative dependence.** Let $d \geq 2$. If $h_i : \mathbb{R}_+ \to \mathbb{R}$ is concave for all $i \in [n]$ and $\theta_{i_1,\ldots,i_r} \leq 0$ for all $r \in [d]$ and $(i_1, \ldots, i_r) \subseteq [n]$, the following function $f : \mathbb{R}_+^n \to \mathbb{R}$ is DR-submodular:

$$f(x) = \sum_{i=1}^{n} h_i(x_i) + \sum_{(i,j):i \neq j} \theta_{ij} x_i x_j + \cdots + \sum_{(i_1,\ldots,i_d):i_r \neq i_s \ \forall r,s \in [d]} \theta_{i_1,\ldots,i_d} x_{i_1} \ldots x_{i_d}.$$

- **Log-determinant function.** Let the function $f : [0,1]^n \to \mathbb{R}$ be defined as

$$f(x) = \log\det\big(\operatorname{diag}(x)(L - I) + I\big),$$

where $L \succeq 0$ is a positive semidefinite matrix and $\operatorname{diag}(x)$ denotes a diagonal matrix with vector x on its diagonal. This function is used as the objective function in Determinantal Point Processes (DPPs). It was proved in [93] that f is a DR-submodular function.

Many other classes of DR-submodular functions were not discussed above (e.g., nested concave functions used for continuous deep submodular functions [75, 28]). See [23, 25, 27] for more examples.

Part I

ONLINE BUDGET-CONSTRAINED DR-SUBMODULAR MAXIMIZATION

Chapter 3

COMPETITIVE ALGORITHMS FOR ONLINE BUDGET-CONSTRAINED DR-SUBMODULAR MAXIMIZATION

3.1 Chapter overview

This chapter presents joint work with Reza Eghbali and Maryam Fazel. A shorter version of this chapter appeared in the International Conference on Machine Learning (ICML) 2020 Workshop on Negative Dependence and Submodularity for ML (NDSML). In this chapter, we study a certain class of online optimization problems, where the goal is to maximize a non-concave and DR-submodular function under budget constraints. We introduce a primal-dual algorithm, called the generalized sequential algorithm, and we obtain the first bound on the competitive ratio of online monotone DR-submodular function maximization subject to linear packing constraints which match the known tight bound in the special case of the linear objective function.

3.2 Introduction

Online optimization covers a large number of problems including online resource allocation, online bipartite matching [117], the "AdWords" problem [147, 42], online submodular welfare maximization [133], online linear programming [43], and online concave packing problem [11, 79]. One type of algorithm proposed for solving such problems is primal-dual algorithms where the dual variable is updated at each step and is used to get the update rule for the primal variable [44].

Depending on how much information about the online input is available in advance to the algorithm, online problems have been categorized into adversarial (worst-case) (e.g., in [147]) and stochastic input models (e.g., in [122]) and we consider the former in this chapter. In the adversarial model, it is assumed that the algorithm does not know the online input.

The performance of online algorithms is measured by their competitive ratio defined as the ratio of the value of the objective function at the output of the algorithm to the maximum objective value attained offline. In the worst-case model, one is interested in deriving lower bounds on the competitive ratio of the algorithm that holds for all arbitrary online inputs. There exists a significant amount of work on online adversarial packing problems in the literature. [43] studied online packing problems with linear objective functions, and [13, 11, 36, 54, 79, 71] considered the more general setting with concave utility functions and obtained competitive ratio bounds in these settings. All the existing works focus on online concave packing problems. In contrast, in this chapter, we discuss a certain class of online packing problems where the objective function is not necessarily concave and is only assumed to be DR-submodular. We use techniques from the submodular set function maximization literature to introduce a greedy primal-dual algorithm, called the generalized sequential algorithm, and we analyze its performance theoretically and numerically under the worst-case input model. Specifically, we make the following contributions:

- We introduce online DR-submodular maximization subject to linear packing constraints which generalize online packing problems by allowing the utility functions to be generally non-concave and DR-submodular rather than concave, and give examples of various online discrete budget-constrained submodular problems whose continuous generalization could be cast in this framework, for example, the generalized continuous version of online knapsack-constrained monotone submodular function maximization [140] and online submodular welfare maximization [133] are well-known cases. Therefore, our framework unifies two existing separate works of literature on online packing problems and submodular maximization into a single rich model.
- We introduce the generalized sequential algorithm (Section 3.4) to solve this class of problems. The generalized sequential algorithm reduces to the sequential algorithm in [79] for a certain choice of parameters, however, the sequential algorithm in [79] could merely be used for concave utility functions with the DR property whereas our generalized sequential algorithm could be exploited for non-concave problems as well. Additionally, our algorithm

could be interpreted as the online counterpart of the Frank-Wolfe variant proposed in [25] for solving offline constrained continuous DR-submodular maximization problems. Denoting the number of linear packing constraints by n, we consider two cases $n > 1$ and $n = 1$ separately and by designing problem-tailored penalty functions for each case to enforce the packing constraints, we derive competitive ratio bounds which are optimal in the special case of linear utility functions (see Section 3.3 for more details about the differences of these two cases):
$\mathbf{n > 1}$: In this case, our problem generalizes the AdWords problem [147] and the online linear programming problem [43] by allowing the utility function to be DR-submodular rather than linear. For this setting, we obtain the first competitive ratio bound in Theorem 3.4.1 which is optimal in the special cases. Specifically, if the objective function is linear, our problem reduces to online linear programming and the algorithm achieves the optimal competitive ratio [79, 43]. If in addition to the linearity of the objective function, the coefficients in the linear packing constraint and the objective function are equal, the problem simplifies to the AdWords problem and we obtain the optimal $1 - \frac{1}{e}$ competitive ratio [147, 42] (note that since we allow fractional assignments for the AdWords problem, we do not need the small bids assumption to obtain the optimal competitive ratio).
$\mathbf{n = 1}$: In this case, our problem is the generalization of online knapsack-constrained monotone submodular function maximization [140] to the continuous setting. For this online problem, we obtain a competitive ratio bound of $\frac{1}{1-\alpha+\ln(\frac{U}{L})}$ in Theorem 3.4.2 where L and U are lower and upper bounds on the value-to-weight ratio of the items respectively and α captures the curvature of the DR-submodular utility function (L, U and α are defined in Section 3.4). For discrete online knapsack-constrained submodular maximization, [140] obtained a $\frac{1}{(1+\kappa_f)(1+ln(\frac{U}{L}))}$ competitive ratio bound where κ_f is the total curvature of the submodular utility function f [68]. If we apply our generalized sequential algorithm to the multilinear extension of the function f (which satisfies the DR property and we denote it by F), we obtain the competitive ratio bound $\frac{1}{1-\alpha_F+\ln(\frac{U}{L})}$ and because we have showed in Remark 3.4.1 that $\alpha_F \geq -\kappa_f$ holds, our bound improves upon the result of [140]. Additionally, if the utility function is linear, we obtain the competitive ratio bound of $\frac{1}{1+\ln(\frac{U}{L})}$ which is

provably optimal [217].

Finally, we present numerical experiments in Section 3.5 on a class of non-concave DR-submodular utility functions to demonstrate the performance of the generalized sequential algorithm.

Note that although our framework can be interpreted as the generalization of online budgeted discrete submodular problems to the continuous setting, we do not aim to solve the discrete problem itself. In other words, our goal is to solve a class of online budgeted problems where the objective function is originally continuous and DR-submodular. Therefore, we do not round the fractional output of our proposed algorithm.

3.3 Problem statement

The offline constrained optimization problem is as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} H_i(\hat{x}_i) \\ \text{subject to} & x_t \in F_t \quad \forall t \in [m] \\ & \hat{c}_i^T \hat{x}_i \leq 1 \quad \forall i \in [n], \end{array} \tag{3.1}$$

where $\hat{x}_i^T \in \mathbb{R}_+^m$ is the i-th row and $x_t \in \mathbb{R}_+^n$ is the t-th column of the variable matrix $X \in \mathbb{R}_+^{n\times m}$, $\hat{c}_i \in \mathbb{R}_+^m$ and $c_t \in \mathbb{R}_+^n$ are the i-th row and t-th column of the cost matrix $C \in \mathbb{R}_+^{n\times m}$ respectively, and $F_t \subseteq \mathbb{R}_+^n$. For all $i \in [n]$, $H_i : K \to \mathbb{R}$, $K \subset \mathbb{R}_+^m$, is a differentiable monotone non-decreasing DR-submodular function which is zero at the origin (i.e., $H_i(0) = 0$). For all $t \in [m]$, F_t is a compact convex constraint set that contains the origin and $\|x\|_2 \leq \lambda$ for all $x \in F_t$.

In the online setting, at step $t \in [m]$, c_t and F_t arrive online and the algorithm should choose $x_t \in F_t$ to maximize the overall objective function. Note that at each step $t \in [m]$, the function H_i $\forall i \in [n]$ is only known over subsets of variables (of size t) that have already arrived. Thus, we do not have access to the objective function in advance.

The penalized formulation of problem (3.1) is the following:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} \left(H_i(\hat{x}_i) + G_i(\hat{c}_i^T \hat{x}_i)\right) \\ \text{subject to} & x_t \in F_t \subseteq \mathbb{R}_+^n \ \forall t \in [m] \end{array}. \tag{3.2}$$

As an example, if for all $i \in [n]$,

$$G_i(u) = \begin{cases} 0 & \text{if } 0 \le u \le 1 \\ -\infty & \text{if } u > 1 \end{cases},$$

i.e., the concave indicator function of the interval $[0, 1]$, the above two optimization problems are equivalent.

We aim to design differentiable, concave, and monotone non-increasing penalty functions $G_i : \mathbb{R}_+ \to \mathbb{R}\ \forall i \in [n]$ and use them in our online algorithm such that the output does not violate any of the linear packing constraints.

Our framework with $n = 1$ corresponds to the case where at each step $t \in [m]$, item t with cost c_t arrives online and the algorithm should decide what fraction $x_t \in [0, 1]$ of the item should be taken to maximize the utility function while satisfying the overall budget constraint. This case is the continuous generalization of the online submodular knapsack problem [217, 140]. The case $n > 1$ considers the online problem where n agents with separate utilities and budget constraints are available offline and at each step $t \in [m]$, item t arrives online, and agent $i \in [n]$ bids c_{it} for this item. The algorithm should decide the fractional allocation of the item among the agents to maximize the overall utilities of all the agents while all the budget constraints are satisfied as well. This case generalizes the AdWords problem [147], single-unit combinatorial auction problem [108], online linear programming [43] and online concave packing problem [11, 79] by allowing the utility functions to be non-concave and DR-submodular. The existing algorithms and competitive ratio bounds for these two cases are quite different.

Multiple applications of this framework are provided in Appendix A.1.

3.3.1 Dual problem

The dual problem of the constrained problem (3.1) is as follows (see Appendix A.2 for the derivation):

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{m} \sigma_{F_t}\Big(\begin{bmatrix} y_{1t} - z_1 c_{1t} \\ \vdots \\ y_{nt} - z_n c_{nt} \end{bmatrix}\Big) - \sum_{i=1}^{n} H_i^*(\hat{y}_i) + \sum_{i=1}^{n} z_i \\ \text{subject to} \quad & z_i \geq 0 \ \forall i \in [n] \end{aligned} , \tag{3.3}$$

where for all $i \in [n]$, $z_i \in \mathbb{R}_+$, $\hat{y}_i^T \in \mathbb{R}^m$ is the i-th row of the dual matrix variable $Y \in \mathbb{R}^{n \times m}$ and y_{it} is the (i,t)-th entry of this matrix.

Karush–Kuhn–Tucker (KKT) conditions for the penalized problem (3.2) can be written as:

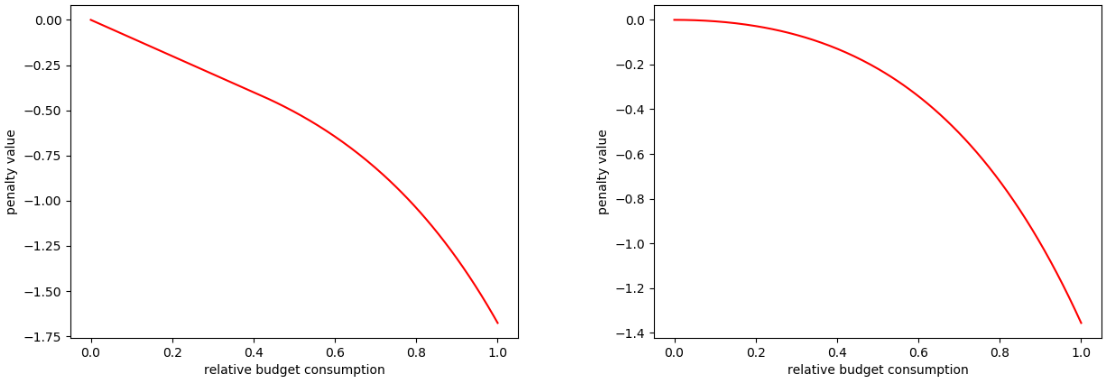
$$\begin{aligned} x_t^* &\in \underset{x \in F_t}{\arg\max} \ \langle x, \begin{bmatrix} y_{1t}^* - z_1^* c_{1t} \\ \vdots \\ y_{nt}^* - z_n^* c_{nt} \end{bmatrix} \rangle, \\ \hat{y}_i^* &= \nabla H_i(\hat{x}_i^*) \ i = 1, \ldots, n, \\ z_i^* &= -G_i'(\hat{c}_i^T \hat{x}_i^*) \ i = 1, \ldots, n, \end{aligned}$$

where G_i' is the derivative of the scalar penalty function G_i. We remind the reader that we aim to design differentiable penalty functions G_i $\forall i \in [n]$ and therefore, we have used G_i' in the KKT conditions.

We will use these KKT conditions to design the generalized sequential algorithm.

3.4 Generalized sequential algorithm and competitive ratio analysis

In this section, we first introduce our design for the penalty functions in Section 3.4.1 and then, propose our algorithm, called the generalized sequential algorithm, in Section 3.4.2.



(a) $n = 1$, $L = 1$ and U=4

(b) $n > 1$, $L = 1$ and U=4

Figure 3.1: Designed penalty functions.

3.4.1 Design of the penalty functions

If $n = 1$, we design the penalty function G_1 as follows:

$$G_1(u) = \begin{cases} -L_1 u & \text{if } 0 \le u < \frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \\ -\frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \frac{L_1}{e} (\frac{U_1 e}{L_1})^u & \text{if } u \ge \frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)} \end{cases}.$$

If $n > 1$, for all $i \in [n]$, we construct the penalty function G_i as below:

$$G_i(u) = \frac{L_i}{(e-1)\ln(1 + \frac{U_i(e-1)}{L_i})}\big(1 - (1 + \frac{U_i(e-1)}{L_i})^u\big) + \frac{L_i}{e-1}u,$$

where for all $i \in [n]$, U_i and L_i are defined as follows:

$$U_i = \max_{t \in [m]} \frac{\sup_{x \in \mathbb{R}^m:\ \hat{c}_i^T x = 1} \nabla_t H_i(x)}{c_{it}},$$

$$L_i = \min_{t \in [m]} \frac{\inf_{x \in \mathbb{R}^m:\ \hat{c}_i^T x \le 1} \nabla_t H_i(x)}{c_{it}}.$$

Roughly speaking, U_i and L_i are upper and lower bounds for the value-to-weight ratio of the items for the i-th agent respectively. We are assuming that these upper and lower bounds

Algorithm 1 Generalized sequential algorithm

```
**Input**: Penalty functions $G_i$ $\forall i \in [n]$, $K$ and $m$.
Initialize $\tilde{X} = 0$.
**for** $t = 1$ **to** $m$ **do**
    $c_t, F_t$ arrive online and gradient of $H_i$ $\forall i \in [n]$ over the first $t$ variables (i.e., all other $m - t$ variables being zero) is accessible.
    $\tilde{x}_t(0) = 0$.
    **for** $k = 1$ **to** $K$ **do**
        $v_t(k) = \arg\max_{v \in F_t} \langle v, d_t(k-1) \rangle$.        $\{d_t(k-1)$ defined in equation (3.4)$\}$
        $\tilde{x}_t(k) = \tilde{x}_t(k-1) + \frac{1}{K} v_t(k)$.
    **end for**
    **Output:** $\tilde{x}_t = \tilde{x}_t(K)$.
**end for**
```

are available offline to design the penalty functions. The penalty functions for $L = 1$ and $U = 4$ are plotted for both cases $n = 1$ and $n > 1$ in figures 3.1a and 3.1b respectively.

Our design for the penalty function for $n = 1$ is inspired by the threshold function proposed by [217]. In the $n > 1$ case, our penalty function is inspired by the allocation rule of the primal-dual algorithm for the AdWords problem [42].

3.4.2 Generalized sequential algorithm

Consider the generalized sequential algorithm in Algorithm 1 which outputs $\tilde{x}_t$ at each online step $t \in [m]$. For all $i \in [n]$, $t \in [m]$ and $k \in \{0, \ldots, K\}$, define:

$$\omega_{it}(k) := \big[[\tilde{x}_1(K)]_i, \ldots, [\tilde{x}_{t-1}(K)]_i, [\tilde{x}_t(k)]_i, \underbrace{0, \ldots, 0}_{m-t \text{ times}} \big]^T,$$

$$[d_t(k)]_i := \nabla_t H_i\big(\omega_{it}(k)\big) + c_{it} G_i'\big(\hat{c}_i^T \omega_{it}(k)\big). \tag{3.4}$$

In the above definitions, we have used the notation $[u]_i$ to denote the i-th entry of the vector u, and ∇_t denotes the t-th entry of the gradient vector.

$\tilde{x}_t$ is the convex combination (average) of vectors in the convex set F_t and hence, $\tilde{x}_t \in F_t$ holds. At each online step $t \in [m]$, the algorithm performs a total of K Frank-Wolfe updates in its inner loop where in each of these updates, a linear maximization problem over the set F_t is solved. Note that in our applications, F_t is usually a box constraint or the simplex, and therefore, the corresponding linear maximization problem could be solved efficiently. See [130] for more details about using Frank-Wolfe for non-convex objectives. To guarantee none of the budget constraints are violated, the penalty design is such that all penalty functions are concave and monotone non-increasing and for all $i \in [n]$, $G'_i(1) = -U_i$ $\big(G'_i(1+\epsilon) = -U_i$ in Remark 3.4.2 and Remark 3.4.3$\big)$ holds. Therefore, for all u such that $\hat{c}_i^T u \geq 1$, $\nabla_t H_i(u) + c_{it} G'_i(\hat{c}_i^T u) \leq 0$ holds for all $t \in [m]$, and considering that $0 \in F_t$, the algorithm would not assign more items to the i-th agent.

The generalized sequential algorithm reduces to the sequential algorithm in [79] for $K = 1$ and hence the name. Note that the sequential algorithm in [79] is used for concave problems with the DR property whereas our generalized sequential algorithm could be exploited for non-concave problems as well. Additionally, our algorithm could be interpreted as the online counterpart of the Frank-Wolfe variant proposed in [25] for solving offline constrained continuous DR-submodular maximization problems. At step $t \in [m], k \in [K]$, the update rule for $v_t(k)$ has the same form as the KKT condition for x_t^*. In other words, the generalized sequential algorithm uses $\nabla H_i(\omega_{it}(k-1))$ and $-G'_i(\hat{c}_i^T \omega_{it}(k-1))$ as the current estimate of $\hat{y}_i^*$ and z_i^* respectively and using them, the algorithm obtains $v_t(k)$ to improve the estimate of x_t^*.

We define:

$$\begin{aligned}
\text{ALG} &:= \sum_{i=1}^{n} H_i(\omega_{im}(K)), \\
\text{P}_{\text{gseq}} &:= \sum_{i=1}^{n} \big(H_i(\omega_{im}(K)) + G_i(\hat{c}_i^T \omega_{im}(K))\big).
\end{aligned}$$

ALG and $\mathrm{P}_{\mathrm{gseq}}$ are the objective values of problems (3.1) and (3.2) at the end of the algorithm respectively.

3.4.3 Competitive ratio analysis

First, we remind the reader that $H_i \ \forall i \in [n]$ are DR-submodular and not necessarily concave. On the other hand, $G_i \ \forall i \in [n]$ are the designed concave penalty functions. In order to derive the competitive ratio, we make the following smoothness assumption about the functions:

Assumption 1: For all $i \in [n]$, functions H_i and G_i have an L-Lipschitz gradient, i.e., for all $x \in K$ and $u \in \mathbb{R}^m$ where $u \succeq 0$ or $u \preceq 0$, the following holds:

$$H_i(x+u) - H_i(x) \geq \langle u, \nabla H_i(x) \rangle - \frac{L}{2}\|u\|^2.$$

Also, for all $x \in \mathbb{R}$ and $v \in \mathbb{R}$, we have:

$$G_i(x+v) - G_i(x) \geq v G_i'(x) - \frac{L}{2} v^2.$$

We also define the parameter α as follows:

Definition 3.4.1 *For all $i \in [n]$, α_{H_i} is defined as:*

$$\begin{aligned}
\alpha_{H_i} &:= \sup\{\beta \mid H_i^*\big(\nabla H_i(u)\big) \geq \beta H_i(u) \ \textit{for all } u \ \textit{s.t. } \hat{c}_i^T u \leq 1\} \\
&= \sup\{\beta \mid \langle \nabla H_i(u), u \rangle \geq (1+\beta) H_i(u) \ \textit{for all } u \ \textit{s.t. } \hat{c}_i^T u \leq 1\} \\
&= \inf_{u : \hat{c}_i^T u \leq 1} \frac{\langle \nabla H_i(u), u \rangle}{H_i(u)} - 1.
\end{aligned}$$

Since H_i is monotone non-decreasing, $0 \leq \langle \nabla H_i(u), u \rangle$ holds. Additionally, because H_i satisfies the DR property and $H_i(0) = 0$, we have $\langle \nabla H_i(u), u \rangle \leq H_i(u)$. Thus, $-1 \leq \alpha_{H_i} \leq 0$ always holds.

The definition above is inspired by the definition of α in [79] (see [79] for a geometric intuition for parameter α for functions with $m = 1$). The parameter α characterizes the curvature of the function over the domain of the algorithm (i.e., where the budget constraint is not violated). For linear functions F, $\alpha_F = 0$ and larger $|\alpha_F|$ corresponds to the function F

being more curved/non-linear. In fact, α of the multilinear extension of a submodular set function and the total curvature of the underlying submodular set function are related as follows:

Remark 3.4.1 ***Connection between submodular total curvature and α:***
If we denote the multilinear extension of a submodular set function F by f, we have:

$$\alpha_f \geq -\kappa_F,$$

where κ_F is the total curvature of F. See Appendix A.3 for the proof and an example.

If we denote the optimal values of the original constrained problem (3.1) and its dual problem (3.3) by OPT and D* respectively, OPT ≤ D* holds due to weak duality.
Now, we have all the required definitions and assumptions to obtain the competitive ratio bounds. In what follows, we first provide Lemma 3.4.1 and Lemma 3.4.2 which are later used to obtain the competitive ratio bounds.

Lemma 3.4.1 *The following inequality holds for the output of the generalized sequential algorithm:*

$$\mathrm{ALG} \geq \sum_{t=1}^{m} \sigma_{F_t}(d_t(K)) - \sum_{i=1}^{n} G_i(\hat{c}_i^T \omega_{im}(K)) - \frac{Lm\lambda^2}{K}. \tag{3.5}$$

Proof See Appendix A.4.1 for the proof. ■

Lemma 3.4.2 *For all $i \in [n]$, we have:*

$$L_i \leq \frac{H_i\big(\omega_{im}(K)\big)}{\hat{c}_i^T \omega_{im}(K)}. \tag{3.6}$$

Proof See Appendix A.4.2 for the proof. ■

Using the above lemmas, we now state and prove our main results, i.e., competitive ratio bounds.

Theorem 3.4.1 *For $n > 1$, if Assumption 1 holds and $K \to \infty$, then for the generalized sequential algorithm, we have:*

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{\text{ALG}}{\text{D}^*} \geq \big(- \min_{i \in [n]} \alpha_{H_i} + \frac{e}{e-1} \max_{i \in [n]} \ln(1 + \frac{U_i(e-1)}{L_i})\big)^{-1}.$$

Proof We can use the definition of α to obtain:

$$H_i^*\big(\nabla H_i(\omega_{im}(K))\big) \geq \alpha_{H_i} H_i(\omega_{im}(K)) \; \forall i \in [n]. \tag{3.7}$$

For all $i \in [n]$, using the definition of G_i for the $n > 1$ case and defining $\gamma_i := \ln(1 + \frac{U_i(e-1)}{L_i})$, we have:

$$\begin{aligned} -G_i'(u) + G_i(u) &= -G_i'(u) + \gamma_i G_i(u) - (\gamma_i - 1)G_i(u) \\ &= \frac{L_i \gamma_i}{e-1} u - (\gamma_i - 1) G_i(u). \end{aligned} \tag{3.8}$$

Combining (3.5), (3.6), (3.7) and (3.8) along with $P_{\text{gseq}} \geq -\frac{Lm\lambda^2}{K}$, we obtain:

$$\begin{aligned} \text{D}^* - \max_{i \in [n]} \gamma_i \frac{Lm\lambda^2}{K} &\leq \sum_{t=1}^{m} \sigma_{F_t}\big(d_t(K)\big) - \max_{i \in [n]} \gamma_i \frac{Lm\lambda^2}{K} - \sum_{i=1}^{n} \big(H_i^*(\nabla H_i(\omega_{im}(K))) + G_i'(\hat{c}_i^T \omega_{im}(K))\big) \\ &\leq \text{ALG} - \sum_{i=1}^{n} \alpha_{H_i} H_i(\omega_{im}(K)) - (\max_{i \in [n]} \gamma_i - 1) \frac{Lm\lambda^2}{K} \\ &\quad + \sum_{i=1}^{n} \big(\frac{L_i \gamma_i}{e-1} \hat{c}_i^T \omega_{im}(K) - (\max_{i \in [n]} \gamma_i - 1) G_i(\hat{c}_i^T \omega_{im}(K))\big) \\ &\leq \text{ALG} - \sum_{i=1}^{n} \alpha_{H_i} H_i(\omega_{im}(K)) + \sum_{i=1}^{n} (\frac{\gamma_i}{e-1} + \max_{i \in [n]} \gamma_i - 1) H_i(\omega_{im}(K)) \\ &\leq \big(1 - \min_{i \in [n]} \alpha_{H_i} + \frac{e}{e-1} \max_{i \in [n]} \gamma_i - 1\big) \text{ALG} \\ &= \big(- \min_{i \in [n]} \alpha_{H_i} + \frac{e}{e-1} \max_{i \in [n]} \gamma_i\big) \text{ALG}. \end{aligned}$$

Therefore, if $K \to \infty$, the competitive ratio bound is derived as $\frac{\text{ALG}}{\text{D}^*} \geq \frac{1}{-\min_{i \in [n]} \alpha_{H_i} + \frac{e}{e-1} \max_{i \in [n]} \gamma_i}$.

■

The bound in Theorem 3.4.1 is tight in several known special cases. For the AdWords problem, since $U_i = L_i = 1$ and $\alpha_{H_i} = 0$ for all $i \in [n]$, competitive ratio of $1 - \frac{1}{e}$ is obtained which is optimal [147]. Additionally, for the online linear programming problem, considering that $\alpha_{H_i} = 0\ \forall i \in [n]$, we obtain $\big(\max_{i\in[n]} \ln(1 + \frac{U_i(e-1)}{L_i})\big)^{-1} \times (1 - \frac{1}{e})$ as the competitive ratio bound which is known to be optimal [79, 43].

Remark 3.4.2 *For $n > 1$, if we allow all the linear packing constraints to be violated by at most ϵ, by modifying the penalty function for all $i \in [n]$ to*

$$G_i(u) = \frac{L_i(1+\epsilon)}{(e-1)\ln(1+\frac{U_i(e-1)}{L_i})}\big(1 - (1+\frac{U_i(e-1)}{L_i})^{\frac{u}{1+\epsilon}}\big) + \frac{L_i}{e-1}u,$$

the competitive ratio bound improves to $(1+\epsilon)\times\big(\max_{i\in[n]}\big\{-(1+\epsilon)\alpha_{H_i} + \ln(1+\frac{U_i(e-1)}{L_i})\frac{e}{e-1}\big\}\big)^{-1}$.

Theorem 3.4.2 *For $n = 1$, if Assumption 1 holds and $K \to \infty$, then for the generalized sequential algorithm, we have:*

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{\text{ALG}}{\text{D}^*} \geq \frac{1}{1 - \alpha_{H_1} + \ln(\frac{U_1}{L_1})}.$$

Proof Considering that $G_1'(u) = \ln(\frac{U_1 e}{L_1})G_1(u)\ ; u \geq \frac{1}{\ln\left(\frac{U_1 e}{L_1}\right)}$, combining (3.5) and (3.7) for $n = 1$ along with $P_{\text{gseq}} \geq -\frac{Lm\lambda^2}{K}$, we obtain:

$$\begin{aligned} D^* - \ln(\frac{U_1 e}{L_1})\frac{Lm\lambda^2}{K} &\leq \sum_{t=1}^{m}\sigma_{F_t}\big(d_t(K)\big) - \ln(\frac{U_1 e}{L_1})\frac{Lm\lambda^2}{K} - H_1^*(\nabla H_1(\omega_{1m}(K))) - G_1'(\hat{c}_1^T\omega_{1m}(K)) \\ &\leq \text{ALG} - \alpha_{H_1}H_1(\omega_{1m}(K)) - \ln(\frac{U_1}{L_1})G_1(\hat{c}_1^T\omega_{1m}(K)) - \ln(\frac{U_1}{L_1})\frac{Lm\lambda^2}{K} \\ &\leq \text{ALG} - \alpha_{H_1}\text{ALG} + \ln(\frac{U_1}{L_1})\text{ALG} \\ &= \big(1 - \alpha_{H_1} + \ln(\frac{U_1}{L_1})\big)\text{ALG}. \end{aligned}$$

Therefore, if $K \to \infty$, the competitive ratio bound is derived as $\frac{1}{1-\alpha_{H_1}+\ln(\frac{U_1}{L_1})}$. ■

If we use the result of Theorem 3.4.2 for the online linear knapsack problem (where $\alpha_{H_1} = 0$), we obtain the competitive ratio bound of $\frac{1}{1+\ln(\frac{U_1}{L_1})}$ which is optimal [217] (note that because we allow fractional assignments, unlike [217], we do not need the $c_{1t} \ll 1\ \forall t \in [m]$ assumption to obtain the optimal competitive ratio).

Remark 3.4.3 *For $n = 1$, if we allow the linear packing constraint to be violated by at most ϵ, modifying the penalty function to*

$$G_1(u) = \begin{cases} -L_1 u & \text{if } 0 \leq u < \frac{1+\epsilon}{\ln\left(\frac{U_1 e}{L_1}\right)} \\ -\frac{1+\epsilon}{\ln\left(\frac{U_1 e}{L_1}\right)} \frac{L_1}{e} \left(\frac{U_1 e}{L_1}\right)^{\frac{u}{1+\epsilon}} & \text{if } u \geq \frac{1+\epsilon}{\ln\left(\frac{U_1 e}{L_1}\right)} \end{cases},$$

we obtain the improved competitive ratio bound of $\frac{1+\epsilon}{-(1+\epsilon)\alpha_{H_1} + \ln(\frac{U_1 e}{L_1})}$.

Theorems 3.4.1 and 3.4.2 provide the first competitive ratio bounds that generalize the results of [11] and [79] for the concave case to general DR-submodular objective functions that are not necessarily concave.

3.5 Experiments

We defined $F_t = \{x \in \mathbb{R}^n : 0 \preceq x \preceq \mathbf{1}\}$ for all $t \in [m]$ and we randomly generated monotone non-convex/non-concave quadratic functions of the form $F(x) = \frac{1}{2}x^T Hx + h^T x$ where $H \in \mathbb{R}^{m \times m}$ is a random matrix with uniformly distributed non-positive entries in $[-100, 0]$ and $h = -H^T \mathbf{1}$ to make the gradient non-negative. Therefore, the utility functions are of the form $F(x) = (\frac{1}{2}x - \mathbf{1})^T Hx$. We set the linear packing constraints to be of the form $Cx \preceq \mathbf{1}$ where $C \in \mathbb{R}^{n \times m}$ has uniformly distributed entries in $[0, 1]$. We set $m = 100$ and $K = 50$. For all $i \in [n]$, the lower and upper bounds L_i and U_i were obtained by the input data. We ran the generalized sequential algorithm for both cases of $n = 1$ and $n > 1$ (note that the penalty function designed in these two cases was different) and to compute the competitive ratio, we divided the output of the algorithm by the offline optimal solution computed by the Frank-Wolfe variant algorithm of [25] with $K = 50$.

Note that the motivating applications mentioned in Appendix A.1 are the continuous counterparts of submodular *generalizations* of some online discrete budgeted problems and in all cases but one, even the online discrete budgeted submodular problem has not been studied before. Therefore, there is no baseline algorithm available to be used for comparison. The only exception is the online discrete submodular knapsack problem studied by [140], however, their algorithm makes integral allocations at each step, and comparing their algorithm with ours which also allows fractional assignments may not be fair.

The average performance of the generalized sequential algorithm over 10 repeated experiments is summarized in Table 3.1.

Quantity	Value (%)
Competitive Ratio	64.33
Budget Usage	74.95

(a) $n = 1, m = 100, K = 50$

Quantity	Value (%)
Competitive Ratio	58.27
Budget 1 Usage	65.68
Budget 2 Usage	58.06
Budget 3 Usage	66.83
Budget 4 Usage	65.11
Budget 5 Usage	74.75

(b) $n = 5, m = 100, K = 50$

Table 3.1: Performance of the generalized sequential algorithm

Table 3.1 shows that the output of the generalized sequential algorithm is not using all of the available budget which is natural in the adversarial input model. In other words, considering that no information about the online input is available, in order to attain a guaranteed competitive ratio, the algorithm needs to be overly cautious so that it does not miss valuable items that are arriving in the later steps due to exhausting all of the budget in the earlier stages.

3.6 Related work

Offline submodular maximization. Consider the problem $\max_{x \in \mathcal{P}} f(x)$ where $f : \mathbb{R}^m \to \mathbb{R}$ is a non-negative monotone DR-submodular function and $\mathcal{P}$ is a down-closed convex set in the positive orthant. In [46], authors considered the special case of maximizing the continuous relaxation of a discrete submodular function subject to a matroid constraint and using a variant of the Frank-Wolfe algorithm, called the continuous greedy process[1], they obtained a $(1-\frac{1}{e})$ approximation ratio. The multilinear extension of a submodular function satisfies the DR property. Our generalized sequential algorithm is the online counterpart of the discretized version of the continuous greedy process where K is the number of discrete steps (i.e., $\delta = \frac{1}{K}$ is the discretization size). [25] obtained a similar approximation ratio for general continuous DR-submodular functions. [128] also exploited the same algorithm to obtain a $(1-\frac{1}{e})$ approximation ratio for submodular maximization subject to multiple linear constraints. Later on, the continuous greedy process has been generalized to obtain approximation ratios for both monotone and non-monotone continuous submodular functions [25, 59, 57, 84, 81, 39]. See [126, 38] for thorough overviews of offline submodular maximization problems and algorithms.

Online submodular maximization. Consider the following problem: At step $t \in \{1, \ldots, T\}$, the online algorithm chooses a feasible point $x_t \in \mathcal{P}$ where $\mathcal{P}$ is a down-closed convex set. Once the algorithm commits to this choice, a monotone continuous DR-submodular function f_t is revealed and the reward $f_t(x_t)$ is received. The goal is to minimize the regret defined as the difference between the total reward obtained by the algorithm and that of the $(1-\frac{1}{e})$ approximation to the best fixed decision in hindsight with $(1-\frac{1}{e})$ being the optimal approximation ratio for an offline monotone continuous DR-submodular maximization problem. Note that although similar to our framework (the objective functions are assumed to be continuous DR-submodular in this setting), there are no packing constraints available and additionally, additive regret bounds are derived in this setting whereas we obtain competitive ratio bounds for our framework which requires different proof techniques. [95] considered the case that the continuous DR-submodular function f_t is the multilinear extension of

---

[1] The continuous greedy process is as follows:

$$\begin{aligned} dy/dt &= v_{\max}(y) \\ v_{\max}(y) &= \arg\max_{v \in \mathcal{P}} \langle v, \nabla f(y) \rangle, \end{aligned}$$

where $y(1) = \int_0^1 v_{\max}(y(\tau))d\tau$ is the output.

a discrete submodular function and $\mathcal{P}$ is the matroid polytope, and achieved an $O(\sqrt{T})$ $(1-\frac{1}{e})$-regret bound. Later on, [61] achieved an $O(\sqrt{T})$ $(1-\frac{1}{e})$-regret bound for general continuous DR-submodular functions. In [60], they further generalized their result and developed a projection-free algorithm that only requires stochastic gradient estimates and achieves a similar regret bound. See [126] for a detailed overview of the online maximization of submodular functions.

Online submodular knapsack problem. Consider the integer problem $\max_{x:c^T x\leq b} F(x)$ where $x=[x_1,\ldots,x_m]^T \in \{0,1\}^m$. In the online setting, at step $t \in [m]$, t-th item arrives, and c_t along with the value of the function F over subsets of $\{1,\ldots,t\}$ is revealed. The algorithm should decide whether to choose this item. [144] showed that in the adversarial setting, there exists no online algorithm achieving any non-trivial competitive ratio for this problem. [217] considered the case where $F(x)=d^T x;\ d \in \mathbb{R}_+^m$ and proved that under the additional assumptions that for all $t \in [m]$, $c_t \ll b$ and $L \leq \frac{d_t}{c_t} \leq U$, there exists an algorithm that achieves the competitive ratio of $\frac{1}{1+ln(\frac{U}{L})}$ and is provably optimal. [140] generalized this algorithm for the case that the function F is submodular and obtained a $\frac{1}{(1+\kappa_F+O(\epsilon))(1+ln(\frac{U}{L}))}$ competitive ratio where $L \leq \frac{f(t|S)}{c_t} \leq U\ \forall t \in [m], S \subset \{1,\ldots,m\}\setminus\{t\}$ and κ_F is the total curvature of F [68]. Note that if we apply our generalized sequential algorithm for $n=1$ to the multilinear extension of the function F (which we denote by f) and allow fractional assignments of items, we obtain the competitive ratio $\frac{1}{1-\alpha_f+\ln(\frac{U}{L})}$ and because $\alpha_f \geq -\kappa_F$ holds by Remark 3.4.1, our bound improves upon the result of [140].

Submodular secretary problems. In this class of problems introduced by [19] and [98], m items are presented to the algorithm in random order. Upon arrival of an item, the algorithm should irrevocably decide whether to accept the current item. The goal is to maximize a monotone submodular function $F:\{0,1\}^m \to \mathbb{R}$ subject to cardinality, matching or linear packing constraints. See [119, 126] for a comprehensive overview of submodular secretary problems. Note that in the submodular secretary problem, the input is assumed to be stochastic while in our framework, the adversarial input model has been considered.

3.7 Conclusion and future work

In this chapter, we considered a class of online optimization problems, where the objective function is monotone DR-submodular under linear packing constraints. We specified various online discrete submodular problems whose continuous generalization could be cast in our framework (see Appendix A.1). We proposed the generalized sequential algorithm for solving such problems and we obtained competitive ratio bounds for this algorithm. Finally, we demonstrated the effectiveness of our algorithm through numerical experiments on a certain class of continuous DR-submodular functions.

This work could be extended in several directions. First, we assumed that the algorithm has access to exact gradient values. It is interesting to extend our results to the setting where only stochastic gradient estimates are available and see how it affects the competitive ratio bounds. Moreover, the generalized sequential algorithm performs K linear maximization steps over the constraint set at every round which could be computationally expensive for more complicated constraint sets. In such cases, designing computationally efficient algorithms is yet to be done.

Chapter 4

ONLINE DR-SUBMODULAR MAXIMIZATION WITH LONG-TERM ADVERSARIAL CONSTRAINTS

4.1 Chapter overview

This chapter is based on a joint work with Maryam Fazel that was published in the International Conference on Artificial Intelligence and Statistics (AISTATS) 2020. In this chapter, we study a class of online optimization problems with long-term budget constraints where the objective functions are not necessarily concave (nor convex), but they instead satisfy the Diminishing Returns (DR) property. In this online setting, a sequence of monotone DR-submodular objective functions and linear budget functions arrive over time, and assuming a limited total budget, the goal is to take actions at each time, before observing the utility and budget function arriving at that round, to achieve sub-linear regret bound while the total budget violation is sub-linear as well. Prior work has shown that achieving sub-linear regret and total budget violation simultaneously is impossible if the utility and budget functions are chosen adversarially. Therefore, we modify the notion of regret by comparing the agent against the best-fixed decision in hindsight which satisfies the budget constraint proportionally over any window of length W. We propose the Online Saddle Point Hybrid Gradient (OSPHG) algorithm to solve this class of online problems. For $W = T$, we recover the aforementioned impossibility result. However, if W is sub-linear in T, we show that it is possible to obtain sub-linear bounds for both the regret and the total budget violation.

4.2 Introduction

4.2.1 Motivating application: Online ad placement

Consider the following online ad placement problem: At round $t \in [T]$, an advertiser should choose an investment vector $x_t \in R^n_+$ over n different websites where i-th entry of x_t denotes the amount that the advertiser is willing to pay per each click on the ad on the i-th website (i.e., cost per click). In other words, each website has different tiers of ads, and choosing x_t corresponds to ordering a certain type of ad. The aggregate cost of investment would be determined when the number of clicks the ad receives is revealed. In other words, the cost of such an investment would be $\langle p_t, x_t \rangle$ where the i-th entry of the vector p_t is the number of clicks the ad on the i-th website receives. Note that the vector p_t is not known ahead of time and could be adversarial. For instance, competing advertisers may click on the ads to deplete their rival's budget. The advertiser needs to balance her total investment against an allotted long-term budget (daily, monthly, etc.), i.e., $\sum_{t=1}^{T} \langle p_t, x_t \rangle \leq B_T$ where B_T is the total targeted budget. At round $t \in [T]$, the advertiser's utility function $f_t(x_t)$ is a monotone DR-submodular function with respect to the vector of investments and this function quantifies the overall amount of impressions of the ads. DR-submodularity of the utility function characterizes the diminishing returns property of the impressions. In other words, making an ad more visible will attract proportionally fewer extra viewers because each website shares a portion of its visitors with other websites. [135] considered the online portfolio management problem (with the online ad placement problem as their running example) and to model diminishing returns, they assumed that the utility functions are separable and concave. Note that concavity is equivalent to DR-submodularity for separable functions. However, for non-separable utility functions, DR-submodularity is the property that captures the diminishing returns of the objective functions (rather than the concavity property) and therefore, in this work, we focus on DR-submodular utility functions that are not necessarily concave. Our results resolve the open problem posed by [135] in the footnote of the third page of their paper. See Appendix B.2 for more applications.

In this chapter, we propose an algorithm for this class of online non-convex problems such that the algorithm has no regret, i.e., a sub-linear regret bound with respect to the horizon T, while the total budget violation is sub-linear as well.

4.2.2 Related work

Online convex optimization with constraints. Consider an online problem where at step $t \in [T]$, the player chooses $x_t \in \mathcal{X}$. Then, cost function $f_t : \mathcal{X} \to \mathbb{R}$ and constraint function $g_t : \mathcal{X} \to \mathbb{R}$ are revealed and the player incurs a loss of $f_t(x_t)$ and her budget is impacted by the amount $g_t(x_t)$. $\mathcal{X}$ is assumed to be convex and compact and the functions f_t, g_t are convex for all $t \in [T]$. The overall goal is to design an algorithm whose output is asymptotically feasible, i.e., the constraint residual $\sum_{t=1}^{T} g_t(x_t)$ is sub-linear, and has a sub-linear regret. [141] considered the case where all constraint functions are equal and are given offline, i.e., $g_t(x) = g(x)\ \forall t \in [T], x \in \mathcal{X}$. For this setting, they achieved $\mathcal{O}(\sqrt{T})$ regret and $\mathcal{O}(T^{\frac{2}{3}})$ constraint residual (i.e., $\sum_{t=1}^{T} g(x_t)$) bounds. [113] studied the exact same framework as [141] and generalized their result by obtaining $\mathcal{O}(T^{\max\{\beta, 1-\beta\}})$ regret and $\mathcal{O}(T^{1-\frac{\beta}{2}})$ constraint residual bounds where $\beta \in (0,1)$ is a tunable parameter. More recently, [213] considered an alternative notion of constraint residual defined as the sum of squares of clipped residuals, $\sum_{t=1}^{T} (\max\{g(x_t), 0\})^2$, and achieved $\mathcal{O}(T^{\max\{\beta, 1-\beta\}})$ regret and $\mathcal{O}(T^{1-\beta})$ constraint residual bounds for this setting. Also, they obtained logarithmic regret bounds for the case that cost functions are strongly convex. The new constraint residual form considered in [213] heavily penalizes large constraint violations and strictly feasible solutions of some rounds cannot cancel out the effect of violated constraints at other rounds. For the setting with constraints arriving online[1], [143] considered the notion of regret with window length $W = T$ and provided a simple counterexample showing that the regret of any algorithm with sub-linear constraint residual is lower bounded by $\Omega(T)$. [159] assumed that there exists an action $x^* \in \mathcal{X}$ such that $g_t(x^*) < 0\ \forall t \in [T]$ (Slater condition) and under

---

[1] In this setting, the regret metric with window length W is defined as $R_T = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x_W^*)$ where $x_W^* = \arg\min_{x \in \mathcal{X}_W} \sum_{t=1}^{T} f_t(x)$ and $\mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ 1 \leq t \leq T - W + 1\}$.

this assumption, they obtained $\mathcal{O}(\sqrt{T})$ bounds for both regret with window size $W = 1$ and constraint residual. However, the fixed decision benchmark action considered in this paper is restricted to be feasible for all constraint functions $g_t \; \forall t \in [T]$ which heavily restricts the performance of the benchmark action and thus, the obtained regret guarantees could be loose. [196] considered the same notion of regret as [159] and using online mirror descent as a subroutine (and without assuming the Slater condition), they obtained a similar $\mathcal{O}(\sqrt{T})$ regret bound and a looser $\mathcal{O}(T^{\frac{3}{4}})$ constraint residual bound. [135] considered the same framework and algorithm as [159], however, they constrained the benchmark action to be feasible in all windows of size $W = T^{\beta}$ where $\beta \in [0, 1)$ (as opposed to [159] where $W = 1$). They obtained $\mathcal{O}(\frac{WT}{V} + \sqrt{T})$ regret bound and $\mathcal{O}(\sqrt{VT})$ residual bound where $V \in [W, T)$ is a tunable parameter which captures the trade-off between the regret and constraint residual bounds. Note that for $W = 1$, since the Slater condition is not assumed by [135], their bound does not achieve the $\mathcal{O}(\sqrt{T})$ regret and constraint residual bound of [159]. More recently, this framework has been extended to distributed settings and bandit feedback as well [210, 211].

Note that in all these works, the objective functions are assumed to be convex. In contrast, we consider a more general class of non-convex/non-concave DR-submodular objective functions to which the aforementioned results are not applicable.

Online submodular maximization. An orthogonal research direction considers the following problem: At step $t \in [T]$, the online algorithm chooses a feasible point $x_t \in \mathcal{P}$. Once the algorithm commits to this choice, a monotone continuous DR-submodular function f_t is revealed and the reward $f_t(x_t)$ is received. The goal is to minimize the regret defined as the difference between the total reward obtained by the algorithm and that of the $(1 - \frac{1}{e})$-approximation to the best fixed decision in hindsight with $(1 - \frac{1}{e})$ being the optimal polynomial time approximation ratio for an *offline* monotone continuous DR-submodular maximization problem [25]. Note that although similar to our framework (the objective functions are assumed to be continuous DR-submodular in this setting), no time-varying constraints are arriving online and therefore, they do not deal with the considerable compli-

Algorithm 1 Online submodular maximization meta-algorithm [61]

```
Input: $\mathcal{P}$ is a convex set and $T$ is the horizon.
Output: $\{x_t : 1 \leq t \leq T\}$.
Choose an off-the-shelf online linear maximization algorithm and initialize $K$ instances $\mathcal{E}_k;\ k \in [K]$ of it for online maximization of linear utility functions over $\mathcal{P}$.
for $t = 1$ to $T$ do
    Set $x_t^{(1)} = 0$.
    for $k = 1$ to $K$ do
        Let $v_t^{(k)}$ be the vector selected by $\mathcal{E}_k$.
        $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
    end for
    Play $x_t = x_t^{(K+1)}$, observe the function $f_t$ and the reward $f_t(x_t)$.
    For all $k \in [K]$, feedback $\langle v_t^{(k)}, \nabla f_t(x_t^{(k)}) \rangle$ as the payoff to be received by $\mathcal{E}_k$.
end for
```

cation of bounding the constraint residual.

The meta-algorithm for the online submodular maximization problem is presented in Algorithm 1. The intuition for using K online maximization subroutines to obtain $x_t;\ t \in [T]$ is the Frank-Wolfe variant proposed by [25] to obtain the optimal approximation guarantee of $(1 - \frac{1}{e})$ for solving the offline DR-submodular maximization problem. To be more precise, consider the first iteration $t = 1$ of the online setting and the corresponding DR-submodular utility function $f_1(\cdot)$ arriving at this step. Note that f_1 is not revealed until the algorithm commits to an action $x_1 \in \mathcal{P}$. If f_1 were available offline, the mentioned Frank-Wolfe variant of [25] could have been used for K iterations to maximize f_1 over $\mathcal{P}$. Starting from $x_1^{(1)} = 0$, for all $k \in [K]$, a vector $v_1^{(k)}$ would have been found that maximizes $\langle x, \nabla f_1(x_1^{(k)}) \rangle$ over $x \in \mathcal{P}$ and using the update $x_1^{(k+1)} = x_1^{(k)} + \frac{1}{K} v_1^{(k)}$, $x_1 = x_1^{(K+1)}$ would have been derived as the output. However, in the online setting, the utility function f_1 is not available before committing to the action x_1. Therefore, for each $k \in [K]$, a separate instance of a no-regret online linear maximization algorithm is used instead to obtain $v_1^{(k)}$. The same process is repeated for the subsequent utility functions $f_t;\ t > 1$ as well.

[95] considered the case that the continuous DR-submodular function f_t is the multilinear extension of a discrete submodular function and $\mathcal{P}$ is the matroid polytope. Using the Perturbed Follow the Leader (PFTL) as the online algorithm, they achieved an $\mathcal{O}(\sqrt{T})$ bound for the $(1 - \frac{1}{e})$-regret. [61] used the Follow The Regularized Leader (FTRL) online algorithm and achieved a similar regret bound for general continuous DR-submodular functions. In [60], they further generalized their result and developed a projection-free algorithm that only requires stochastic gradient estimates of $f_t\ \forall t \in [T]$ and achieves the same regret guarantees. See [126] for a detailed overview of the online maximization of submodular set functions.

4.2.3 Contributions

In this chapter, we design an algorithm for online continuous DR-submodular maximization problems with long-term budget constraints to achieve sub-linear regret and budget violation bounds simultaneously. Specifically, we make the following contributions:

- We introduce the online continuous DR-submodular maximization problem with long-term budget constraints. The online ad placement example mentioned in section 4.2.1 is an application of this framework. We also provide several other motivating applications for this framework in Appendix B.2.
- We propose the Online Saddle Point Hybrid Gradient (OSPHG) algorithm to solve this class of online problems. Our algorithm is inspired by that of [196] and [61]. We consider a refined notion of static regret where the agent's utility is compared against a $(1 - \frac{1}{e})$-approximation to the best fixed decision in hindsight which satisfies the budget constraint proportionally over any window of length W. For $W = T$, we recover the known impossibility result obtained by [143]. However, for $W = o(T)$, we obtain sub-linear bounds for both the $(1 - \frac{1}{e})$-regret and the total budget violation. In particular, if $W = T^{1-\epsilon}$ for $0 < \epsilon \leq 1$, we obtain a $(1 - \frac{1}{e})$-regret bound of $\mathcal{O}(T^{1-\frac{\epsilon}{2}})$ while the total budget violation is $\mathcal{O}(T^{1-\frac{\epsilon}{4}})$.

Finally, we illustrate the performance of our algorithm through numerical examples for a class of non-convex/non-concave continuous DR-submodular objective functions.

4.3 Problem statement

The overall offline optimization problem is the following:

$$\begin{array}{ll} \max_{x_t \in \mathcal{X}\ \forall t \in [T]} & \sum_{t=1}^{T} f_t(x_t) \\ \text{subject to} & \sum_{t=1}^{T} \langle p_t, x_t \rangle \leq B_T \end{array}. \tag{4.1}$$

The online framework is as follows: At step $t \in [T]$, the player chooses $x_t \in \mathcal{X}$. Then, utility function $f_t : \mathcal{X} \to \mathbb{R}$ and $p_t \in \mathbb{R}_+^n$ are revealed, the player obtains the reward $f_t(x_t)$ and her budget is impacted by the amount $\langle p_t, x_t \rangle$. It is assumed that $\mathcal{X} \subset \mathbb{R}_+^n$ is convex and compact. For all $t \in [T]$, $f_t : \mathcal{X} \to \mathbb{R}$ is a differentiable normalized monotone continuous DR-submodular function, and the constraint function $g_t : \mathcal{X} \to \mathbb{R}$, where $g_t(x) = \langle p_t, x \rangle - \frac{B_T}{T}$, is linear and monotone, i.e., $p_t \succeq 0$.

As mentioned before, there are existing works on online continuous submodular maximization without online constraints (e.g., [61]) and online convex optimization with constraints (e.g., [196]). However, our results are *not* a straightforward combination of the mentioned works. First of all, since $p_t \succeq 0\ \forall t \in [T]$, the Lagrangian $f_t(x) - \lambda_t g_t(x)$ is not monotone non-decreasing and therefore, we cannot simply apply the algorithm of [61] to the Lagrangian to obtain the desired results. In this work, we exploit other properties such as the linearity and non-negativity of the constraint functions to obtain the results. Secondly, the techniques and algorithms for online convex optimization are quite different from online continuous DR-submodular maximization and thus, the analysis in [196] could not be simply adapted to our framework.

4.3.1 Performance metric

In order to quantify the performance of our proposed algorithm, we first define our notion of regret and total budget violation below:

Definition 4.3.1 (Regret Metric) *The* $(1-\frac{1}{e})$*-regret is defined as:*

$$R_T = (1-\frac{1}{e})\sum_{t=1}^{T} f_t(x_W^*) - \sum_{t=1}^{T} f_t(x_t),$$

where:

$$x_W^* = \arg\max_{x\in\mathcal{X}_W} \sum_{t=1}^{T} f_t(x),$$

$$\mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ 1 \leq t \leq T-W+1\}.$$

Note that [135] first introduced the notion of a "K-benchmark", i.e., a comparator that meets the problem's allotted budget over any window of length K, and used this notion for online convex problems with time-varying constraints.

Definition 4.3.2 (Total Budget Violation) *The total budget violation is defined as:*

$$C_T = \sum_{t=1}^{T} g_t(x_t) = \sum_{t=1}^{T} \langle p_t, x_t\rangle - B_T.$$

The regret metric R_T measures the difference between the output of the algorithm and the $(1-\frac{1}{e})$-approximation to the best fixed decision in hindsight which satisfies the budget constraint proportionally over any window of length W. The $(1-\frac{1}{e})$ approximation ratio in the definition is the optimal polynomial time approximation ratio for an offline monotone continuous DR-submodular maximization problem [25] and it is commonly used in online submodular maximization literature (e.g., [61]). If we choose $W = T$ as the window size, we obtain the usual notion of regret where the benchmark action is only required to satisfy the long-term budget constraint and thus, the benchmark action is more aggressive. However, [143] provided a simple counterexample showing that the regret of *any algorithm* with window length $W = T$ which has a sub-linear total budget violation is lower bounded by $\Omega(T)$. Hence, inspired by [135], we restricted the benchmark action to satisfy the budget constraint proportionally over any window of size W.

We design online algorithms that achieve sub-linear bounds for both the $(1-\frac{1}{e})$-regret R_T

and the total budget violation C_T. We obtain results showing that although it is impossible to obtain sub-linear regret and total budget violation bounds simultaneously for $W = T$, such sub-linear bounds could be guaranteed against a weaker benchmark with window length $W = o(T)$.

4.3.2 Assumptions

We make the following assumptions:

- $\mathcal{X} \subset \mathbb{R}^n_+$ is a compact and convex set and it contains the origin, i.e., $0 \in \mathcal{X}$.
- The bounded diameter of the compact set $\mathcal{X}$ is R, i.e., we have:

$$R := \max_{x,y \in \mathcal{X}} \|y - x\|.$$

- Both the utility functions f_t, $\forall t \in [T]$ and constraint functions g_t, $\forall t \in [T]$ are Lipschitz continuous with parameters β_f and β_g respectively and $\beta = \max\{\beta_f, \beta_g\}$. In other words, for all $x, y \in \mathcal{X}$ and $t \in [T]$, we have:

$$\begin{aligned} |f_t(y) - f_t(x)| &\leq \beta_f \|y - x\|, \\ |g_t(y) - g_t(x)| &\leq \beta_g \|y - x\|. \end{aligned}$$

Note that since g_t is linear for all $t \in [T]$, $\beta_g = \max_{t \in [T]} \|p_t\|$ holds.

- For all $t \in [T]$, the utility functions f_t are L-smooth, i.e., for all $x \in \mathcal{X}$ and $u \in \mathbb{R}^n$ where $u \succeq 0$ or $u \preceq 0$, the following holds:

$$f_t(x + u) - f_t(x) \geq \langle u, \nabla f_t(x) \rangle - \frac{L}{2}\|u\|^2.$$

Using the above assumptions, we have:

$$\begin{aligned} F &:= \max_{t \in [T]} \max_{x,y \in \mathcal{X}} |f_t(x) - f_t(y)| \leq \beta_f R, \\ G &:= \max_{t \in [T]} \max_{x \in \mathcal{X}} |g_t(x)| \leq \beta_g R - \frac{B_T}{T}. \end{aligned}$$

Algorithm 2 Online Saddle Point Hybrid Gradient (OSPHG) algorithm

```
Input: $\mathcal{X}$ is the constraint set, $T$ is the horizon, $\mu > 0$, $\delta > 0$ and $K$.
Output: $\{x_t : 1 \leq t \leq T\}$.
Initialize $K$ instances $\mathcal{E}_k;\ k \in [K]$ of Online Gradient Ascent with step size $\mu$ for online maximization of linear functions over $\mathcal{X}$.
$\lambda_1 = 0$.
for $t = 1$ to $T$ do
  $x_t^{(1)} = 0$.
  for $k = 1$ to $K$ do
    Let $v_t^{(k)}$ be the output of oracle $\mathcal{E}_k$.
    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  end for
  Play $x_t = x_t^{(K+1)}$ and observe the Lagrangian function $\mathcal{L}_t(x_t, \lambda_t) = f_t(x_t) - \lambda_t g_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$.
  for $k = 1$ to $K$ do
    Feedback $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\rangle$ as the payoff to be received by $\mathcal{E}_k$.
  end for
  $\lambda_{t+1} = [\lambda_t - \mu \nabla_\lambda \mathcal{L}_t(x_t, \lambda_t)]_+$.
end for
```

4.4 Online Saddle Point Hybrid Gradient (OSPHG): Algorithm and analysis

We first introduce our proposed algorithm, the Online Saddle Point Hybrid Gradient (OSPHG) algorithm, in Section 4.4.1 and then, the analysis for obtaining the regret and total budget violation bounds is provided in Section 4.4.2.

4.4.1 Algorithm

Consider the Online Saddle Point Hybrid Gradient (OSPHG) algorithm presented in Algorithm 2. First, note that x_t is the convex combination (average) of vectors in the convex set $\mathcal{X}$ and hence, $x_t \in \mathcal{X}$ as well.

The OSPHG algorithm could be interpreted as running two no-regret procedures:

1. K instances $\mathcal{E}_k$ of Online Gradient Ascent where for each $k \in [K]$, at online step $t \in [T]$, the algorithm chooses the point $v_t^{(k)}$ and after committing to this choice, it receives a reward of $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\rangle$. Note that each instance $\mathcal{E}_k$ $\forall k \in [K]$ corresponds to an online linear maximization problem. The update for $v_{t+1}^{(k)}$ is as follows:

   $$v_{t+1}^{(k)} = \mathcal{P}_{\mathcal{X}}\big(v_t^{(k)} + \mu \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\big),$$

   where $\mathcal{P}_{\mathcal{X}}$ is the projection onto set $\mathcal{X}$. Note that in our applications, the domain set $\mathcal{X}$ is usually a box constraint or the simplex, and therefore, projection on $\mathcal{X}$ can be efficiently computed.

2. Online Gradient Descent for the sequence of losses $\{\mathcal{L}_t(x_t, \lambda)\}_{t=1}^T$ where at each online step $t \in [T]$, the algorithm chooses $\lambda_t \geq 0$ and then, observes the loss $-\lambda_t g_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$. Note that this is an online quadratic minimization problem.

Therefore, the OSPHG algorithm is solving an online saddle point problem at each step hence the name. It is noteworthy that although we used Online Gradient Descent/Ascent as subroutines in the OSPHG algorithm, *any* other off-the-shelf no-regret online optimization algorithms (such as Online Mirror Descent, Follow The Regularized Leader, etc.) could have been used instead and similar bounds would have been derived. Potential advantages of any such no-regret algorithm over the other could indeed be an interesting research direction.

Our choice of Lagrangian function is inspired by the quadratic penalty method in constrained optimization [165]. The penalized formulation of the overall optimization problem (4.1) with quadratic penalty function could be written as follows:

$$\begin{aligned} \max \quad & \sum_{t=1}^{T} f_t(x_t) - \frac{1}{2\delta\mu}\big(\sum_{t=1}^{T}\langle p_t, x_t\rangle - B_T\big)^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Considering that the Fenchel conjugate of the function $h(\cdot) = \frac{1}{2\delta\mu}(\cdot)^2$ is $h^*(\cdot) = \frac{\delta\mu}{2}(\cdot)^2$, we can write the above problem in the following equivalent form:

$$\begin{aligned} \max_{x_t} \min_{\lambda} \quad & \sum_{t=1}^{T} f_t(x_t) - \lambda\big(\sum_{t=1}^{T}\langle p_t, x_t\rangle - B_T\big) + \frac{\delta\mu}{2}\lambda^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Therefore, the corresponding Lagrangian function at round $t \in [T]$ is $\mathcal{L}_t(x, \lambda) = f_t(x) - \lambda(\langle p_t, x\rangle - \frac{B_T}{T}) + \frac{\delta\mu}{2}\lambda^2$.

4.4.2 Analysis

In order to prove the regret and budget violation bounds, we first provide Lemmas 4.4.1, 4.4.2, and 4.4.3.

Lemma 4.4.1 *For all $t \in [T]$, the following holds:*

$$\mu \sum_{s=1}^{t} \gamma^{t-s} g_s(x_s) \le \lambda_{t+1} \le \mu \sum_{s=1}^{t} \gamma^{t-s} |g_s(x_s)|$$

where $\gamma = 1 - \delta\mu^2$.

Proof See Appendix B.3.1 for the proof. ■

Using Lemma 4.4.1 and the inequality $1 - \delta\mu^2 \le 1$, we can conclude that for all $t \in [T]$, $\lambda_{t+1} \le \mu t G$ holds. We will use this fact multiple times in the proofs.

Lemma 4.4.2 *For a fixed $t \in \{1, \ldots, T - W + 1\}$, if δ and μ are chosen such that $\delta\mu^2 \le \frac{1}{2}$, the following holds:*

$$\sum_{\tau=0}^{W-1} \lambda_{t+\tau} g_{t+\tau}(x_W^*) \le \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + G^2\mu W(W-1).$$

Proof See Appendix B.3.2 for the proof. ■

Lemma 4.4.3 *For $\mu = \frac{R}{\beta\sqrt{WT}}$, $\delta = 4\beta^2$ and any $\lambda \geq 0$, if T is large enough such that $T \geq \frac{16R^2}{W}$ holds, we have:*

$$R_T + C_T\lambda - \frac{\delta\mu}{2}T\lambda^2 - \frac{\lambda^2}{\mu} \leq (F + \beta R)(W-1) + \frac{R^2}{\mu} + (G^2 + \beta^2)\mu T$$
$$+ \frac{G}{2}(G + \beta R)\mu(W-1)(T-1) + G^2\mu(W-1)(T-W+1) + \frac{LR^2}{2K}(T-W+1). \quad (4.2)$$

Proof See Appendix B.3.3 for the proof. ■

Lemmas 4.4.1, 4.4.2 and 4.4.3 are all new results. Note that although [196] use the same Lagrangian function $\mathcal{L}_t$ as our work, they study online *convex* problems (as opposed to our non-convex problem) with window length $W = 1$ and their algorithm and results do not apply to our framework. [135] provide results similar to our lemmas 4.4.1 and 4.4.2 for the convex problem, however, their algorithm and choice of Lagrangian function $\mathcal{L}_t(x, \lambda)$, $\forall t \in [T]$ is different from ours and therefore, their results do not apply to our work.
Now, we have all the required tools to prove the performance bounds of the OSPHG algorithm.

Theorem 4.4.1 (Regret bound) *For $W = o(T)$, if we choose $\mu = \frac{R}{\beta\sqrt{WT}} = \mathcal{O}(\frac{1}{\sqrt{WT}})$, $K = \mathcal{O}(\sqrt{\frac{T}{W}})$ and T is large enough such that $T \geq \frac{16R^2}{W}$ holds, the $(1-\frac{1}{e})$-regret R_T satisfies the following:*

$$R_T \leq \mathcal{O}(\sqrt{WT}).$$

Thus, for $W = T^{1-\epsilon}$, $\forall\epsilon > 0$, the $(1-\frac{1}{e})$-regret of the OSPHG algorithm is $\mathcal{O}(T^{1-\frac{\epsilon}{2}})$ and hence sub-linear.

Proof If we plug in $\lambda = 0$, $\mu = \frac{R}{\beta\sqrt{WT}} = \mathcal{O}(\frac{1}{\sqrt{WT}})$ and $K = \mathcal{O}(\sqrt{\frac{T}{W}})$ in inequality (4.2), the dominating terms on the right-hand side of the inequality are $\frac{G}{2}(G+\beta R)\mu(W-1)(T-1) = \mathcal{O}(\sqrt{WT})$, $\frac{R^2}{\mu} = \mathcal{O}(\sqrt{WT})$, $G^2\mu(W-1)(T-W+1) = \mathcal{O}(\sqrt{WT})$, and $\frac{LR^2}{2K}(T-W+1) = \mathcal{O}(\sqrt{WT})$ and therefore, the result follows. ■

Theorem 4.4.2 (Budget violation bound) *For $W = o(T)$, if we choose $\mu = \frac{R}{\beta\sqrt{WT}} = \mathcal{O}(\frac{1}{\sqrt{WT}})$, $K = \mathcal{O}(\sqrt{\frac{T}{W}})$ and T is large enough such that $T \geq \frac{16R^2}{W}$ holds, C_T is bounded as follows:*

$$C_T \leq \mathcal{O}(W^{\frac{1}{4}}T^{\frac{3}{4}}).$$

Therefore, for $W = T^{1-\epsilon}$ $\forall\epsilon > 0$, the OSPHG algorithm achieves a sub-linear budget violation bound of $\mathcal{O}(T^{1-\frac{\epsilon}{4}})$.

Proof First, we observe that by assumption, $R_T \geq -FT$ holds where F is defined at the end of Section 4.3.2. Assume that $C_T \geq 0$ (otherwise, we are done). Setting $\lambda = \frac{C_T}{\delta\mu T+\frac{2}{\mu}}$ in inequality (4.2), we obtain:

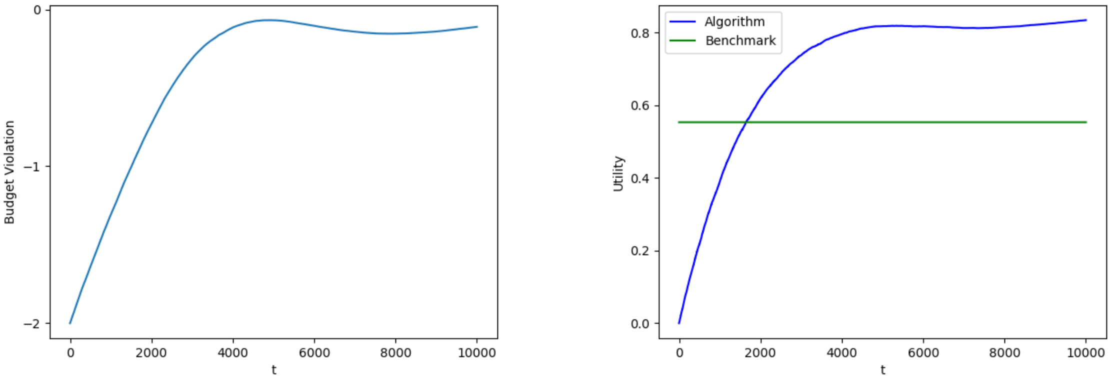
$$\frac{C_T^2}{2\delta\mu T + \frac{4}{\mu}} \leq FT + (F + \beta R)(W-1) + \frac{G}{2}(G+\beta R)\mu(W-1)(T-1) + \frac{R^2}{\mu}$$
$$+ (G^2+\beta^2)\mu T + G^2\mu(W-1)(T-W+1) + \frac{LR^2}{2K}(T-W+1).$$

Plugging in $\mu = \frac{R}{\beta\sqrt{WT}} = \mathcal{O}(\frac{1}{\sqrt{WT}})$ and $K = \mathcal{O}(\sqrt{\frac{T}{W}})$ in the above inequality and multiplying both sides by $2\delta\mu T + \frac{4}{\mu}$, the dominating term on the right-hand side of the inequality is $FT(2\delta\mu T + \frac{4}{\mu}) = \mathcal{O}(W^{\frac{1}{2}}T^{\frac{3}{2}})$. Therefore, $C_T^2 \leq \mathcal{O}(W^{\frac{1}{2}}T^{\frac{3}{2}})$ holds. Taking the square root of both sides, we obtain the desired result. ■

Theorems 4.4.1 and 4.4.2 provide the first sub-linear regret and total budget violation bounds respectively for the online DR-submodular maximization problem with long-term budget constraints.

4.5 Numerical examples

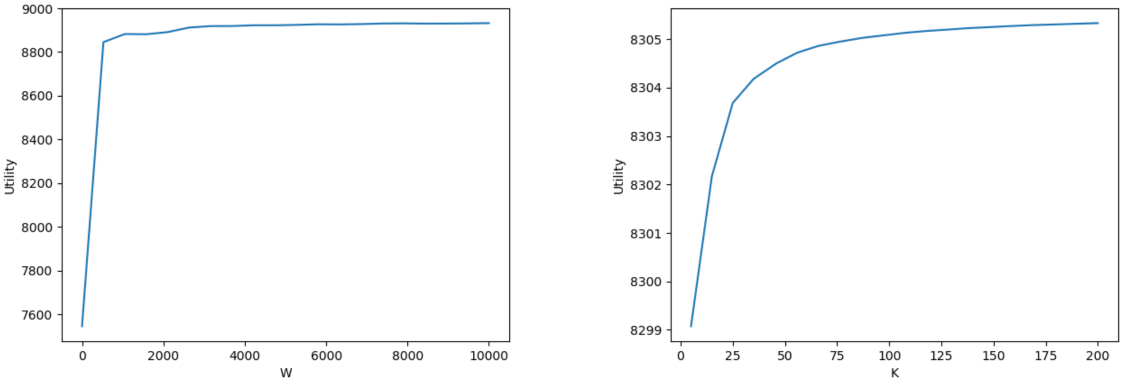
We defined $\mathcal{X} = \{x \in \mathbb{R}^n : 0 \preceq x \preceq \mathbf{1}\}$ and for all $t \in [T]$, we randomly generated monotone non-convex/non-concave quadratic utility functions of the form $f_t(x) = \frac{1}{2}x^T H_t x + h_t^T x$ where $H_t \in \mathbb{R}^{n\times n}$ is a random matrix with uniformly distributed non-positive entries in $[-1, 0]$ and $h_t = -H_t^T\mathbf{1}$ to make the gradient non-negative. Therefore, the utility functions are



(a) (b)

Figure 4.1: Running average of (a) budget violation and (b) regret of the OSPHG algorithm for $W = \sqrt{T}$.

of the form $f_t(x) = (\frac{1}{2}x - \mathbf{1})^T H_t x$. For all $t \in [T]$, we generated random linear budget functions such that p_t has uniformly distributed entries in $[2, 4]$. We set $T = 10000$, $n = 2$, $B_T = 2T$ and $K = 100$. We ran the OSPHG algorithm for $W = \sqrt{T}$. All codes were implemented in Python 3.7. The running average of the budget violation and utility of the OSPHG algorithm is depicted in Figure 4.1(a) and Figure 4.1(b) respectively which verifies the sub-linearity of the total budget violation and regret of our algorithm (note that the average total budget violation is negative and also, the algorithm achieves higher utilities compared to the benchmark). Additionally, we used the Frank-Wolfe variant algorithm of [25] with $K = 100$ (the same value of K used in the OSPHG algorithm) for solving offline constrained DR-submodular optimization problems to obtain the utility of the benchmark for different window lengths. As it can be seen in Figure 4.2(a), choosing larger window sizes leads to higher utility for the corresponding benchmark and hence, tighter regret guarantees are obtained. However, for a large enough W, there is merely a small difference between the obtained benchmark utility versus the case that $W = T$. Overall obtained utility of



(a) (b)

Figure 4.2: Overall utility of (a) benchmark and (b) algorithm for different values of (a) W and (b) K.

the algorithm for different values of K is plotted in Figure 4.2(b) and it could be seen that choosing larger values of K leads to a very small increase in the performance which is not significant.

Note that the algorithm of [61] does not consider online constraints and the algorithms of [196, 135] are designed for online convex problems, and therefore, neither of these algorithms (nor the vanilla online gradient descent) could be simply adapted to our framework to be used as the baseline algorithm for comparison. Our work indeed provides the first algorithm for online DR-submodular maximization with long-term budget constraints.

4.6 Conclusion and future work

In this chapter, we studied a class of online optimization problems with long-term linear budget constraints where the utility functions are monotone continuous DR-submodular. We proposed the Online Saddle Point Hybrid Gradient (OSPHG) algorithm to solve such problems. We considered a refined notion of static regret and proved sub-linear $(1-\frac{1}{e})$-regret and budget violation bounds. Finally, we verified our theoretical findings through a numerical example for a class of continuous DR-submodular functions.

In this chapter, we focused on the setting with adversarial constraints. In the next chapter, we study the same problem framework under the assumption that the long-term linear constraints are stochastic, and we show that in this easier case, it is possible to obtain sublinear regret and constraint violation bounds without the need to refine the regret metric.

Chapter 5

ONLINE DR-SUBMODULAR MAXIMIZATION WITH LONG-TERM STOCHASTIC CONSTRAINTS

5.1 Chapter overview

This chapter presents the joint work with Prasanna Raut and Maryam Fazel that was published at the AAAI 2021 Conference on Artificial Intelligence. In this chapter, we consider online continuous DR-submodular maximization with linear stochastic long-term constraints. Compared to the prior work on online submodular maximization, our setting introduces the extra complication of stochastic linear constraint functions that are i.i.d. generated at each round. In particular, at each time step a DR-submodular utility function and a constraint vector, i.i.d. generated from an unknown distribution, are revealed after committing to an action and we aim to maximize the overall utility while the expected cumulative resource consumption is below a fixed budget. Stochastic long-term constraints arise naturally in applications where there is a limited budget or resource available and resource consumption at each step is governed by stochastically time-varying environments. We propose the Online Lagrangian Frank-Wolfe (OLFW) algorithm to solve this class of online problems. We analyze the performance of the OLFW algorithm and obtain sub-linear regret bounds as well as sub-linear cumulative constraint violation bounds, both in expectation and with high probability.

5.2 Introduction

The Online Convex Optimization (OCO) problem has been extensively studied in the literature [104, 189, 218, 167]. In this problem, a sequence of arbitrary convex cost functions $\{f_t(\cdot)\}_{t=1}^{T}$ are revealed one by one by "nature" and at each round $t \in [T]$, the decision maker chooses an action $x_t \in \mathcal{X}$, where $\mathcal{X}$ is the fixed domain set, before the corresponding function

$f_t(\cdot)$ is revealed. The goal is to minimize the regret defined as [218]

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x).$$

In other words, regret characterizes the difference between the overall cost incurred by the decision maker and that of a fixed benchmark action which has access to all the cost functions $\{f_t\}_{t=1}^{T}$.

In many applications, however, in addition to maximizing the total reward (minimizing the overall cost), there are restrictions on the sequence of decisions made by the learner that need to be satisfied on average [7, 16, 6, 176]. Therefore, it may be beneficial to sacrifice some of the reward to meet other desired goals or restrictions over the time horizon. Such long-term constraints arise naturally in applications with limited budget (resource) availability [18, 21, 87].

As an illustrative example, consider the online ad allocation problem for an advertiser. At each round $t \in [T]$, the advertiser should choose her investment on ads to be placed on n different websites. Beyond the immediate goal of maximizing the overall impressions of the ads, the advertiser needs to balance her total investment against an allotted budget on a daily, monthly, or yearly basis [18]. However, the cost of ad placement in each round depends on the number of clicks the ads receive, so they are not known ahead of time. Therefore, the advertiser needs to strike the right balance between the total reward and the budget used. See Appendix C.1 for several other motivating applications that can be naturally cast in our framework.

In this chapter, we study a new class of online allocation problems with long-term resource constraints where the utility functions are DR-submodular (and not necessarily concave) and the constraint functions are linear with coefficient vectors drawn i.i.d. from some unknown underlying distribution. The problem has been extensively studied in the convex setting [213, 206, 159, 135]; furthermore, [181] considered a similar framework under the assumption that the linear constraint functions are chosen adversarially. However, [181] do not provide any high probability bounds for the regret and constraint violation with random i.i.d. linear constraints, and their expected constraint violation bound is worse than ours as well (see Section 5.3.2 and Appendix C.5 for an overview of related work and comparison of our results with the existing bounds respectively). In this chapter, we provide the *first* sub-linear bounds for the regret and total budget violation that hold in expectation as well as with high probability. Specifically, our contributions are as follows:

- In Section 5.4.1, We propose the Online Lagrangian Frank-Wolfe (OLFW) algorithm for this class of online continuous DR-submodular maximization problems with stochastic cumulative constraints. The OLFW algorithm is inspired by the quadratic penalty method in constrained optimization literature [165] and it generalizes a Frank-Wolfe variant proposed by [61] for solving online continuous DR-submodular maximization problems to take into account the additional stochastically time-varying linear constraints. Note that this extension is not straightforward and the choice of the penalty function and the update rule for the dual variable are crucial in obtaining bounds for the total budget violation as well as the regret (see Section 5.4.1 for more details).
- We analyze the performance of the OLFW algorithm with high probability and in expectation in Section 5.4.2 and Section 5.4.3 respectively and we establish the first sub-linear *expected* and *high probability* bounds on both the *regret* and *total budget violation* of the algorithm.

Finally, in Section 5.5, we demonstrate the effectiveness of our proposed algorithm on simulated and real-world problem instances and compare the performance of the OLFW algorithm with several baseline algorithms.

5.3 Problem statement

Consider the following overall offline optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_{t=1}^{T} f_t(x_t) \\ \text{subject to} & x_t \in \mathcal{X},\ \forall t \in [T] \\ & \sum_{t=1}^{T} \langle p, x_t \rangle \leq B_T. \end{array} \tag{5.1}$$

The online setup is as follows: At each round $t \in [T]$, the algorithm chooses an action $x_t \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}_+^n$ is a fixed and known set. Upon committing to this action, the utility function $f_t : \mathcal{X} \to \mathbb{R}_+$ and a random i.i.d. sample $p_t \sim \mathcal{D}(p, \Sigma)$ are revealed and the algorithm receives a reward of $f_t(x_t)$ while using $\langle p, x_t \rangle$ of its fixed total allotted budget B_T. The overall goal is to maximize the total obtained reward while satisfying the budget constraint asymptotically (i.e., $\sum_{t=1}^{T} \langle p, x_t \rangle - B_T$ being sub-linear in T). [142] considered a similar setup and performance metric for the special case of linear utility functions.

Note that our proposed algorithm can handle multiple linear constraints as well, and similar regret and constraint violation bounds can be derived. However, for ease of notation, we focus on the case with only one linear constraint.

We make the following assumptions about our problem framework:

A1. The domain $\mathcal{X} \subset \mathbb{R}_+^n$ is a closed, bounded, and convex set containing the origin, i.e., $0 \in \mathcal{X}$. We denote the diameter of $\mathcal{X}$ with R; i.e., $R := \max_{x,y \in \mathcal{X}} \|y - x\|$.

A2. For all $t \in [T]$, the utility function $f_t(\cdot)$ is normalized (i.e., $f_t(0) = 0$), monotone, DR-submodular, β_f-Lipschitz, and L-smooth. In other words, for all $x, y \in \mathcal{X}$ and $u \in \mathbb{R}^n$ where $u \succeq 0$ or $u \preceq 0$, the following holds:

$$f_t(x + u) - f_t(x) \geq \langle u, \nabla f_t(x) \rangle - \frac{L}{2} \|u\|^2$$

$$|f_t(y) - f_t(x)| \leq \beta_f \|y - x\|.$$

A3. For all $t \in [T]$, $p_t \in \mathbb{R}_+^n$ is i.i.d. generated from the distribution $\mathcal{D}$ with bounded support $\beta_p \mathcal{B} \cap \mathbb{R}_+^n$ (where $\mathcal{B}$ is the unit ball $\{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$), mean $p \succeq 0$ and covariance matrix Σ, i.e., $p_t \sim \mathcal{D}(p, \Sigma)$.

Let $\beta = \max\{\beta_f, \beta_p\}$. Under the above assumptions, we have:

$$F := \max_{t \in [T]} \max_{x,y \in \mathcal{X}} |f_t(x) - f_t(y)| \leq \beta R < \infty$$

$$G := \max_{p' \sim \mathcal{D}(p,\Sigma)} \max_{x \in \mathcal{X}} |\langle p', x \rangle - \frac{B_T}{T}| \leq \beta R - \frac{B_T}{T} < \infty.$$

5.3.1 Performance metric

We characterize the performance of our proposed algorithm by bounding the notions of regret and cumulative constraint violation which are defined below:

Definition 5.3.1 *The* $(1-\frac{1}{e})$*-regret is defined as:*

$$R_T = (1-\frac{1}{e})\max_{x\in\mathcal{X}^*}\sum_{t=1}^{T} f_t(x) - \sum_{t=1}^{T} f_t(x_t),$$

where:

$$\mathcal{X}^* = \{x \in \mathcal{X} : \sum_{t=1}^{T}\langle p, x\rangle \leq B_T\} = \{x \in \mathcal{X} : \langle p, x\rangle \leq \frac{B_T}{T}\}.$$

The regret metric R_T quantifies the difference between the reward obtained by the algorithm and the $(1-\frac{1}{e})$-approximation of the reward of the best fixed benchmark action that has access to all the utility functions f_t $\forall t \in [T]$, the *mean* p of the linear constraint functions, and satisfies the cumulative budget constraint. Note that $1-\frac{1}{e}$ is the optimal approximation ratio for offline continuous DR-submodular maximization; in other words, even if all the online input were available beforehand, we could only obtain a $(1-\frac{1}{e})$ fraction of the maximum reward in polynomial time. The $(1-\frac{1}{e})$-regret is commonly used in the online submodular maximization literature [61].

Definition 5.3.2 *The cumulative constraint violation is defined as follows:*

$$C_T = \sum_{t=1}^{T}\langle p, x_t\rangle - B_T.$$

Note that since p_t $\forall t \in [T]$ is i.i.d. drawn from the distribution $\mathcal{D}$ with mean p, our cumulative constraint violation metric C_T is defined with respect to the true underlying fixed linear constraint p (as opposed to p_t).

5.3.2 Related work

Consider the following general framework of online problems with long-term constraints: At round $t \in [T]$, the player chooses $x_t \in \mathcal{X}$. Then, cost (utility) function $f_t : \mathcal{X} \to \mathbb{R}$

Paper	Cost (utility)	Constraint	Window size	R_T	C_T
[213]	convex	convex (fixed)	—	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(T^{\frac{3}{4}})$
[206]	convex	convex (stochastic)	—	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\sqrt{T})$
[159]	convex	convex (adversarial)	1	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\sqrt{T})$
[135]$^{(a)}$	convex	convex (adversarial)	W	$\mathcal{O}(\sqrt{T} + \frac{WT}{V})$	$\mathcal{O}(\sqrt{VT})$
[181]	DR-submodular	linear (adversarial)	W	$\mathcal{O}(\sqrt{WT})$	$\mathcal{O}(W^{\frac{1}{4}}T^{\frac{3}{4}})$

Table 5.1: State-of-the-art results for online problems with cumulative constraints in various settings. Note that in (a), $V \in (W, T)$ is a tunable parameter.

($\mathcal{X}$ is a fixed convex set) and constraint function $g_t : \mathcal{X} \to \mathbb{R}$ are revealed, the player incurs a loss (obtains a reward) of $f_t(x_t)$ and uses the amount $g_t(x_t)$ of her budget (with the long-term constraint $\sum_{t=1}^{T} g_t(x_t) \leq 0$). This problem has been extensively studied under various assumptions where the cost (utility) functions are adversarially chosen and are assumed to be linear, convex, or DR-submodular and the constraint functions are linear or convex and are either fixed (i.e., $g_t(\cdot) = g(\cdot)\ \forall t \in [T]$), stochastic and i.i.d drawn from some unknown distribution, or adversarial. For the setting with adversarial utility and constraint functions, [143] provided a simple counterexample to show that regardless of the decisions of the algorithm, it is impossible to guarantee sub-linear regret against the benchmark action while the overall budget violation is sub-linear. Therefore, prior works in this setting have further restricted the fixed comparator action to be chosen from $\mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ 1 \leq t \leq T - W + 1\}$. In other words, in addition to merely satisfying the overall cumulative constraint (which corresponds to the $W = T$ case), the benchmark action is required to satisfy the budget constraint proportionally over any window of length W. On the other hand, for fixed or stochastic constraint functions, sub-linear regret and constraint violation bounds have been derived in the literature. A summary of the state-of-the-art results for online problems with long-term constraints is provided in

Table 5.1. After the publication of this work, [86] extended our problem framework to allow weakly DR-submodular utility functions and obtained high probability regret and constraint violation bounds for the problem.

5.4 Online Lagrangian Frank-Wolfe (OLFW) algorithm

In this section of the chapter, we first introduce our proposed algorithm, namely the Online Lagrangian Frank-Wolfe (OLFW) algorithm, in Section 5.4.1 and subsequently, we analyze the performance of the algorithm with high probability and in expectation in Section 5.4.2 and Section 5.4.3 respectively.

5.4.1 Algorithm

The Online Lagrangian Frank-Wolfe (OLFW) algorithm is presented in Algorithm 1. First, note that for all $t \in [T]$, $x_t = \frac{1}{K}\sum_{k=1}^{K} v_t^{(k)}$ is the average of vectors in the convex domain $\mathcal{X}$ and hence, $x_t \in \mathcal{X}$. The intuition for using K online maximization subroutines to update x_t is the Frank-Wolfe variant proposed in [25] to obtain the optimal approximation guarantee of $1 - \frac{1}{e}$ for solving the offline DR-submodular maximization problem without the additional linear constraints. To be more precise, consider the first iteration $t = 1$ of our online setting (ignoring the linear cumulative constraints) and the corresponding DR-submodular utility function $f_1(\cdot)$ arriving at this step. Note that f_1 is not revealed until the algorithm commits to an action $x_1 \in \mathcal{X}$. If we were in the offline setting, we could use the mentioned Frank-Wolfe variant of [25], run it for K iterations, and maximize f_1 over $\mathcal{X}$. Starting from $x_1^{(1)} = 0$, for all $k \in [K]$, we would find a vector $v_1^{(k)}$ that maximizes $\langle x, \nabla f_1(x_1^{(k)})\rangle$ over $x \in \mathcal{X}$, perform the update $x_1^{(k+1)} = x_1^{(k)} + \frac{1}{K} v_1^{(k)}$ and derive $x_1 = x_1^{(K+1)}$ as the output. However, in the online setting, the utility function f_1 is not available before committing to the action x_1. Therefore, for each $k \in [K]$, we instead use a separate instance of a no-regret online linear maximization algorithm to obtain $v_1^{(k)}$. We repeat the same process for the subsequent utility functions $\{f_t\}_{t>1}$. This intuition was first provided in [61] and they managed to obtain an $\mathcal{O}(\sqrt{T})$ regret bound for the unconstrained online monotone submodular maximization

Algorithm 1 Online Lagrangian Frank-Wolfe (OLFW)

```
**Input:** $\mathcal{X}$ is the constraint set, $T$ is the horizon, $\mu > 0$, $\delta > 0$, $\{\gamma_t\}_{t=1}^{T}$ and $K$.
**Output:** $\{x_t : 1 \leq t \leq T\}$.
Initialize $K$ instances $\{\mathcal{E}_k\}_{k \in [K]}$ of Online Gradient Ascent with step size $\mu$ for online maximization of linear functions over $\mathcal{X}$.
for $t = 1$ to $T$ do
  $x_t^{(1)} = 0$.
  for $k = 1$ to $K$ do
    Let $v_t^{(k)}$ be the output of oracle $\mathcal{E}_k$ from round $t-1$.
    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  end for
  Set $x_t = x_t^{(K+1)}$.
  Let $\widehat{p}_t := \frac{1}{t-1} \sum_{s=1}^{t-1} p_s$ for $t > 1$.
  Let
    $$\widetilde{g}_t(\cdot) = \begin{cases} \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} & \text{(I)} \\ \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} - \gamma_t & \text{(II)} \end{cases}.$$
  Set $\lambda_t = \frac{[\widetilde{g}_t(x_t)]_+}{\delta\mu}$ for $t > 1$ and 0 otherwise.
  Play $x_t$ and observe the Lagrangian function $\mathcal{L}_t(x_t, \lambda_t) = f_t(x_t) - \lambda_t \widetilde{g}_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$.
  for $k = 1$ to $K$ do
    Feedback $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) \rangle$ as the payoff to be received by $\mathcal{E}_k$.
  end for
end for
```

problem.

Our choice of Lagrangian function is inspired by the quadratic penalty method in constrained optimization [165]. The penalized formulation of the overall optimization problem (5.1) with quadratic penalty function could be written as follows:

$$\begin{aligned} \max_{x_t} \quad & \sum_{t=1}^{T} f_t(x_t) - \frac{1}{2\delta\mu}\big(\sum_{t=1}^{T}\langle p, x_t\rangle - B_T\big)^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Considering that the Fenchel conjugate of the function $h(\cdot) = \frac{1}{2\delta\mu}(\cdot)^2$ is $h^*(\cdot) = \frac{\delta\mu}{2}(\cdot)^2$, we can write the above problem in the following equivalent form:

$$\begin{aligned} \max_{x_t}\min_{\lambda} \quad & \sum_{t=1}^{T} f_t(x_t) - \lambda\big(\sum_{t=1}^{T}\langle p, x_t\rangle - B_T\big) + \frac{\delta\mu}{2}\lambda^2 \\ \text{subject to} \quad & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned}$$

Therefore, the corresponding Lagrangian function at round $t \in [T]$ is $\mathcal{L}_t(x, \lambda) = f_t(x) - \lambda(\langle p, x\rangle - \frac{B_T}{T}) + \frac{\delta\mu}{2}\lambda^2$. However, p is unknown to the online algorithm. Therefore, we alternatively use $\widehat{p}_t := \frac{1}{t-1}\sum_{s=1}^{t-1} p_s$ instead of p in the Lagrangian function. Note that $\widehat{p}_t$ is the empirical estimation of p at round t.

We first provide a lemma that is central to obtaining the regret and constraint violation bounds both in expectation and with high probability.

Lemma 5.4.1 *Let $x \in \mathcal{X}$ be a fixed vector. In the OLFW algorithm, set $\delta = \beta^2$. We have:*

$$\sum_{t=1}^{T}\big((1 - \frac{1}{e})f_t(x) - f_t(x_t)\big) \leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x). \tag{5.2}$$

5.4.2 Performance analysis with high probability

In order to analyze the performance of the OLFW algorithm with high probability, the following lemmas detailing the concentration inequalities for the stochastic linear constraints are used.

Lemma 5.4.2 *The following holds with probability at least* $1-\epsilon$*:*

$$\sum_{t=2}^{T} \|\widehat{p}_t - p\| \leq C\sigma\sqrt{T\log\big(\frac{2nT}{\epsilon}\big)}.$$

Lemma 5.4.3 *Let* $x \in \mathcal{X}$ *be fixed. Define* $\widehat{g}_t(x) := \langle \widehat{p}_t, x\rangle - \frac{B_T}{T}$ *and* $g(x) := \langle p, x\rangle - \frac{B_T}{T}$*. For a fixed* $t \in \{2,3,\dots,T\}$ *and* $\{\gamma_t := \sqrt{\frac{2G^2\log(\frac{2T}{\epsilon})}{t}}\}_{t=2}^{T}$*,* $|\widehat{g}_t(x) - g(x)| \leq \gamma_t$ *holds with probability at least* $1 - \frac{\epsilon}{T}$*.*

Proof First, note that $\mathbb{E}[\widehat{g}_t(x)] = \mathbb{E}[\langle \widehat{p}_t, x\rangle - \frac{B_T}{T}] = \langle p, x\rangle - \frac{B_T}{T} = g(x)$. If $y_t = g_t(x)$ is a random variable, then by assumption, $y_t \in [-G, G]$ holds for each t, i.e., y_t is a bounded random variable. Therefore we can apply Hoeffding's inequality and get $\mathbb{P}\{|\widehat{g}_t(x) - g(x)| > \gamma_t\} \leq 2\exp(-\frac{t\gamma_t^2}{2G^2})$. Substituting the value of γ_t in the right hand side, we get that $\mathbb{P}\{|\widehat{g}_t(x) - g(x)| > \gamma_t\} \leq \frac{\epsilon}{T}$. The result follows immediately. ■

Now, we have all the machinery to obtain the high probability performance bounds of the OLFW algorithm.

Theorem 5.4.1 ***(High probability regret bound)*** *Let* $\epsilon \in (0,1)$ *be given. Set* $\mu = \frac{R}{\beta\sqrt{T}}$*,* $K = \sqrt{T}$*,* $\delta = \beta^2$ *and* $\{\gamma_t\}_{t=2}^{T}$ *be chosen according to Lemma 5.4.3. Then, the OLFW algorithm with update (II) for* $\widetilde{g}_t(\cdot)$ *obtains the following regret bound with probability at least* $1-\epsilon$*.*

$$R_T \leq \big(\frac{LR^2}{2} + 2R\beta\big)\sqrt{T}.$$

Proof We begin from Lemma 5.4.1. Substitute the benchmark $x = x^*$ as the fixed vector in (5.2) and the constants as given in the hypothesis. We have $R_T \leq (\frac{LR^2}{2} + 2R\beta)\sqrt{T} + \sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x^*)$. Now let us bound $\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x^*)$. From Lemma 5.4.3, we have that with probability at least $1-\frac{\epsilon}{T}$, $\widehat{g}_t(x^*) - \gamma_t \leq g(x^*)$ holds, i.e., $\widetilde{g}_t(x^*) \leq g(x^*)$. Also, $g(x^*) \leq 0$ holds according to the definition of the benchmark action. Therefore, we have $\widetilde{g}_t(x^*) \leq 0$. As $\lambda_t \geq 0$, $\lambda_t\widetilde{g}_t(x^*) \leq 0$ holds. Now, taking union bound over all $t \in [T]$, we have with probability at least $1-\epsilon$ that $\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x^*) \leq 0$. The result follows immediately. ■

We will use the following lemma to get performance bounds for the constraint violation.

Lemma 5.4.4 *Let $\{\gamma_t\}_{t=2}^T$ be defined as in Lemma 5.4.3, then the following holds.*

$$C_T \leq \sum_{t=1}^T [\widetilde{g}_t(x_t)]_+ + R\sum_{t=1}^T \|\widehat{p}_t - p\| + \sum_{t=2}^T \gamma_t,$$

where $\widetilde{g}_t(\cdot)$ is derived using update (II).

Theorem 5.4.2 ***(High probability constraint violation bound)*** *Let $\epsilon \in (0,1)$ be given. Set $\mu = \frac{R}{\beta\sqrt{T}}$, $K = \sqrt{T}$, $\delta = \beta^2$ and $\{\gamma_t\}_{t=2}^T$ be chosen according to Lemma 5.4.3. Then the following holds with probability at least $1-\epsilon$ for the OLFW algorithm with update rule (II).*

$$C_T \leq \sqrt{2G^2T\log(\frac{2T}{\epsilon})} + CR\sigma\sqrt{T\log(\frac{2nT}{\epsilon})} + \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta) + R\beta.$$

So, we obtain $\tilde{\mathcal{O}}(\sqrt{T})$ constraint violation bound with high probability.

Proof We begin with Lemma 5.4.1 again but now substitute $x = 0$ as the fixed vector in (5.2).

$$\frac{B_T}{T}\sum_{t=1}^T \lambda_t + \sum_{t=1}^T \lambda_t\gamma_t \leq \underbrace{\sum_{t=1}^T f_t(x_t)}_{\leq FT} + \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T. \tag{5.3}$$

Rearranging and substituting the values of input parameters as given in the hypothesis, we get:

$$\sum_{t=1}^T [\widetilde{g}_t(x_t)]_+ + \frac{T\delta\mu}{B_T}\sum_{t=1}^T \lambda_t\gamma_t \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta).$$

Both terms in the left-hand side of the above equation are positive. Thus, we can drop the second term. We have:

$$\sum_{t=1}^T [\widetilde{g}_t(x_t)]_+ \leq \frac{T}{B_T}R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta).$$

Combining Lemma 5.4.4 and the equation above, we obtain:

$$C_T \leq \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta) + R\sum_{t=1}^{T} \|\widehat{p}_t - p\| + \sum_{t=2}^{T} \gamma_t.$$

Therefore, we can conclude:

$$C_T \leq \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta) + \sqrt{2G^2 T \log(\frac{2T}{\epsilon})} + \underbrace{R\sum_{t=1}^{T} \|\widehat{p}_t - p\|}_{\text{(A)}},$$

where the last inequality follows from summing γ_t's. Now, Lemma 5.4.2 tells us that (A) $\leq R\beta + CR\sigma\sqrt{T\log(\frac{2nT}{\epsilon})}$ holds with probability at least $1-\epsilon$. Thus, we obtain the desired result. ■

Theorem 5.4.1 and Theorem 5.4.2 are indeed the first high probability bounds obtained for the online DR-submodular maximization problem with stochastic cumulative constraints. Note that the $\mathcal{O}(\sqrt{T})$ regret bound obtained in Theorem 5.4.1 is known to be optimal.

5.4.3 Performance analysis in expectation

We first provide a lemma that will be used throughout the analysis in expectation.

Lemma 5.4.5 *For $t > 1$, we have:*

$$\mathbb{E}\|\widehat{p}_t - p\|^2 = \frac{Tr(\Sigma)}{t-1},$$

where $Tr(\Sigma)$ denotes the trace of the covariance matrix Σ.

Now, we present the main performance bounds in expectation, namely the expected regret bound and the expected cumulative constraint violation bound. In the Appendix, we have also considered the case where we only have access to unbiased stochastic gradient estimates of the utility functions $\{f_t\}_{t=1}^{T}$, and exact gradient computation is not possible. For this setting, we modify the OLFW algorithm by incorporating the variance reduction technique introduced by [60] and we obtain similar regret and constraint violation bounds in expectation for the modified algorithm.

Theorem 5.4.3 ***(Expected regret bound)*** *The regret bound of the OLFW algorithm with update rule (I) is the following:*

$$\mathbb{E}[R_T] \leq \tilde{\mathcal{O}}(T^{\frac{3}{4}}).$$

Proof We first observe from (5.3) that $\sum_{t=1}^{T} \lambda_t \leq \mathcal{O}(T)$. Now, substitute $x = x^*$, the benchmark, in (5.2) and take expectation on both sides to obtain:

$$\begin{aligned}
\mathbb{E}[R_T] &\leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \mathbb{E}[\sum_{t=1}^{T} \lambda_t(\widehat{g}_t(x^*) - g(x^*))] + \sum_{t=1}^{T} \lambda_t g(x^*) \\
&\leq \frac{LR^2T}{2K} + \frac{R^2}{\mu} + \beta^2\mu T + \underbrace{\mathbb{E}\,[\sum_{t=1}^{T} \lambda_t(\widehat{g}_t(x^*) - g(x^*))]}_{\text{(B)}}\,.
\end{aligned}$$

Now we bound (B) as follows:

$$\begin{aligned}
\text{(B)} &= \sum_{t=1}^{T} \lambda_t(\widehat{g}_t(x^*) - g(x^*)) \\
&\leq \sqrt{\sum_{t=1}^{T} \lambda_t^2}\sqrt{\sum_{t=1}^{T}(\widehat{g}_t(x^*) - g(x^*))^2} \\
&= \sqrt{\|\lambda\|^2}\sqrt{\sum_{t=1}^{T}(\langle \widehat{p}_t - p, x^*\rangle)^2} \\
&\leq \|\lambda\|\sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2 R^2} \\
&= R\|\lambda\|\sqrt{\sum_{t=1}^{T} \|\widehat{p}_t - p\|^2}.
\end{aligned}$$

Both the inequalities above are obtained using the Cauchy-Schwarz inequality, where $\lambda := [\lambda_1, \lambda_2, \ldots, \lambda_T]^T$.

Using the Cauchy-Schwarz inequality again, we have $\|\lambda\| \leq \sqrt{\|\lambda\|_1\|\lambda\|_\infty}$. Thus, we obtain $\|\lambda\| \leq \sqrt{(\sum_{t=1}^{T} \lambda_t)(\frac{G}{\delta\mu})} \leq \mathcal{O}(T^{3/4})$. Therefore, the following holds:

$$\|\lambda\| \leq \mathcal{O}(T^{3/4}). \tag{5.4}$$

Using Jensen's inequality, we have

$$\mathbb{E}\sqrt{\sum_{t=1}^{T}\|\widehat{p}_t - p\|^2} \leq \sqrt{\sum_{t=1}^{T}\mathbb{E}\|\widehat{p}_t - p\|^2}.$$

We can use Lemma 5.4.5 and write:

$$\mathbb{E}\sqrt{\sum_{t=1}^{T}\|\widehat{p}_t - p\|^2} \leq \sqrt{Tr(\Sigma)\log(T)}. \tag{5.5}$$

Thus, combining (5.4) and (5.5), we obtain:

$$\mathbb{E}[(\text{B})] \leq \mathcal{O}(T^{3/4}\sqrt{Tr(\Sigma)\log(T)}).$$

The result thus follows. ■

Remark 5.4.1 *The main challenge in bounding* R_T *in expectation is the fact that in our algorithm, the choice of* λ_t *is dependent on* $\widehat{p}_t$*, and thus, we cannot use*

$$\mathbb{E}[\lambda_t \widehat{g}_t(x^*)] = \mathbb{E}[\lambda_t \mathbb{E}[\widehat{g}_t(x^*)]] = \mathbb{E}[\lambda_t g(x^*)] \leq 0,$$

and this term is indeed the dominating term in the regret bound. However, as we saw earlier, we do not encounter this problem in the high probability setting due to subtracting γ_t *from all the constraint functions and using the concentration inequalities, and thus we were able to obtain* $\mathcal{O}(\sqrt{T})$ *high probability regret bound.*

Theorem 5.4.4 ***(Expected cumulative constraint violation bound)*** *For the OLFW algorithm with update rule (I), we have:*

$$\mathbb{E}[C_T] \leq \frac{T}{B_T}R\beta F\sqrt{T} + R\sqrt{Tr(\Sigma)}\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta) + R\beta.$$

Therefore, $\mathbb{E}[C_T] \leq \tilde{\mathcal{O}}(\sqrt{T})$ *holds.*

Theorem 5.4.3 and Theorem 5.4.4 provide the first sub-linear expectation bounds on the regret and cumulative constraint violation of the online DR-submodular maximization problem with stochastic cumulative constraints.

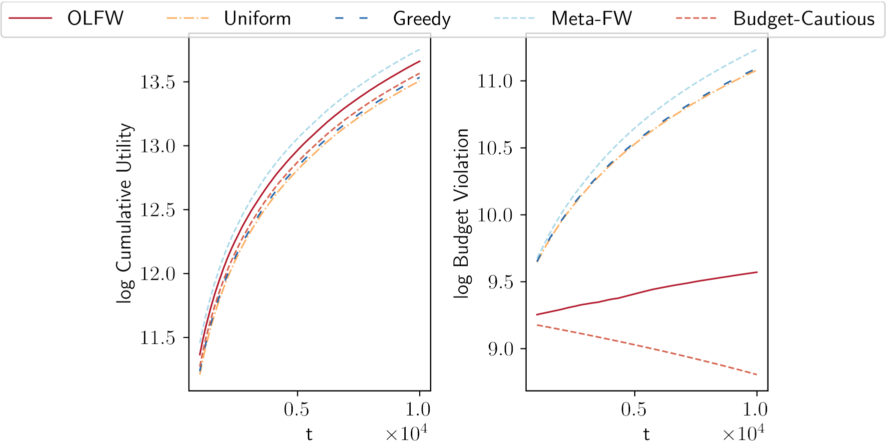


Figure 5.1: Comparison of the overall utility and cumulative budget violation over the Jester dataset.

5.5 Numerical results

We conduct numerical experiments, over simulated and real-world datasets in the following.

Joke Recommendation. We look at the problem of DR-submodular function maximization over the *Jester* dataset[1]. We consider a fraction of the dataset where there are 100 jokes and user ratings from 10000 users are available for these jokes. The ratings take values in $[-10, 10]$, we re-scale them to be in $[0, 10]$. Let $R_{u,j}$ be the rating of user u for joke j. As some of the user ratings are missing in the dataset, we set such ratings to be 5. In the online setting, a user arrives and we have to recommend at most $M = 15$ jokes to her. The utility function for each round $t \in [T]$ is of the form $f_t(x) = \sum_{i=1}^{100} R^t_{u_t,i} x_i + \sum_{i,j:i \neq j} \theta_{ij} x_i x_j$, where u_t is the user being served in the current round. $\{\theta_{ij}\}_{i \neq j}$ are chosen such that the function is monotone. These DR-submodular utility functions capture the overall impression of the displayed jokes on the user. There is a limited total time (denoted by $B_T = 1.5T$) available to recommend the jokes to the users. For all $i \in [n]$, p_i denotes the average time it

---

[1] http://eigentaste.berkeley.edu/dataset/

takes to read joke i. As some jokes are relatively longer, we do not want the user to spend more time on jokes that do not lead to larger utility. The linear budget functions are chosen randomly with entries uniformly drawn from $[0.03, 0.35]$. We compare the performance of our algorithm against the following strategies:

- *Uniform*: At every round, we assign 15 randomly chosen jokes to the user.
- *Greedy*: We deploy an exploration-exploitation strategy where with probability 0.1, we randomly assign 15 jokes, and with probability 0.9, we present the top 15 jokes based on the ratings observed so far.
- *Meta-FW* [61]: This corresponds to solving the unconstrained DR-submodular maximization problem (i.e., ignoring the budget constraints).
- *Budget-Cautious*: At each round, we assign 15 jokes that have the lowest average budget consumption observed so far.

The results are presented in Figure 5.1. As it can be seen in the plots, our OLFW algorithm obtains a reasonable utility while approximately satisfying the budget constraint as well.

Indefinite quadratic functions. We choose $\mathcal{X} = \{x \in \mathbb{R}^2 : 0 \preceq x \preceq 1\}$ and for each $t \in [T]$, we generate quadratic functions of the form $f_t(x) = \frac{1}{2}x^T H_t x + h_t^T x$ where $H_t \in \mathbb{R}^{2\times 2}$ is a random matrix whose entries are chosen uniformly from $[-1, 0]$. We let $h_t = -H_t^T \mathbf{1}$ to ensure the monotonicity of the objective. We let $T = 1000$. At each round, we randomly generate linear budget functions whose entries are chosen uniformly from $[0.5, 2.5]$ and the mean vector is $p = [1, 2]^T$. Also, we set the total budget to be $B_T = 2T$. We run the OLFW algorithm 10 times and take the respective averages for the cumulative utility and total remaining budget. We vary δ, the parameter of the penalty function, in the range $[0.1, 1000]$ and plot the trade-off curve $\big($i.e., $\sum_{t=1}^{1000} f_t(x_t)$ versus $B_T - \sum_{t=1}^{1000} \langle p, x_t \rangle\big)$ for 100 chosen values of δ in Figure 5.2. In this example, our choice of δ in the OLFW algorithm, highlighted in the plot, achieves the highest possible cumulative utility while satisfying the total budget constraint.

Log-determinant functions. We choose $\mathcal{X} = \{x \in \mathbb{R}^{10} : 0 \preceq x \preceq 1\}$ and for each $t \in [T]$, we generate log-determinant functions of the form $f_t(x) = \log\det\big(\mathrm{diag}(x)(L_t - I) + I\big)$, where each L_t is a random positive definite matrix with eigenvalues falling in the range $[2, 3]$. The choice of eigenvalues ensures the monotonicity of the function. Let $T = 4900$. At each round, we generate linear budget functions whose entries are chosen uniformly from the range $[0.3, 5.7]$. We run the OLFW algorithm for different choices of the step size μ and plot the cumulative utility and the total budget violation in Figure 5.3. Our OLFW algorithm chooses μ such that the overall utility and cumulative budget consumption are balanced.

5.6 Conclusion and future work

In this work, we studied online continuous DR-submodular maximization with stochastic linear cumulative constraints. We proposed the Online Lagrangian Frank-Wolfe (OLFW) algorithm to solve this problem and we obtained the first sub-linear bounds, both in expectation and with high probability, for the regret and constraint violation of this algorithm. The current work could be further extended in several interesting directions. First, it is yet to be seen whether the online DR-submodular maximization setting could handle general, stochastic, or adversarial, convex long-term constraints. Furthermore, it is interesting to see whether it is possible to improve the expected regret bound to match the $\mathcal{O}(\sqrt{T})$ high probability regret bound. Finally, studying this problem under bandit feedback (as opposed to the full information setting considered in this chapter) is left to future work.

After studying adversarial and stochastic linear long-term constraints separately in the previous chapter and this chapter respectively, we provide a unified framework in the next chapter and design a single algorithm that manages to either improve or match the aforementioned results without the knowledge of the regime. This algorithm can handle convex constraints as well.

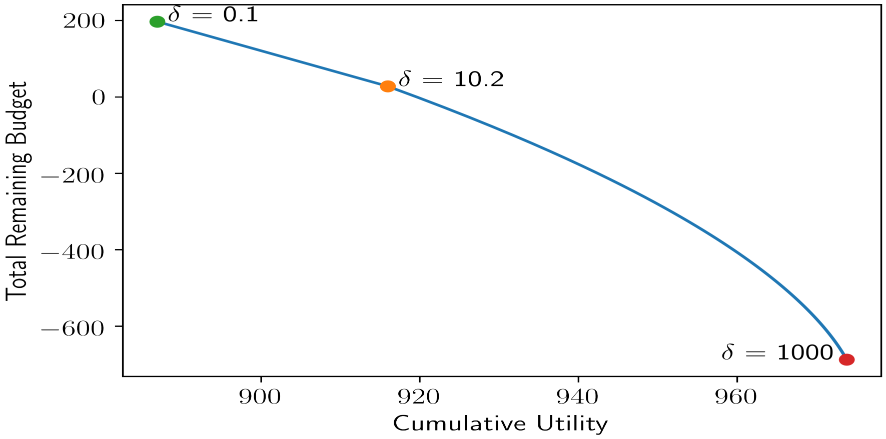


Figure 5.2: Trade-off between the overall utility and the total remaining budget of quadratic functions for different choices of parameter δ, $\delta = 10.2$ is our choice of the penalty parameter.

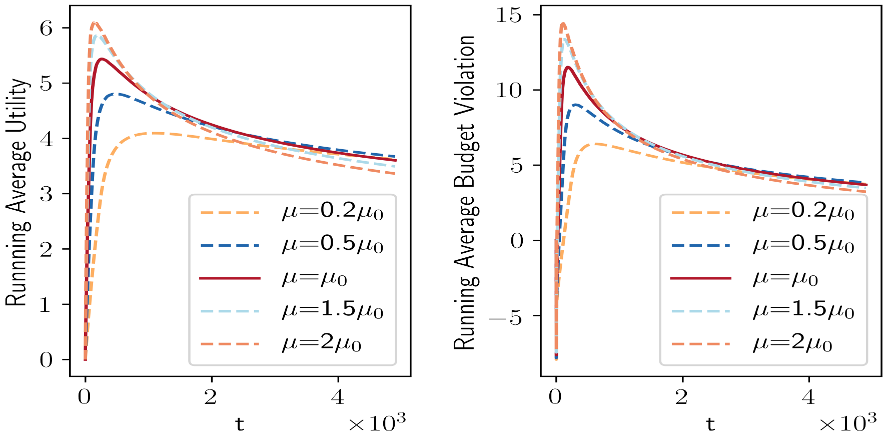


Figure 5.3: Running average of cumulative utility and running average of budget violation for different choices of the step size μ, where $\mu_0 := \frac{R}{\beta\sqrt{T}}$ is our choice of step size in the OLFW algorithm.

Chapter 6

A SINGLE RECIPE FOR ONLINE DR-SUBMODULAR MAXIMIZATION WITH ADVERSARIAL OR STOCHASTIC CONSTRAINTS

6.1 Chapter overview

This chapter is based on the joint work with Prasanna Raut and Maryam Fazel published at the Conference on Neural Information Processing Systems (NeurIPS) 2020. In this chapter, we consider an online optimization problem in which the reward functions are DR-submodular, and in addition to maximizing the total reward, the sequence of decisions must satisfy some convex constraints on average. Specifically, at each round $t \in \{1, \ldots, T\}$, upon committing to action x_t, a DR-submodular utility function $f_t(\cdot)$ and a convex constraint function $g_t(\cdot)$ are revealed, and the goal is to maximize the overall utility while ensuring the average of the constraint functions $\frac{1}{T}\sum_{t=1}^{T} g_t(x_t)$ is non-positive. Such cumulative constraints arise naturally in applications where the average resource consumption is required to remain below a prespecified threshold. We study this problem under an adversarial model and a stochastic model for the convex constraints, where the functions g_t can vary arbitrarily or according to an i.i.d. process over time slots $t \in \{1, \ldots, T\}$, respectively. We propose a single algorithm that achieves sub-linear (with respect to T) regret as well as sub-linear constraint violation bounds in both settings, without prior knowledge of the regime. Prior works have studied this problem in the special case of linear constraint functions. Our results not only improve upon the existing bounds under linear cumulative constraints but also give the first sub-linear bounds for general convex long-term constraints.

6.2 Introduction

Online optimization covers a large number of problems in which information is revealed incrementally (i.e., *online*), and irrevocable decisions should be made at each step in the face of uncertainty about the future arriving information [42, 146, 43, 218, 10]. Such problems could be formulated as a repeated game between the decision maker (i.e., the learner) and the adversary (i.e., nature or environment). At each iteration of this game, the learner chooses an action from a fixed domain set and then, it receives feedback in the form of utility or reward for her selected action. In the non-stochastic feedback model, no assumptions are made on the sequence of arriving rewards except their boundedness. As time goes by, the learner aims to observe the past and make better decisions to maximize the overall reward. The performance of online algorithms is usually measured through the *regret* or the *competitive ratio* of the algorithm. In the regret analysis framework, at each round, the learner has to commit to action before observing the corresponding reward function and the goal is to design algorithms whose total accumulated reward differs sub-linearly (in the time horizon T) from the reward of the best fixed benchmark action (or sequence) with hindsight information [218, 35, 189, 104]. On the other hand, in the competitive analysis setting, the decision maker is allowed to first observe the reward function at each step and then, choose her action accordingly (i.e., the *1-lookahead setting*). In this setting, the goal is to obtain bounds for the ratio of the total reward of the algorithm and the offline optimum (i.e., the competitive ratio) [44, 180]. In this work, we focus on the regret analysis setting.

In most of the prior work on online learning, there are no constraints on the sequence of decisions made by the learner, and maximizing the overall reward is the sole objective [10, 218]. However, in many applications, there indeed exist some constraints on the decisions of the algorithm which need to be satisfied on average [6, 7, 16, 18]. For instance, in an online task assignment problem in crowdsourcing markets, the requester needs to balance her total payment to workers against a prespecified allotted budget [192]. The advertiser in an online ad placement problem has a limited budget to invest in buying ads on different websites [135, 181, 174]. Note that in both of these problems, the resource (budget) consumption at each round is not known ahead of time. In crowdsourcing, the consumed resource depends on the workers' overall cost for performing the task, and even if a worker's hourly rate is known, the length of time required may not be known beforehand; in the online ad allocation problem, resource use depends on the number of clicks on the ads.

6.2.1 Related work

Online submodular maximization. Consider an online unconstrained optimization problem in which the reward functions are monotone DR-submodular. [61] proposed the Meta-Frank-Wolfe algorithm for this problem and obtained $\mathcal{O}(\sqrt{T})$ regret bound against the $(1-\frac{1}{e})$ approximation to the best fixed decision in hindsight where $(1-\frac{1}{e})$ is the best polynomial-time approximation ratio in the offline setting. The Meta-Frank-Wolfe algorithm requires access to the full gradient of the reward functions and performs $\mathcal{O}(\sqrt{T})$ gradient evaluations per step. More recently, [60] generalized this algorithm to the setting where only stochastic gradient estimates are available. Moreover, [216] proposed the Mono-Frank-Wolfe algorithm which performs only one gradient evaluation per round and requires only unbiased estimates of the gradient.

Online optimization with adversarial constraints. Online convex optimization with constraints, where both the convex objective functions $\{f_t\}_{t=1}^{T}$ and the convex constraint functions $\{g_t\}_{t=1}^{T}$ can vary arbitrarily, was first studied by [143]. They provided a surprisingly simple counterexample which showed that it is not always possible to achieve a sub-linear regret against the best fixed benchmark action in hindsight while the total constraint violation is sub-linear. Therefore, subsequent works added more assumptions to the problem setting to be able to obtain meaningful results. In particular, not only did they require the fixed benchmark action to satisfy the long-term constraint (i.e., $\sum_{t=1}^{T} g_t(x^*) \leq 0$), but they also restricted the benchmark to satisfy the constraint proportionally over any window of size $W \in [1, T]$. In other words, the fixed comparator action was required to be chosen from the set $\mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ 1 \leq t \leq T-W+1\}$. Note that if $W = T$, we recover the usual definition of the benchmark and the smaller W is, the comparator action is more restricted. See Table 6.1 for an overview of results under different choices of window length. Note that the setting in [181] is different, the objective functions are monotone DR-submodular (and generally non-concave) and the constraint functions are linear. More recently, this framework has been extended to distributed settings and bandit feedback as well [210, 211]. The performance of online convex optimization with adversarial convex constraints has also been analyzed against a dynamic benchmark sequence (i.e., *dynamic regret*), and sub-linear regret and constraint violation bounds have been derived under full and bandit feedback settings [47, 63].

Paper	Cost (utility)	Constraint	Window size	Regret	Constraint violation
[196]	convex	convex	1	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(T^{\frac{3}{4}})$
[159]	convex	convex	1	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\sqrt{T})$
[135]$^{(a)}$	convex	convex	W	$\mathcal{O}(\sqrt{T}+\frac{WT}{V})$	$\mathcal{O}(\sqrt{VT})$
[181]	DR-submodular	linear	W	$\mathcal{O}(\sqrt{WT})$	$\mathcal{O}(W^{\frac{1}{4}}T^{\frac{3}{4}})$

Table 6.1: Prior results for online problems with adversarial cumulative constraints in various settings. Note that in (a), $V \in (W, T)$ is a tunable parameter.

Online optimization with stochastic constraints. In light of the aforementioned impossibility result of [143], many subsequent works focused on stochastically time-varying constraints in which the constraint functions over $t \in [T]$ are assumed to be an i.i.d. process. In this setting, the benchmark action is required to satisfy the constraint in expectation, i.e., $\mathbb{E}[g_t(x^*)] \leq 0 \; \forall t \in [T]$. In this framework, [159, 212, 206] obtained $\mathcal{O}(\sqrt{T})$ regret and constraint violation bounds simultaneously, both in expectation and with high probability. Outside of the convex setting, [174] analyzed this problem for monotone DR-submodular utility functions and linear constraint functions. They managed to obtain $\mathcal{O}(\sqrt{T})$ constraint violation bound in expectation and with high probability. In addition, they derived $\mathcal{O}(T^{\frac{3}{4}})$ and $\mathcal{O}(\sqrt{T})$ regret bounds, in expectation and with high probability respectively. After the publication of our work, [86] extended our problem framework to allow weakly DR-submodular utility functions and obtained high probability regret and constraint violation bounds for the problem.

6.2.2 Contributions

In this chapter, we focus on a general class of online optimization problems where the reward functions $\{f_t\}_{t=1}^T$ are monotone DR-submodular and are chosen adversarially. Moreover, the constraint functions $\{g_t\}_{t=1}^T$ are monotone and convex. We study this problem in two settings. In the first setting, the constraint functions are assumed to vary arbitrarily. In the second model, we further restrict the sequence of constraint functions to be an i.i.d. process over time slots $t \in [T]$. We make the following contributions:

- Inspired by the Meta-Frank-Wolfe algorithm of [61] and the algorithm of [159], we propose Algorithm 1 in Section 6.4 for both adversarial or stochastic constraints without prior knowledge of the regime. In particular, for the adversarial setting, we obtain an $\mathcal{O}(T^{1-\frac{\epsilon}{2}})$ static regret bound against the benchmark with window length $W = T^{1-\epsilon}$ and an $\mathcal{O}(T^{1-\frac{\epsilon}{2}})$ total constraint violation bound. Moreover, if we consider dynamic regret as the utility performance metric, we obtain $\mathcal{O}(\sqrt{TP_T^*})$ bounds for both the dynamic regret and the total constraint violation where $P_T^* := \sum_{t=1}^{T-1} \left|\left|x_{t-1}^* - x_t^*\right|\right|$. In the setting with stochastic constraints, using the same algorithm (Algorithm 1), we obtain $\mathcal{O}(\sqrt{T})$ regret and total constraint violation bounds, both in expectation and with high probability.
- In Section 6.5, we propose Algorithm 2 which is based on the Mono-Frank-Wolfe algorithm of [216]. Compared to Algorithm 1, Algorithm 2 is computationally more efficient, but it achieves slightly worse performance guarantees. In particular, Algorithm 2 obtains an $\mathcal{O}(T^{\frac{2}{3}})$ static regret against the benchmark with window size $W \in [1, T^{\frac{1}{3}}]$ and an $\mathcal{O}(T^{\frac{2}{3}})$ total constraint violation bound. Similar bounds can also be derived in the stochastic setting, both in expectation and with high probability.

Lastly, we validate our theoretical findings and demonstrate the advantages of our proposed algorithms over prior work in a series of numerical experiments in Section 6.6.

6.3 Problem formulation

We consider the following protocol for online DR-submodular maximization with long-term convex constraints. At each iteration $t \in [T]$, the online algorithm chooses an action $x_t \in \mathcal{X}$ where $\mathcal{X}$ is the fixed domain set. Upon committing to this action, (i) a monotone DR-submodular utility function $f_t : \mathcal{X} \to \mathbb{R}$ is revealed and the algorithm obtains the reward $f_t(x_t)$ and (ii) a monotone convex constraint function $h_t : \mathcal{X} \to \mathbb{R}$ is revealed and $h_t(x_t)$ amount of resource is consumed. The total available resource is denoted by B_T which is given offline. Also, we assume the horizon T is known in advance, however, if T is not available offline, we can use the well-known doubling trick to obtain the same performance guarantees with slightly worse constants. The goal is to maximize the overall obtained reward while ensuring the resource constraint is satisfied on average, i.e., $\lim_{T\to\infty}\frac{1}{T}\big(\sum_{t=1}^{T} h_t(x_t) - B_T\big) \leq 0$. Denoting $g_t(\cdot) = h_t(\cdot) - \frac{B_T}{T}$, the resource constraint could be written as $\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} g_t(x_t) \leq 0$. In other words, we aim to maximize the overall utility while ensuring the total constraint violation $\sum_{t=1}^{T} g_t(x_t)$ grows sub-linearly in T. The offline optimization problem is as follows:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{t=1}^{T} f_t(x_t) \\ \text{subject to} \quad & \textstyle\sum_{t=1}^{T} g_t(x_t) \leq 0 \\ & x_t \in \mathcal{X} \ \forall t \in [T]. \end{aligned} \tag{6.1}$$

We study this problem under two settings. In the first online model, we assume that for all $t \in [T]$, both the utility function f_t and the constraint function h_t are chosen adversarially. In other words, we do not make any assumptions on the arriving functions f_t and h_t. In the second model, while the utility function f_t is still assumed to be arbitrary, the constraint function h_t is a random i.i.d. sample drawn from some unknown underlying distribution over a class $\mathcal{H}$ of monotone convex functions.

Note that our proposed algorithms can easily handle multiple online convex constraints with the same performance guarantees. However, for ease of notation, we focus on the special case of a single resource constraint.

To analyze this online optimization problem, we will make several assumptions that are common for online submodular problems.

Assumption 1. $\mathcal{X}$ is a convex and compact set with diameter R, and $0 \in \mathcal{X}$.

Assumption 2. For all $t \in [T]$, the reward function f_t is monotone, DR-submodular, β_f-Lipschitz, L-smooth along non-negative directions and normalized (i.e., $f_t(0) = 0$).

Assumption 3. For all $t \in [T]$, the constraint function h_t is monotone, convex, β_h-Lipschitz and normalized (i.e., $h_t(0) = 0$). In the stochastic setting, we assume that these assumptions hold for all $h \in \mathcal{H}$. Note that since g_t was defined as $g_t(\cdot) = h_t(\cdot) - \frac{B_T}{T}$, these assumptions apply to g_t as well.

Since $\mathcal{X}$ is compact and $f_t, h_t\ \forall t \in [T]$ are β-Lipschitz, where $\beta = \max\{\beta_f, \beta_h\}$, $f_t(\cdot)$ and $g_t(\cdot) = h_t(\cdot) - \frac{B_T}{T}$ are both bounded, i.e., $|f_t(x)| \le F$ and $|g_t(x)| \le G$ for all $x \in \mathcal{X}$ and $t \in [T]$.

6.3.1 Motivating Applications

There are a number of interesting applications that could be cast in our framework. In particular, we have described three examples below which have been used in Section 6.6 for the numerical experiments. See [181, 174] for more examples. In all the following applications, if the utility function is a submodular set function, we apply our algorithms to the DR-submodular continuous extension of the set function and use the lossless pipage rounding technique of [46] to make integral allocations.

Online joke recommendation. In this problem, we aim to design a joke recommendation algorithm to assign jokes to a sequence of users arriving online such that the overall impression of the jokes is maximized in a fixed time horizon B_T. At each step $t \in [T]$, a user arrives and the algorithm should assign a bundle of at most m jokes, $x_t \in \{x \in \{0,1\}^n : 1^T x \le m\}$, to her. If joke i is assigned to user t, she spends $[p_t]_i$ amount of time to read the joke and submit her rating $[r_t]_i$. In other words, $h_t(x) = \langle p_t, x\rangle$. The overall impression is the submodular set function $F_t(x) = r_t^T x + \sum_{i,j:i<j} \theta_{ij}^{(t)} x_i x_j$ where $\theta_{ij}^{(t)} \le 0$ penalizes the similarity of jokes i and j. This function has been extensively used in the literature to encourage diversity [136].

Online task assignment in crowdsourcing markets. In this problem, there exists a requester with a limited budget B_T that submits jobs and benefits from them being completed. There are n types of jobs available to be assigned to workers arriving online. At each step $t \in [T]$, a worker arrives and the requester has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}^n_+ : 0 \preceq x \preceq 1\}$ of the jobs to the worker. The worker has a private cost $[p_t]_i$ $\forall i \in [n]$ for performing one unit of the assigned job i, where $[p_t]_i$ denotes the i-th entry of vector p_t. In other words, we have $h_t(x) = \langle p_t, x\rangle$. The rewards obtained by the requester from this job assignment is a DR-submodular function $f_t(x) = \sum_{i=1}^{n}[u_t]_i \log(1 + x_i) + \sum_{i,j:i\neq j}[\theta_t]_{ij}x_i x_j$, where $[u_t]_i \geq 0$ and $[\theta_t]_{ij} \leq 0$. The DR-submodularity of the utility function captures the diminishing returns of assigning more jobs to the worker, i.e., as the number of assigned jobs to the worker increases, she has less time, energy, and resource available to devote to each fixed job $i \in [n]$ and therefore, the reward (quality of the completed task) obtained from the worker performing one unit of job i decreases. In other words, if $x \preceq y$, $\nabla_i f(x) \geq \nabla_i f(y)$ $\forall i \in [n]$ holds. The goal is to maximize the overall reward obtained by the requester while the budget constraint is satisfied on average.

Online welfare maximization with production cost. In this problem, there is a seller who has n types of products for sale that may be produced on demand using a fixed limited budget B_T. At each step $t \in [T]$, an agent (customer) arrives online and the seller has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}^n_+ : 0 \preceq x \preceq 1\}$ of products to the agent. The production cost for this assignment is $h_t(x_t) = x_t^T P_t x_t$, where P_t is an entry-wise non-negative positive definite matrix. Quadratic production cost functions with increasing gradients are commonly used in the literature [11, 54]. The agent has an unknown private DR-submodular valuation function $f_t(x) = \log\det\big(\text{diag}(x)(L_t - I) + I\big)$ over the items, where L_t is a positive semidefinite matrix and the DR-submodularity property characterizes the diversity of the assigned bundle. Therefore, the utility obtained by assigning the bundle x_t equals $f_t(x_t)$. The goal is to maximize the overall valuation of the agents while satisfying the budget constraint of the seller on average.

6.3.2 Benchmarks

We measure the performance of our proposed algorithms with the notions of *regret* and *total constraint violation* to quantify the overall utility and total resource consumption of the algorithms respectively. We define these notions below.

Total constraint violation. The total constraint violation of an online algorithm with outputs $\{x_t\}_{t=1}^T$ is the following:

$$C_T := \sum_{t=1}^{T} g_t(x_t) = \sum_{t=1}^{T} h_t(x_t) - B_T.$$

We aim to design algorithms whose total constraint violation is sub-linear in T.

In Online Convex Optimization (OCO), the utility performance of the algorithm is usually compared against *static* or *dynamic* benchmark sequences. Static regret metric has been extensively used in the literature [189, 104, 190, 194, 76, 215]. However, in problems where the environment is changing (i.e., dynamic), static regret is no longer a suitable measure and an enhanced measure, i.e., dynamic regret, is used [99, 214, 151, 111, 209]. In our setting, the regret metric should also specify how the benchmark actions behave with respect to the long-term adversarial or stochastic constraint. We thus introduce three notions of regret below.

Adversarial regret. In the adversarial setting, where both the utility and constraint functions are chosen arbitrarily, the $(1-\frac{1}{e})$-regret of an algorithm with outputs $\{x_t\}_{t=1}^T$ against a static benchmark action x_W^* with window length $W \in [1, T]$ is defined as:

$$R_{W,T}^{(A,S)} = (1-\frac{1}{e})\sum_{t=1}^{T} f_t(x_W^*) - \sum_{t=1}^{T} f_t(x_t),$$

where:

$$x_W^* = \arg\max_{x\in\mathcal{X}_W} \sum_{t=1}^{T} f_t(x),\ \ \mathcal{X}_W = \{x \in \mathcal{X} : \sum_{\tau=t}^{t+W-1} g_\tau(x) \leq 0,\ \ 1 \leq t \leq T-W+1\}.$$

Furthermore, in this adversarial setting, the $(1-\frac{1}{e})$-regret against a dynamic benchmark sequence $\{x_t^*\}_{t=1}^T$ is as follows:

$$R_T^{(A,D)} = (1-\frac{1}{e})\sum_{t=1}^{T} f_t(x_t^*) - \sum_{t=1}^{T} f_t(x_t),$$

where x_t^* is any benchmark action for which $g_t(x_t^*) \leq 0$ holds.

Stochastic regret. In the stochastic input model, where the utility functions $\{f_t\}_{t=1}^T$ are chosen adversarially and the constraint functions $\{h_t\}_{t=1}^T$ are drawn i.i.d. according to an unknown underlying distribution over the class $\mathcal{H}$, the $(1-\frac{1}{e})$-regret of an algorithm with outputs $\{x_t\}_{t=1}^T$ against a static benchmark action x^* is the following:

$$R_T^{(S,S)} = (1-\frac{1}{e})\sum_{t=1}^{T} f_t(x^*) - \sum_{t=1}^{T} f_t(x_t), \;\; x^* = \underset{x\in\mathcal{X}:\mathbb{E}[g_t(x)]\leq 0 \;\; \forall t\in[T]}{\arg\max} \sum_{t=1}^{T} f_t(x).$$

6.4 One practical algorithm for adversarial or stochastic constraints

In this section, we propose our first algorithm that could be applied to online DR-submodular maximization problems with both adversarial or stochastic constraints without prior information about the regime. The algorithm is provided in Algorithm 1. The algorithm generalizes that of [159] for the convex setting to handle generally non-concave DR-submodular utility functions. In particular, inspired by the Meta-Frank-Wolfe algorithm of [61], we have changed the primal update of the algorithm of [159] to be able to obtain regret bounds in this setting. The following lemma provides an equivalent formulation of the primal update of the algorithm.

Lemma 6.4.1 *For all $t \in [T]$ and $k \in [K]$, the update rule of Algorithm 1 for $v_t^{(k)}$ is equivalent to the following:*

$$v_t^{(k)} = \mathbb{P}_{\mathcal{X}}\Big(v_{t-1}^{(k)} + \frac{1}{2\alpha}\big(V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)})\big)\Big),$$

where $\mathbb{P}_{\mathcal{X}}$ denotes the projection onto set $\mathcal{X}$.

Thus, for each $k \in [K]$, the algorithm runs an instance of online gradient ascent with step size $\frac{1}{2\alpha}$ to choose the point $v_t^{(k)}$, $\forall t \in [T]$, and upon committing to this action, it receives a reward of $\langle V\nabla f_t(x_t^{(k)}), v_t^{(k)}\rangle - \lambda_{t+1}^{(k)} g_t(v_t^{(k)})$. Note that using the average of the output of K online maximization algorithms to obtain $\{x_t\}_{t=1}^T$ is common in online submodular maximization [61, 60, 216]. The parameter V characterizes the trade-off between maximizing

Algorithm 1

```
Input: $\mathcal{X}$ is the constraint set, $T$ is the horizon, $V > 0$, $\alpha > 0$ and $K \in \mathbb{N}$.
Output: $\{x_t : 1 \leq t \leq T\}$.
Initialize $\lambda_1^{(k)} = v_0^{(k)} = x_0^{(k)} = 0 \ \forall k \in [K]$.
for $t = 1$ to $T$ do
  $x_t^{(1)} = 0$.
  for $k = 1$ to $K$ do
    $v_t^{(k)} = \arg\max_{x \in \mathcal{X}} \left( \langle V \nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)} \nabla g_{t-1}(v_{t-1}^{(k)}), x \rangle - \alpha \left|\left| x - v_{t-1}^{(k)} \right|\right|^2 \right)$,
    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  end for
  Set $x_t = x_t^{(K+1)}$ and play $x_t$.
  Observe the utility function $f_t$ and the constraint function $g_t$.
  for $k = 1$ to $K$ do
    $\lambda_{t+1}^{(k)} = [\lambda_t^{(k)} + g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle]_+$.
  end for
end for
```

the reward and satisfying the resource constraint. In other words, choosing a larger V leads to a higher overall reward while the constraint is further violated. The output of the algorithm at each step, $x_t, \ \forall t \in [T]$, is the average of K vectors $v_t^{(k)}$ in the convex domain $\mathcal{X}$; hence, $x_t \in \mathcal{X}$ also holds.

Furthermore, the algorithm needs to maintain K dual variables at every time step t, $\lambda_t^{(k)}$ $k \in [K]$ (as opposed to a single dual variable in [159]). To better understand the algorithm, we first provide the following two lemmas.

Lemma 6.4.2 *The cumulative constraint violation of Algorithm 1 could be bounded as follows:*

$$C_T \leq \frac{1}{K} \sum_{k=1}^{K} \lambda_{T+1}^{(k)} + \frac{\beta^2 T}{4V} + \frac{V}{K} \sum_{k=1}^{K} \sum_{t=1}^{T} \left|\left| v_t^{(k)} - v_{t-1}^{(k)} \right|\right|^2 .$$

Lemma 6.4.3 *Let* $\Delta_t^{(k)} := \frac{(\lambda_{t+1}^{(k)})^2}{2} - \frac{(\lambda_t^{(k)})^2}{2}$ *for all* $t \in [T]$ *and* $k \in [k]$*. We have:*

$$\Delta_t^{(k)} \leq \frac{(G+\beta R)^2}{2} + \lambda_t^{(k)}\big(g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle\big).$$

$\Delta_t^{(k)}$ has been commonly used in the literature and is called Lyapunov quadratic drift [158]. The result of Lemma 6.4.2 suggests that in order to minimize the total constraint violation C_T, the algorithm needs to maintain small dual variables. Equivalently, for all $t \in [T]$ and $k \in [K]$, the drift of the dual variable , $\Delta_t^{(k)}$, needs to be minimized. Using the result of Lemma 6.4.3, we obtain:

$$\begin{aligned}&\Delta_t^{(k)} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle + \alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \\ &\leq \underbrace{\frac{(G+\beta R)^2}{2} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle + \lambda_t^{(k)} g_{t-1}(v_{t-1}^{(k)}) + \alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2}_{\text{(a)}}.\end{aligned}$$

In Algorithm 1, $v_t^{(k)}$ is chosen to be the minimizer of (a) over $\mathcal{X}$ and the update rule for $\lambda_{t+1}^{(k)}$ corresponds to moving along the direction of the gradient of (a) with respect to the dual variable.

6.4.1 Performance guarantees

In this section, we provide the total constraint violation and regret bounds under different settings.

Theorem 6.4.1 ***(Total constraint violation bound)*** *The total constraint violation of Algorithm 1 is bounded as follows:*

$$C_T \leq \theta V + \frac{\beta^2 T}{4V} + \frac{\beta^2(1+\theta)^2 V^3 T}{4\alpha^2},$$

where $\theta = \max\{G+\beta R, \frac{\frac{(G+\beta R)^2}{2} + (\beta R + \frac{V\beta^2}{4\alpha})V}{VB_T/T} + \frac{\alpha R^2}{V(V+1)B_T/T} + \frac{(G+\beta R)(V+2)}{2V}\}$*. In particular, if* $\alpha \leq \mathcal{O}(V^2)$*, we have* $\theta = \mathcal{O}(1)$*.*

Theorem 6.4.1 characterizes the total constraint violation bound of Algorithm 1 in both adversarial and stochastic settings.

Theorem 6.4.2 ***(Adversarial static regret bound)*** *The regret of Algorithm 1 in the adversarial setting against a benchmark with window length W is bounded as:*

$$R_{W,T}^{(A,S)} \leq F(W-1) + \frac{1}{2V}\min\{\theta^2 V^2, (G+\beta R)^2 \frac{(W-1)(2W-1)}{6}\} + \frac{V\beta^2(T-W+1)}{4\alpha} + \frac{(G+\beta R)^2(T-W+1)}{2V} + \frac{LR^2(T-W+1)}{2K} + \frac{(G+\beta R)^2(W-1)(T-W+1)}{2V} + \frac{\alpha R^2}{V}.$$

Therefore, setting $\alpha = V\sqrt{T}$, the adversarial static regret bound of Algorithm 1 is $\mathcal{O}(\frac{WT}{V} + \sqrt{T} + \frac{T}{K})$. In particular, for the adversarial setting with window size $W = T^{1-\epsilon}$, if we choose $V = \mathcal{O}(T^{1-\frac{\epsilon}{2}})$, $\alpha = V\sqrt{T}$ and $K = \mathcal{O}(T^{\frac{\epsilon}{2}})$ in Theorem 6.4.1 and Theorem 6.4.2, we have $R_{W,T}^{(A,S)} \leq \mathcal{O}(T^{1-\frac{\epsilon}{2}})$ and $C_T \leq \mathcal{O}(T^{1-\frac{\epsilon}{2}})$. In comparison, [181] obtains a similar $\mathcal{O}(T^{1-\frac{\epsilon}{2}})$ regret bound and a worse $\mathcal{O}(T^{1-\frac{\epsilon}{4}})$ total constraint violation bound, and only for the special case of linear constraint functions.

Theorem 6.4.3 ***(Adversarial dynamic regret bound)*** *The adversarial regret of Algorithm 1 against a dynamic benchmark sequence $\{x_t^*\}_{t=1}^T$ is bounded as follows:*

$$R_T^{(A,D)} \leq \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K} + \frac{2\alpha R P_T^*}{V},$$

where $P_T^* := \sum_{t=1}^{T-1} ||x_{t-1}^* - x_t^*||$.

If we set $V = K = \mathcal{O}(\sqrt{\frac{T}{P_T^*}})$ and $\alpha = V^2$, we have $R_T^{(A,D)} \leq \mathcal{O}(\sqrt{TP_T^*})$ and $C_T \leq \mathcal{O}(\sqrt{TP_T^*})$. However, since P_T^* is not known ahead of time, the parameters of the algorithm cannot be chosen as mentioned. To remedy this issue, we can extend the adaptive algorithm of [214] to our framework to obtain $\mathcal{O}(\sqrt{TP_T^*})$ regret and total constraint violation bounds simultaneously without prior knowledge of P_T^*.

Theorem 6.4.4 ***(Expected Regret Bound)*** *In the stochastic setting, the expected regret of Algorithm 1 could be bounded as follows:*

$$\mathbb{E}[R_T^{(S,S)}] \leq \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K}.$$

Theorem 6.4.5 ***(High Probability Regret Bound)*** *The regret of Algorithm 1 satisfies the following with probability at least $1-\delta$ in the stochastic setting:*

$$R_T^{(S,S)} \leq \theta G\sqrt{2T\log(\frac{1}{\delta})} + \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K}.$$

If we choose $V = \mathcal{O}(\sqrt{T})$, $\alpha = V^2$ and $K = \mathcal{O}(\sqrt{T})$ in Theorem 6.4.1, Theorem 6.4.4 and Theorem 6.4.5, we have $\mathbb{E}[R_T^{(S,S)}] \leq \mathcal{O}(\sqrt{T})$, $R_T^{(S,S)} \leq \tilde{\mathcal{O}}(\sqrt{T})$ w.h.p. and $C_T \leq \mathcal{O}(\sqrt{T})$. In comparison, for the special case of linear constraint functions, [174] obtains similar $\mathcal{O}(\sqrt{T})$ bounds for the total constraint violation, both in expectation and with high probability. However, despite achieving $\mathcal{O}(\sqrt{T})$ high probability regret bound, their algorithm obtains a worse $\mathcal{O}(T^{\frac{3}{4}})$ regret bound in expectation.

6.5 Trading performance for efficiency: A faster algorithm

In this section, we propose our second algorithm, presented in Algorithm 2 in the appendix, for online DR-submodular maximization problems with adversarial or stochastic constraints. Inspired by the Mono-Frank-Wolfe algorithm of [216], we divide the upcoming online rounds $1, \dots, T$ to Q equisized blocks of length K (i.e., $T = QK$) and for all the rounds $t \in \{(q-1)K + 1, \dots, qK\}$ in the block $q \in Q$, we play the same action x_q. Using this technique, the computational complexity of Algorithm 2 reduces with a factor of K compared to Algorithm 1. However, the efficiency of Algorithm 2 comes at the price of slightly worse regret and total constraint violation bounds, as presented below.

Theorem 6.5.1 *The adversarial (static) regret bound of Algorithm 2 against the benchmark with window length $W \in [1, T^{\frac{1}{3}}]$ is as follows:*

$$\mathbb{E}[R_{W,T}^{(A,S)}] \leq \frac{V\beta^2 QK}{4\alpha} + \frac{(G + \beta R)^2 QK}{2V} + \frac{\alpha R^2 K}{V} + \frac{LR^2 Q}{2},$$

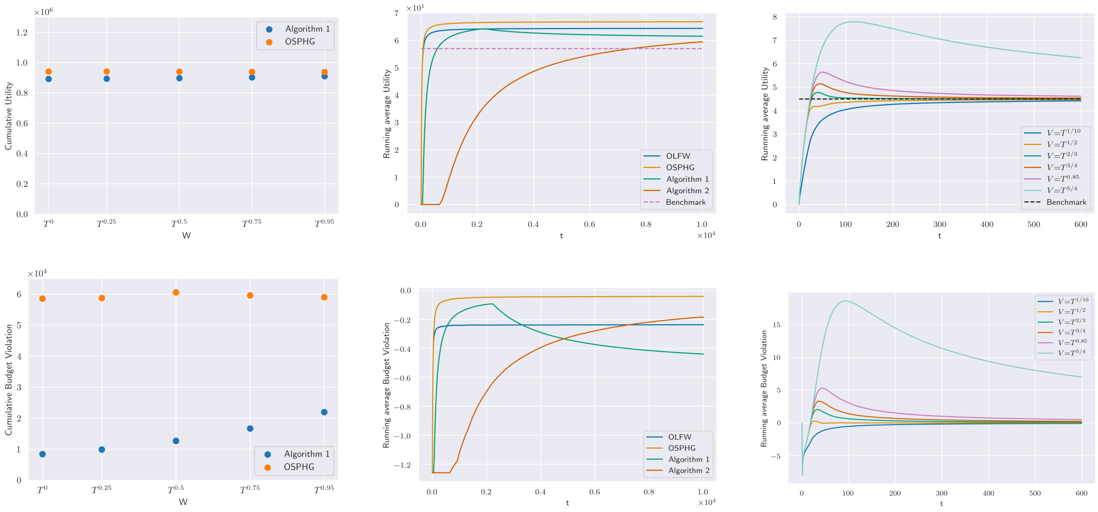
where the expectation is taken with respect to the randomness of the algorithm.

Theorem 6.5.2 *The total constraint violation of Algorithm 2 is bounded as below:*

$$\mathbb{E}[C_T] \leq \theta KV + \frac{\beta^2 T}{4V} + \frac{\beta^2 (1+\theta)^2 V^3 T}{4\alpha^2},$$

where the expectation is taken with respect to the randomness of the algorithm.

Thus, in the adversarial setting with window length $W \in [1, T^{\frac{1}{3}}]$, if we choose $V = K = \mathcal{O}(T^{\frac{1}{3}})$ and $\alpha = V^2$ in Theorem 6.5.1 and Theorem 6.5.2, we have $\mathbb{E}[R_{W,T}^{(A,S)}] \leq \mathcal{O}(T^{\frac{2}{3}})$ and



(a) (b) (c)

Figure 6.1: (a) Cumulative utility and total budget violation for experiment 1, (b) and (c) Running average of cumulative utility and budget violation for experiments 2 and 3 respectively.

$\mathbb{E}[C_T] \leq \mathcal{O}(T^{\frac{2}{3}})$.

Note that similar to the analysis in Section 6.4, we can extend the above results to a benchmark with general window length in the adversarial setting. Moreover, we can obtain similar $\mathcal{O}(T^{\frac{2}{3}})$ regret and total constraint violation bounds in the stochastic setting.

6.6 Experiments

In order to verify our theoretical findings, we run our algorithms for the three experiments described in Section 6.3.1 and we plot the performance in Figure 6.1.

1) *Online joke recommendation.* We choose $n = 100$ jokes, $T = 10000$ and $B_T = 1.5T$. We vary the window length W and choose V, α and K according to Section 6.4. We set $\mathcal{X} = \{x \in [0, 1]^n : 1^T x \leq 15\}$. We consider the utility functions $f_t(x) = r_t^T x + \sum_{i,j:i<j} \theta_{ij}^{(t)} x_i x_j \ \forall t \in [T]$

where $0 \leq [r_t]_i \leq 10$ is the rating of user t for joke i in the *Jester* dataset[1], and $\theta_{ij}^{(t)}$ is uniformly chosen from $[-0.5, 0]$. Also, $[p_t]_i$ is chosen uniformly from the range $[0.3, 6]$. We compare the overall utility and total budget violation of Algorithm 1 and the OSPHG algorithm of [181] for different choices of W. Note that we use the pipage rounding technique of [46] for both algorithms to make an integral allocation of jokes to the users. Figure 6.1(a) verifies the superiority of Algorithm 1 compared to the OSPHG algorithm in terms of budget consumption while obtaining similar overall utility.

2) *Online task assignment in crowdsourcing markets.* We set $n = 13$, $T = 10000$ and $B_T = 0.86T$. We choose V, α and K according to Section 6.4. We set $\mathcal{X} = \{x : 0 \preceq x \preceq 1\}$. We consider the utility functions $f_t(x) = \sum_{i=1}^{n} [u_t]_i \log(1 + x_i) + \sum_{i,j:i<j} \theta_{ij}^{(t)} x_i x_j \; \forall t \in [T]$ where $[u_t]_i$ and $\theta_{ij}^{(t)}$ are uniformly chosen from $[1, 13]$ and $[-0.07, 0]$ respectively. $[p_t]_i$ is uniformly chosen from $[0.05, 1]$. We compare the average performance of Algorithm 1, Algorithm 2, the OLFW algorithm of [174] and the OSPHG algorithm of [181]. Figure 6.1(b) demonstrates that Algorithm 1 strikes the right balance between the utility and budget used.

3) *Online welfare maximization with production cost.* For this experiment, we use the utility function $f_t(x) = \log\det\big(\operatorname{diag}(x)(L_t - I) + I\big)$ and the quadratic convex constraint function $h_t(x) = x^T P_t x$ for all $t \in [T]$, where L_t and P_t are positive definite matrices whose eigenvalues are uniformly chosen from $[2, 3]$ and $[0.3, 6]$ respectively. We consider the domain $\mathcal{X} = \{x : 0 \preceq x \preceq 1\}$. We set $n = 10$, $T = 1000$, $K = W = \sqrt{T}$ and $B_T = 4T$. We vary V and choose $\alpha = V\sqrt{T}$ to see the effect of the choice of V in the performance of Algorithm 1. Considering the $\mathcal{O}(\frac{WT}{V} + \sqrt{T} + \frac{T}{K})$ regret bound and $\mathcal{O}(V + \frac{T}{V})$ total constraint violation bound derived in Section 6.4, Figure 6.1(c) verifies the theoretical analysis that we need to choose $V \in (W, T)$ to obtain sub-linear regret and total constraint violation bounds simultaneously.

---

[1] http://eigentaste.berkeley.edu/dataset/

6.7 Conclusion

We studied an online optimization problem in which the reward functions are monotone DR-submodular, and in addition, the sequence of decisions of the learner should satisfy some adversarially or stochastically varying monotone convex constraints on average. We propose a single algorithm for both adversarial or stochastic constraints without prior knowledge of the regime. In the special case of linear constraint functions, our proposed algorithm obtains improved regret and constraint violation bounds in both adversarial and stochastic settings compared to prior work. Moreover, we derive the first sub-linear bounds for the more general case of convex constraint functions.

Part II

STRONGLY DR-SUBMODULAR MAXIMIZATION

Chapter 7

OFFLINE STRONGLY DR-SUBMODULAR MAXIMIZATION

7.1 Chapter overview

In this chapter, we present the joint work with Maryam Fazel that was published at the SIAM Conference on Applied and Computational Discrete Algorithms (ACDA) 2023. Continuous DR-submodular functions are a class of functions that satisfy the Diminishing Returns (DR) property, which implies that they are concave along non-negative directions. Existing works have studied monotone continuous DR-submodular maximization subject to a convex constraint and have proposed efficient algorithms with approximation guarantees. However, in many applications, e.g., computing the stability number of a graph and mean-field inference for probabilistic log-submodular models, the DR-submodular function has the additional property of being *strongly* concave along non-negative directions that could be utilized for obtaining faster convergence rates. In this chapter, we first introduce and characterize the class of *strongly DR-submodular* functions and show how such a property implies strong concavity along non-negative directions. Then, we study L-smooth monotone strongly DR-submodular functions that have bounded curvature, and we show how to exploit such additional structure to obtain algorithms with improved approximation guarantees and faster convergence rates for the maximization problem. In particular, we propose the SDRFW algorithm that matches the provably optimal $1 - \frac{c}{e}$ approximation ratio after only $\lceil \frac{L}{\mu} \rceil$ iterations, where $c \in [0, 1]$ and $\mu \geq 0$ are the curvature and the strong DR-submodularity parameter. Furthermore, we study the Projected Gradient Ascent (PGA) method for this problem and provide a refined analysis of the algorithm with an improved $\frac{1}{1+c}$ approximation ratio (compared to $\frac{1}{2}$ in prior works) and a linear convergence rate. Given that both algorithms require knowledge of the smoothness parameter L, we provide a *novel* characterization of L for DR-submodular functions showing that in many cases, computing L could be formulated as a convex optimization problem, i.e., a geometric program, that could be solved efficiently. Experimental results illustrate and validate the efficiency and effectiveness of our algorithms.

7.2 Introduction

Continuous DR-submodular functions find applications in multiple domains such as influence and revenue maximization, MAP inference for DPP (Determinantal Point Process), and mean-field inference of probabilistic graphical models (see [27, Section 6] for more applications and details). While DR-submodular functions are generally non-convex/non-concave, the DR property provides a natural structure that allows the designing of tractable approximation algorithms. In particular, DR-submodular functions are concave along non-negative directions [25], i.e., for all x, y such that $x_i \leq y_i \ \forall i \in [n]$, we have $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle$.

Monotone DR-submodular maximization subject to a convex constraint has been previously studied in the literature. [25] proposed a Frank-Wolfe variant that obtains a provably optimal $1 - \frac{1}{e}$ approximation guarantee at a sub-linear convergence rate. Later, [102] studied the well-known Projected Gradient Ascent (PGA) method for this problem and proved that PGA has a $\frac{1}{2}$ approximation ratio and sub-linear rate of convergence.

A number of the DR-submodular objective functions in the applications above of the continuous DR property are indeed strongly concave along non-negative directions. For instance, consider a graph $G = (V, E)$ with adjacency matrix A. Computing the stability number $s(G)$ of the graph (i.e., the cardinality of the largest subset of vertices such that no two vertices in this subset are adjacent) is a well-known NP-hard combinatorial problem. This problem could be formulated as $s(G)^{-1} = \min_{x \in \Delta} x^T (A + I) x$ where $\Delta = \{x \in \mathbb{R}^{|V|} : 1^T x = 1, x_v \geq 0 \ \forall v \in V\}$ is the standard simplex and I is the identity matrix [155]. We can rewrite the problem as:

$$\begin{aligned} s(G)^{-1} &= \min_{x\in\Delta} x^T(A+I)x \\ &= -\max_{x\in\Delta} x^T(-A-I)x \\ &= 2 - \max_{x\in\Delta} x^T(-A-I)x + \underbrace{21^T x}_{=2}. \end{aligned}$$

It will soon be clear that the function $x^T(-A-I)x + 21^T x$ is 2-strongly DR-submodular (see Definition 7.3.1 in Section 7.3) and thus, finding the stability number of a graph could be formulated as a convex-constrained monotone 2-strongly DR-submodular maximization problem. Another example is the mean-field inference problem for probabilistic log-submodular models (Section 6.6 of [27]). Let $F : 2^V \to \mathbb{R}$ be a submodular set function. Consider distributions over subsets $S \subseteq V$ of the form $P(S) = \frac{1}{Z}\exp(F(S))$ where the normalizing quantity $Z = \sum_{S\subseteq V} \exp(F(S))$ is called the *partition function*. In general, computing the partition function involves summing over an exponential number of terms and is not computationally feasible. Alternatively, one can use mean-field inference to approximate $P(S)$ by a completely factorized distribution $Q(S)$, i.e., elements $i \in V$ are picked independently and $Q(S|x) = \prod_{i\in S} x_i \prod_{j\notin S}(1-x_j)$ where $x \in [0,1]^V$ is the vector of marginals, by minimizing the KL divergence $\mathbf{KL}(x) = \sum_{S\subseteq V} Q(S|x)\ln\big(\frac{Q(S|x)}{p(S)}\big)$ between the two distributions. $\mathbf{KL}(x)$ can be written as:

$$\mathbf{KL}(x) = -\sum_{S\subseteq V} F(S)\prod_{i\in S} x_i \prod_{j\notin S}(1-x_j) + \sum_{i=1}^{|V|}\big(x_i\ln x_i + (1-x_i)\ln(1-x_i)\big) + \ln Z.$$

It will be clear later that the problem $\max_{x\in[0,1]^V} -\mathbf{KL}(x)$ (equivalent to minimizing $\mathbf{KL}(x)$) is a 4-strongly DR-submodular maximization problem.

7.2.1 Contributions

In this chapter, we precisely characterize the class of *strongly DR-submodular* functions, i.e., DR-submodular functions that are strongly concave along non-negative directions. We consider the optimization problem $\max_{x\in\mathcal{K}} f(x)$ where $\mathcal{K}$ is a convex set and f is a

Paper	Setting	Approximation ratio	Convergence rate
[25]	$\mu = 0$	$1 - \frac{1}{e}$	sub-linear
[102]	$\mu = 0$	$\frac{1}{2}$	sub-linear
This work (PGA)	$\mu = 0$	$\frac{1}{1+c_f}$	sub-linear
This work (PGA)	$\mu > 0$	$\frac{1}{1+c_f}$	linear
This work (SDRFW)	$\mu > 0$	$1 - \frac{c_f}{e}$	after $\lceil \frac{L}{\mu} \rceil$ iterations

Table 7.1: Comparison of our results and the prior works.

monotone, smooth and strongly DR-submodular function with bounded curvature $c_f \in [0, 1]$, and we provide algorithms with refined approximation guarantees that exploit the strong DR-submodularity structure of the objective function. Specifically, we make the following contributions:

- We propose the SDRFW algorithm in Section 7.4 that obtains the approximation guarantee $1 - \frac{c_f}{e}$ after $\lceil \frac{L}{\mu} \rceil$ iterations, where L is the smoothness parameter and μ is the strong DR-submodularity parameter of the function f.
- In Section 7.5, we analyze PGA for our problem and we provide a refined and sharper analysis of the algorithm showing that PGA has an improved $\frac{1}{1+c_f}$ approximation guarantee (compared to $\frac{1}{2}$ in prior works) at a linear convergence rate, i.e., it only takes $\mathcal{O}(\ln(\frac{1}{\epsilon}))$ iterations to get within ϵ of $\frac{1}{1+c_f}$OPT, where OPT denotes the optimal value of the problem.
- We also study *online* strongly DR-submodular maximization in Section 7.5.1. We analyze the online counterpart of PGA, called Online Gradient Ascent (OGA), for this problem and we show how our techniques could be used to obtain improved logarithmic bounds for the $(\frac{1}{1+c})$-regret of the algorithm.
- Given that both SDRFW and PGA require knowledge of the smoothness parameter L, in Section 7.6, we provide a *novel* characterization of L for (strongly) DR-submodular functions showing that in many cases, computing L could be formulated as a convex optimization

problem, i.e., a geometric program, that could be solved efficiently. Therefore, we can efficiently compute L before running the algorithms.

Finally, we test our algorithms on a class of strongly DR-submodular functions and also for the problem of computing the stability number of graphs in Section 7.7. A summary of our results is provided in Table 7.1.

7.3 Preliminaries

7.3.1 Strongly DR-submodular functions

We define the class of strongly DR-submodular functions below.

Definition 7.3.1 *For $\mu \geq 0$, we call a differentiable function $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$, to be μ-strongly DR-submodular if any of the following equivalent properties hold:*

- *$f(\cdot) + \frac{\mu}{2}\|\cdot\|^2$ is DR-submodular.*
- *For all $x, y \in \mathcal{K}$ such that $x \preceq y$, we have*

$$\nabla f(x) \succeq \nabla f(y) + \mu(y - x).$$

- *If f is twice differentiable, $\nabla^2_{ii} f(x) \leq -\mu \ \forall i \in [n]$ and $\nabla^2_{ij} f(x) \leq 0 \ \forall i \neq j$ holds for all $x \in \mathcal{K}$.*

We provide two classes of strongly DR-submodular functions below:

- **Indefinite quadratic functions.** Let $f(x) = \frac{1}{2}x^T H x + h^T x + c$ where H is a symmetric matrix. If H is entry-wise non-positive, f is a DR-submodular function and if in addition, $H_{ii} \leq -\mu$ holds for all $i \in [n]$, f is μ-strongly DR-submodular. Such quadratic utility functions have a wide range of applications. As an example, consider the problem of computing the stability number of a graph (introduced in Section 7.2). The Hessian of the objective function $x^T(-A - I)x + 21^T x$ is $H = -2A - 2I$. Given that $H_{ii} = -2$ and $H_{ij} \leq 0$ for all $i \neq j \in [n]$, the objective function is 2-strongly DR-submodular. In addition, price optimization with continuous prices [109] is also a non-concave (strongly) DR-submodular quadratic optimization problem.

- **Concave functions with negative dependence.** Let $d \geq 2$. If $h_i : \mathbb{R}_+ \to \mathbb{R}$ is (strongly) concave for all $i \in [n]$ and $\theta_{i_1,\ldots,i_r} \leq 0$ for all $r \in [d]$ and $(i_1, \ldots, i_r) \subseteq [n]$, the following function $f : \mathbb{R}_+^n \to \mathbb{R}$ is (strongly) DR-submodular:

$$f(x) = \sum_{i=1}^{n} h_i(x_i) + \sum_{(i,j):i\neq j} \theta_{ij} x_i x_j + \cdots + \sum_{(i_1,\ldots,i_d):i_r \neq i_s \ \forall r,s\in[d]} \theta_{i_1,\ldots,i_d} x_{i_1} \ldots x_{i_d}.$$

We can prove that μ-strongly DR-submodular functions are indeed μ-strongly concave along non-negative directions. This property is extensively used in the design and analysis of our algorithms.

Lemma 7.3.1 *If $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}_+^n$, is differentiable and μ-strongly DR-submodular, for all $x \in \mathcal{K}$ and $v \succeq 0$ or $v \preceq 0$, the following holds:*

$$f(x+v) \leq f(x) + \langle \nabla f(x), v \rangle - \frac{\mu}{2}\|v\|^2.$$

Proof Without loss of generality, assume $v \succeq 0$ holds (analysis for $v \preceq 0$ is similar). For any $z \in \mathcal{K}$, $v \succeq 0$ and $i \in [n]$, we have:

$$\begin{aligned}
[\nabla^2 f(z)v]_i + \mu v_i &= \nabla^2_{ii} f(z) v_i + \sum_{j\neq i} \nabla^2_{ij} f(z) v_j + \mu v_i \\
&= \underbrace{(\nabla^2_{ii} f(z) + \mu)}_{\leq 0} \underbrace{v_i}_{\geq 0} + \sum_{j\neq i} \underbrace{\nabla^2_{ij} f(z)}_{\leq 0} \underbrace{v_j}_{\geq 0} \\
&\leq 0.
\end{aligned}$$

Therefore, $\nabla^2 f(z)v + \mu v \preceq 0$ holds. We can use the mean value theorem to write:

$$\begin{aligned}
&f(x+v) - f(x) - \langle \nabla f(x), v \rangle \\
&= \int_0^1 \langle \nabla f(x+tv), v \rangle dt - \langle \nabla f(x), v \rangle \\
&= \int_0^1 \langle \nabla f(x+tv) - \nabla f(x), v \rangle dt \\
&= \int_0^1 \sum_{i=1}^{n} t(\underbrace{[\nabla^2 f(z_{i,t})v]_i + \mu v_i}_{\leq 0}) \underbrace{v_i}_{\geq 0} \, dt \underbrace{- \mu \int_0^1 t\langle v, v \rangle dt}_{= \frac{-\mu}{2}\|v\|^2} \\
&\leq \frac{-\mu}{2}\|v\|^2,
\end{aligned}$$

where $z_{i,t}$ $\forall i \in [n]$ is in the line segment between x and $x + tv$. Thus, f is μ-strongly concave along the non-negative direction v. ■

[27] defined μ-strongly DR-submodular functions to be the class of functions that are μ-strongly concave along non-negative directions. Note that our definition of μ-strong DR-submodularity is a stronger condition than μ-strong concavity along non-negative directions. For instance, μ-strongly concave functions are μ-strongly concave along any direction, but they may not even be DR-submodular.

7.3.2 Smooth functions

A differentiable function $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$, is called L-smooth if for all $x, y \in \mathcal{K}$, we have

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle - \frac{L}{2}\|y - x\|^2.$$

If f is twice differentiable, there is an equivalent definition of smoothness: f is L-smooth if $\nabla^2 f(x) \succeq -LI$ holds for all $x \in \mathcal{K}$ where I is the identity matrix. In other words, the smallest eigenvalue of the Hessian of f is uniformly lower bounded by $-L$ everywhere. Combining the definition of smooth functions and the result of Lemma 7.3.1, it is clear that for a μ-strongly DR-submodular and L-smooth function, $\mu \leq L$ holds.

7.3.3 Curvature

We define the notion of curvature for monotone continuous functions below.

Definition 7.3.2 *Given a monotone differentiable function $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$, we define the curvature of f as follows:*

$$c_f = 1 - \inf_{x,y \in \mathcal{K}, i \in [n]} \frac{\nabla_i f(y)}{\nabla_i f(x)}.$$

If f is DR-submodular and $0 \in \mathcal{K}$, we have $c_f = 1 - \inf_{x \in \mathcal{K}, i \in [n]} \frac{\nabla_i f(x)}{\nabla_i f(0)}$.

It is easy to see that $c_f \in [0, 1]$ holds for all monotone f. $c_f \leq 1$ is due to monotonicity of f (i.e., ∇f being non-negative) and $c_f \geq 0$ follows from setting $x = y$ in the definition. If

$c_f = 0$, f is linear, and larger c_f corresponds to f being more curved.

A similar notion of curvature was introduced in [188]. The definition is inspired by the curvature of submodular set functions [68]. If f is the multilinear extension of a monotone submodular set function F, c_f coincides with the curvature of F [182]. Submodular set function maximization with bounded curvature has been widely studied in the literature. [68] showed that the greedy algorithm applied to the monotone submodular set function maximization problem subject to a cardinality constraint has a $\frac{1}{\kappa}(1-e^{-\kappa})$ approximation ratio, where κ is the curvature of the set function. More recently, [197] proposed two approximation algorithms for the more general problem of monotone submodular maximization subject to a matroid constraint and obtained a $1 - \frac{\kappa}{e}$ approximation ratio for these two algorithms. They also provided matching upper bounds for this problem showing that the $1 - \frac{\kappa}{e}$ approximation ratio is indeed optimal. Later on, [83] managed to obtain the same $1 - \frac{\kappa}{e}$ approximation ratio with an algorithm that is much faster than the one proposed by [197].

7.4 Strongly DR-submodular Frank-Wolfe (SDRFW) algorithm

In this section, we propose the SDRFW algorithm for strongly DR-submodular maximization with bounded curvature. Throughout the section, we make the further assumption that the domain set $\mathcal{K}$ contains the origin, i.e., $0 \in \mathcal{K}$. Furthermore, we consider f to be a monotone, μ-strongly DR-submodular and L-smooth function with curvature $c_f \in [0, 1]$. Without loss of generality, we also assume that f is normalized, i.e., $f(0) = 0$. For the DR-submodular setting ($\mu = 0$), [25] proposed a Frank-Wolfe variant for solving the problem. Starting from $x_0 = 0$, their algorithm performs K Frank-Wolfe updates where at each iteration $k \in \{0, \ldots, K-1\}$, it finds v_k such that $v_k = \arg\max_{x \in \mathcal{K}} \langle x, \nabla f(x_k) \rangle$, performs the update $x_{k+1} = x_k + \frac{1}{K} v_k$ and outputs x_K.

Define $g(x) = f(x) - \ell^T x$ where for all $i \in [n]$, $\ell_i = \min_x \nabla_i f(x)$. Note that similar to f, g is also a normalized, monotone, μ-strongly DR-submodular and L-smooth function.

The SDRFW algorithm is presented in Algorithm 1. First, note that the output of the algorithm ($x = x_K$) is the average of K points $\{v_k\}_{k=0}^{K-1}$ in the convex domain set $\mathcal{K}$, and therefore, $x \in \mathcal{K}$. Also, it is noteworthy that the update rule for $\{v_k\}_{k=1}^{K}$ can be computed efficiently in many cases. To see this, for all $k \in \{0, \ldots, K-1\}$, we can equivalently write:

$$v_k = \text{Proj}_{\mathcal{K}}\big(\frac{1}{\mu}\nabla g(x_k) + \frac{1}{\mu(1-\frac{1}{K})^{K-k-1}}\ell\big).$$

In many cases, such projection could be computed in linear time $\mathcal{O}(n)$ [33, 168], e.g., for $\mathcal{K} = \{x \in \mathbb{R}^n : 1^T x \leq 1, 0 \preceq x \preceq 1\}$.

Algorithm 1 is different from the Frank-Wolfe variant of [25] in two important aspects:

1) At step $k \in \{0, \ldots, K-1\}$, Algorithm 1 is applied to the function $(1-\frac{1}{K})^{K-k-1} g(\cdot) + \langle \ell, \cdot \rangle$ (instead of $f(\cdot) = g(\cdot) + \langle \ell, \cdot \rangle$),

2) The linear maximization step for computing $\{v_k\}_{k=0}^{K-1}$ in the Frank-Wolfe variant of [25] is replaced by a strongly concave maximization problem. Modification 1 is inspired by a similar idea in [83] where they provided an algorithm for the setting where the objective function is the multilinear extension of a submodular set function ($\mu = 0$) and obtained a $1 - \frac{c_f}{e}$ approximation ratio for the problem. They also proved matching negative results showing that no polynomial time algorithm can perform better in terms of the approximation ratio. The same upper bound applies to our framework as well. However, the additional strong DR-submodularity of f allows for a faster convergence to $(1-\frac{c_f}{e})$OPT. We provide the approximation guarantee of Algorithm 1 below.

Theorem 7.4.1 *Let $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$ and $0 \in \mathcal{K}$, be a normalized, monotone, μ-strongly DR-submodular and L-smooth function. If we set $K = \lceil \frac{L}{\mu} \rceil$, the output of Algorithm 1 has the following performance guarantee:*

$$f(x) \geq (1 - \frac{c_f}{e}) f(x^*),$$

where $x^* = \arg\max_{x \in \mathcal{K}} f(x)$.

Proof Define $g(x) = f(x) - \ell^T x$ where for all $i \in [n]$, $\ell_i = \min_x \nabla_i f(x)$. Note that similar to f, g is also a normalized, monotone, μ-strongly DR-submodular, and L-smooth function. For instance, in order to verify the μ-strong concavity of g along non-negative directions

Algorithm 1 SDRFW

```
Input: Convex set $\mathcal{K}$ with $0 \in \mathcal{K}$, $L$-smooth and $\mu$-strongly DR-submodular $f$, $g(\cdot) = f(\cdot) - \langle \ell, \cdot \rangle$ where for all $i \in [n]$, $\ell_i = \min_x \nabla_i f(x)$, $K > 0$.
Set $x_0 = 0$.
for $k = 0$ to $K - 1$ do
    Set $v_k = \arg\max_{x \in \mathcal{K}} \langle (1 - \frac{1}{K})^{K-k-1} \nabla g(x_k) + \ell, x \rangle - \frac{\mu (1 - \frac{1}{K})^{K-k-1}}{2} \|x\|^2$.
    $x_{k+1} = x_k + \frac{1}{K} v_k$.
end for
Output: $x = x_K$.
```

(which will be used in the proof), for all $x \in \mathcal{K}$ and $v \succeq 0$ or $v \preceq 0$, we can write:

$$\begin{aligned} g(x+v) &= f(x+v) - \langle \ell, x+v \rangle \\ &\leq f(x) + \langle \nabla f(x), v \rangle - \frac{\mu}{2}\|v\|^2 - \langle \ell, x+v \rangle \\ &= \underbrace{f(x) - \langle \ell, x \rangle}_{=g(x)} + \langle \underbrace{\nabla f(x) - \ell}_{=\nabla g(x)}, v \rangle - \frac{\mu}{2}\|v\|^2 \\ &= g(x) + \langle \nabla g(x), v \rangle - \frac{\mu}{2}\|v\|^2, \end{aligned}$$

where the inequality uses the μ-strong DR-submodularity of f. Therefore, g is μ-strongly concave along non-negative directions.

Now, we move on to prove the theorem. For all $k \in \{0, \dots, K-1\}$, we can write:

$$g(x_{k+1}) \geq g(x_k) + \frac{1}{K}\langle \nabla g(x_k), v_k \rangle - \frac{L}{2K^2}\|v_k\|^2.$$

Rearranging the terms, we have:

$$g(x_{k+1}) - g(x_k) \geq \frac{1}{K}\langle \nabla g(x_k), v_k \rangle - \frac{L}{2K^2}\|v_k\|^2. \tag{7.1}$$

For all $k \in \{0, \dots, K\}$, define the function $\phi(k)$ as follows:

$$\phi(k) = (1 - \frac{1}{K})^{K-k} g(x_k) + \langle \ell, x_k \rangle.$$

For a fixed $k \in \{0, \dots, K-1\}$, we have:

$$\begin{aligned}
&K\big(\phi(k+1) - \phi(k)\big) = K(1-\frac{1}{K})^{K-k-1} g(x_{k+1}) - K(1-\frac{1}{K})^{K-k} g(x_k) + \langle \ell, v_k \rangle \\
&= \frac{(1-\frac{1}{K})^{K-k-1}\big(g(x_{k+1}) - g(x_k)\big)}{1/K} + (1-\frac{1}{K})^{K-k-1} g(x_k) + \langle \ell, v_k \rangle \\
&\overset{(a)}{\geq} \frac{(1-\frac{1}{K})^{K-k-1}\big(1/K\langle \nabla g(x_k), v_k \rangle - L\|v_k\|^2/2K^2\big)}{1/K} + (1-\frac{1}{K})^{K-k-1} g(x_k) + \langle \ell, v_k \rangle \\
&= (1-\frac{1}{K})^{K-k-1}\big(\langle \nabla g(x_k), v_k \rangle - \frac{L}{2K}\|v_k\|^2\big) + (1-\frac{1}{K})^{K-k-1} g(x_k) + \langle \ell, v_k \rangle \\
&= \langle (1-\frac{1}{K})^{K-k-1} \nabla g(x_k) + \ell, v_k \rangle - (1-\frac{1}{K})^{K-k-1}\frac{L}{2K}\|v_k\|^2 + (1-\frac{1}{K})^{K-k-1} g(x_k) \\
&\overset{(b)}{\geq} (1-\frac{1}{K})^{K-k-1}\langle \nabla g(x_k), x^* \rangle + \langle \ell, x^* \rangle + \frac{\mu(1-\frac{1}{K})^{K-k-1}}{2}\|v_k\|^2 - \frac{\mu(1-\frac{1}{K})^{K-k-1}}{2}\|x^*\|^2 \\
&- (1-\frac{1}{K})^{K-k-1}\frac{L}{2K}\|v_k\|^2 + (1-\frac{1}{K})^{K-k-1} g(x_k),
\end{aligned}$$

where (a) follows from inequality 7.1 and (b) is due to the update rule of the SDRFW algorithm for v_k. We can use the monotonicity and strong DR-submodularity of g respectively to write:

$$\begin{aligned}
g(x^*) - g(x_k) &\leq g(x^* + x_k) - g(x_k) \\
&\leq \langle \nabla g(x_k), x^* \rangle - \frac{\mu}{2}\|x^*\|^2.
\end{aligned}$$

Putting the above two inequalities together, we have:

$$\begin{aligned}
K\big(\phi(k+1) - \phi(k)\big) &\geq (1-\frac{1}{K})^{K-k-1}\big(g(x^*) - g(x_k) + \frac{\mu}{2}\|x^*\|^2\big) + \langle \ell, x^* \rangle + (1-\frac{1}{K})^{K-k-1} g(x_k) \\
&+ \frac{\mu(1-\frac{1}{K})^{K-k-1}}{2}\|v_k\|^2 - \frac{\mu(1-\frac{1}{K})^{K-k-1}}{2}\|x^*\|^2 - (1-\frac{1}{K})^{K-k-1}\frac{L}{2K}\|v_k\|^2 \\
&= (1-\frac{1}{K})^{K-k-1} g(x^*) + \langle \ell, x^* \rangle + (1-\frac{1}{K})^{K-k-1}\big(\frac{\mu}{2} - \frac{L}{2K}\big)\|v_k\|^2.
\end{aligned}$$

Therefore, if we set $K = \lceil \frac{L}{\mu} \rceil$ and divide both sides by K, we obtain:

$$\phi(k+1) - \phi(k) \geq \frac{1}{K}(1-\frac{1}{K})^{K-k-1} g(x^*) + \frac{1}{K}\langle \ell, x^* \rangle.$$

Applying the inequality for all $k \in \{0, \dots, K-1\}$ and taking the sum, we have:

$$\begin{aligned} f(x_K) &= g(x_K) + \langle \ell, x_K \rangle \\ &= \phi(K) - \phi(0) \\ &\geq \Big(\frac{1}{K}\sum_{k=0}^{K-1}(1-\frac{1}{K})^{K-k-1}\Big)g(x^*) + \langle \ell, x^* \rangle \\ &= \Big(\frac{1}{K}\frac{1-(1-\frac{1}{K})^K}{1/K}\Big)g(x^*) + \langle \ell, x^* \rangle \\ &\geq (1-\frac{1}{e})g(x^*) + \langle \ell, x^* \rangle, \end{aligned}$$

where the last inequality uses $(1-\frac{1}{K})^K \leq \frac{1}{e}$. Therefore, $f(x) \geq (1-\frac{1}{e})g(x^*) + \langle \ell, x^* \rangle$ holds. Using the mean value theorem and considering that $f(0) = 0$, we have $f(x^*) = \sum_{i=1}^{n} \nabla_i f(tx^*)x_i^*$ where $t \in [0,1]$. Therefore, we can use the definition of ℓ and the curvature to write:

$$\ell^T x^* = \sum_{i=1}^{n} \ell_i x_i^* \geq (1-c_f)\sum_{i=1}^{n} \nabla_i f(tx^*)x_i^* = (1-c_f)f(x^*).$$

Putting the above inequalities together, we have:

$$\begin{aligned} f(x) &\geq (1-\frac{1}{e})g(x^*) + \ell^T x^* \\ &= (1-\frac{1}{e})f(x^*) - (1-\frac{1}{e})\ell^T x^* + \ell^T x^* \\ &= (1-\frac{1}{e})f(x^*) + \frac{1}{e}\ell^T x^* \\ &\geq (1-\frac{1}{e})f(x^*) + \frac{1-c_f}{e}f(x^*) \\ &= (1-\frac{c_f}{e})f(x^*). \end{aligned}$$

■

In comparison, the Frank-Wolfe variant of [25] obtains the approximation guarantee $f(x) \geq (1-\frac{1}{e})f(x^*) - \frac{LR^2}{2K}$ where $R = \max_{x,y\in\mathcal{K}} \|y - x\|$ is the diameter of $\mathcal{K}$. Therefore, Algorithm 1 has an improved approximation ratio for objective functions f with curvature $c_f < 1$ and its guarantee does not deteriorate as the diameter of $\mathcal{K}$ becomes larger. Intuitively, for K chosen large enough (i.e., $K \geq \frac{L}{\mu}$) in the analysis of Algorithm 1, the $-\frac{LR^2}{2K}$ term is canceled

Algorithm 2 PGA

```
**Input**: Convex set $\mathcal{K}$, $x_1 \in \mathcal{K}$, $L$-smooth and $\mu$-strongly DR-submodular $f$, $K > 0$.
for $k = 1$ to $K$ do
    Set $x_{k+1} = \text{Proj}_{\mathcal{K}}\big(x_k + \frac{1}{L}\nabla f(x_k)\big)$.
end for
Output: $x = x_{K+1}$.
```

with a positive term resulting from μ-strong DR-submodularity of the objective function. We need to know the smoothness parameter L and the strong DR-submodularity parameter μ to set K in Algorithm 1. For μ-strongly DR-submodular functions, if we run Algorithm 1 with $\hat{\mu}$ instead of μ, the algorithm obtains the same guarantee as long as $\hat{\mu} \leq \mu$. In order to compute such a lower bound, one can investigate the diagonal entries of the Hessian matrix. In Section 7.6, we show how to compute L efficiently using convex optimization tools.

7.5 Projected Gradient Ascent (PGA) algorithm

In this section, we study the well-known Projected Gradient Ascent (PGA) method for strongly DR-submodular maximization with bounded curvature. The PGA algorithm is provided in Algorithm 2 [162]. Given an initial point $x_1 \in \mathcal{K}$, PGA iteratively applies the update $x_{k+1} = \text{Proj}_{\mathcal{K}}\big(x_k + \frac{1}{L}\nabla f(x_k)\big)$. In other words, at each iteration $k \in [K]$, the current iterate x_k is updated by adding $\frac{1}{L}\nabla f(x_k)$ and the resulting point is then projected back to the constraint set $\mathcal{K}$. The algorithm outputs the final iterate $x = x_{K+1}$. Unlike the SDRFW algorithm, PGA is not required to start from the origin and for any feasible initial point $x_1 \in \mathcal{K}$, PGA still converges to a competitive solution. However, as we will soon see in the result of Theorem 7.5.1, the rate of convergence depends on the distance between the initial point x_1 and the optimal point x^*.

We first provide a key lemma below that is used in the analysis of Algorithm 2.

Lemma 7.5.1 *For any $x, z \in \mathcal{K}$, if f is a non-negative monotone μ-strongly DR-submodular* *function with curvature c_f, we have:*

$$f(z) - (1 + c_f)f(x) \leq \langle \nabla f(x), z - x \rangle - \frac{\mu}{2}\|z - x\|^2.$$

Proof Let $u = x \vee z$ and $w = x \wedge z$. Using the μ-strong DR-submodularity property of f, we can write:

$$\begin{aligned} f(u) - f(x) &\leq \langle \nabla f(x), u - x \rangle - \frac{\mu}{2}\|u - x\|^2, \\ f(w) - f(x) &\leq \langle \nabla f(x), w - x \rangle - \frac{\mu}{2}\|w - x\|^2. \end{aligned}$$

Taking the sum of the two inequalities and using the fact that $u + w = x + z$, we have:

$$f(u) + f(w) - 2f(x) \leq \langle \nabla f(x), z - x \rangle - \frac{\mu}{2}\|z - x\|^2. \tag{7.2}$$

Using the mean value theorem, we can write:

$$\begin{aligned} f(u) - f(z) &= \int_0^1 \langle u - z, \nabla f\big(z + t(u - z)\big) \rangle dt, \\ f(w) - f(x) &= \int_0^1 \langle w - x, \nabla f\big(w + t(x - w)\big) \rangle dt. \end{aligned}$$

Given that $x - w = u - z$ and $\nabla_i f\big(z + t(u - z)\big) \geq (1 - c_f)\nabla_i f\big(w + t(x - w)\big)$ holds for all $i \in [n]$, we can bound the first inequality as follows:

$$\begin{aligned} f(u) - f(z) &\geq -(1 - c_f) \int_0^1 \langle w - x, \nabla f\big(w + t(x - w)\big) \rangle dt \\ &= -(1 - c_f)\big(f(w) - f(x)\big). \end{aligned}$$

Equivalently, we can write:

$$f(u) + f(w) \geq f(z) + (1 - c_f)f(x) + c_f f(w). \tag{7.3}$$

Combining the inequalities 7.2 and 7.3, we conclude:

$$f(z) - (1 + c_f)f(x) + c_f f(w) \leq \langle \nabla f(x), z - x \rangle - \frac{\mu}{2}\|z - x\|^2.$$

Given that $c_f \in [0, 1]$ and f is non-negative, we can drop the term $c_f f(w)$ and derive the result as stated. ■

We can now exploit the result of Lemma 7.5.1 to obtain the approximation guarantee of the PGA algorithm.

Theorem 7.5.1 *If f is a monotone, μ-strongly DR-submodular and L-smooth function, PGA obtains the following approximation guarantee:*

$$f(x) \geq \frac{1}{1+c_f} f(x^*) - \frac{e^{-\mu K/L}}{1+c_f}\big(f(x^*) - (1+c_f)f(x_1)\big).$$

Moreover, for the DR-submodular setting ($\mu = 0$), the utility of the output of the PGA algorithm is bounded as follows:

$$f(x) \geq \frac{1}{1+c_f} f(x^*) - \frac{L}{2K(1+c_f)} \|x_1 - x^*\|^2.$$

[102] analyzed the PGA method in the DR-submodular setting ($\mu = 0$) and proved that $f(x) \geq \frac{1}{2} f(x^*) - \frac{LR^2}{2K}$. In comparison, thanks to Lemma 7.5.1, we obtain an improved $\frac{1}{1+c_f}$ approximation ratio with a similar sub-linear convergence rate in the DR-submodular setting. Moreover, if f is μ-strongly DR-submodular, Theorem 7.5.1 shows that the $\frac{1}{1+c_f}$ approximation ratio could be achieved at a faster linear convergence rate. Furthermore, if $c_f < \frac{1}{e-1} \approx 0.58$, the $\frac{1}{1+c_f}$ approximation ratio obtained in Theorem 7.5.1 is larger than the $1 - \frac{1}{e}$ approximation ratio guaranteed by the Frank-Wolfe variant of [25]. However, $\frac{1}{1+c_f} \leq 1 - \frac{c_f}{e}$ always holds, i.e., the approximation ratio of the SDRFW algorithm is greater than that of the PGA algorithm.

7.5.1 Online setting

We can also use Lemma 7.5.1 to obtain improved regret bounds in the online setting for the Online Gradient Ascent (OGA) algorithm. To be precise, consider the following online optimization protocol: A convex constraint set $\mathcal{K}$ with diameter R is given. At each iteration $t \in [T]$, the online algorithm first chooses an action $x_t \in \mathcal{K}$. Upon committing to this choice, a monotone (strongly) DR-submodular function $f_t : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$, is revealed and the algorithm receives the reward $f_t(x_t)$. The goal is to maximize the total obtained reward or equivalently minimize the α-regret α-$R_T = \alpha \max_{x \in \mathcal{K}} \sum_{t=1}^T f_t(x) - \sum_{t=1}^T f_t(x_t)$, i.e., the

Algorithm 3 OGA

```
Input: Convex set $\mathcal{K}$, $x_1 \in \mathcal{K}$, $L$-smooth and $\mu$-strongly DR-submodular functions $\{f_t\}_{t=1}^T$, step sizes $\{\eta_t\}_{t=1}^T$.
Output: $\{x_t\}_{t=1}^T$.
for $t = 1$ to $T$ do
    Play $x_t$ and receive the reward $f_t(x_t)$.
    Set $x_{t+1} = \mathrm{Proj}_{\mathcal{K}}\big(x_t + \eta_t \nabla f_t(x_t)\big)$.
end for
```

difference between the cumulative reward of the algorithm and the α approximation to that of the best fixed decision in hindsight where $\alpha \in (0, 1]$.

The Online Gradient Ascent (OGA) algorithm is provided in Algorithm 3. OGA is the online counterpart of the PGA algorithm for the offline setting. Starting from an arbitrary initial point $x_1 \in \mathcal{K}$, for all $t \in [T-1]$, OGA uses the update $x_{t+1} = \mathrm{Proj}_{\mathcal{K}}\big(x_t + \eta_t \nabla f_t(x_t)\big)$ to obtain the next iterate x_{t+1}, where $\eta_t > 0$ is the step size. [61] analyzed the OGA algorithm in the DR-submodular setting ($\mu = 0$) and provided $\mathcal{O}(\sqrt{T})$ bounds for the $\frac{1}{2}$-regret of the algorithm. Using Lemma 7.5.1, we can obtain improved $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(\ln T)$ ($\frac{1}{1+c}$)-regret bounds in the DR-submodular and strongly DR-submodular settings respectively where $c = \max_{t \in [T]} c_{f_t}$. This result is stated in the theorem below.

Theorem 7.5.2 *Assume that the functions $\{f_t\}_{t=1}^T$ are all monotone, β-Lipschitz and μ-strongly DR-submodular. If for all $t \in [T]$, we set $\eta_t = \frac{1}{\mu t}$, the OGA algorithm has the following ($\frac{1}{1+c}$)-regret bound.*

$$\frac{1}{1+c} \sum_{t=1}^{T} f_t(x^*) - \sum_{t=1}^{T} f_t(x_t) \leq \frac{\beta^2}{2\mu(1+c)}(1 + \ln T),$$

where $x^* = \arg\max_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)$. Moreover, in the DR-submodular setting ($\mu = 0$), if we set $\eta_t = \eta = \frac{R}{\beta\sqrt{T}}$ $\forall t \in [T]$, the ($\frac{1}{1+c}$)-regret of the algorithm is bounded as follows:

$$\frac{1}{1+c} \sum_{t=1}^{T} f_t(x^*) - \sum_{t=1}^{T} f_t(x_t) \leq \frac{R\beta}{1+c}\sqrt{T}.$$

For the online monotone DR-submodular maximization problem with bounded curvature, the only prior study was done by [101] where the authors proposed an algorithm for the special setting where the DR-submodular functions $\{f_t\}_{t=1}^T$ are the multilinear extensions of corresponding submodular set functions $\{F_t\}_{t=1}^T$ and they showed that the algorithm obtains an $\mathcal{O}(\sqrt{T})$ $(1 - \frac{c}{e} - \epsilon)$-regret bound with $\frac{n^2}{\epsilon}$ projections per iteration. In comparison, while the approximation ratio in Theorem 7.5.2 is slightly worse, Algorithm 3 performs *only a single projection per step* (hence a significantly lower computational complexity) and its logarithmic regret bound is superior in the strongly DR-submodular setting.

7.6 Computing the smoothness parameter

Both the SDRFW and PGA algorithms require knowledge of the smoothness parameter L of the objective function to be implemented. In this section, we show how computing L of a twice differentiable μ-strongly DR-submodular objective function f $(\mu \geq 0)$ could be formulated as a convex optimization problem that could be solved efficiently before running the algorithms. Given that our technique applies to the DR-submodular setting (i.e., $\mu = 0$) as well, the results of this section are useful for the proposed algorithms in prior works for DR-submodular maximization. In particular, while [102] chose the step size of (stochastic) PGA as a function of the smoothness parameter L to obtain the theoretical approximation guarantees in their work, they mentioned that estimating L is difficult in general and poses a challenge for implementation. Therefore, they suggested an alternative adaptive step size rule (as a function of the iteration number) with no theoretical performance guarantees in their experiments. This section precisely addresses the aforementioned challenge.

As we defined smoothness earlier in Section 7.3, we need to find a constant $L > 0$ such that for all $x \in \mathcal{K}$, the smallest eigenvalue of $\nabla^2 f(x)$ is lower bounded by $-L$, i.e., $\nabla^2 f(x) \succeq -LI$. This is equivalent to finding $L > 0$ such that the largest eigenvalue of $-\nabla^2 f$ is uniformly upper bounded by L everywhere. Given that f is μ-strongly DR-submodular, $-\nabla^2 f$ is an element-wise non-negative matrix. The Perron-Frobenius theorem [107, Theorem 8.4.4] states that if $-\nabla^2 f$ is irreducible, i.e., the matrix $(I - \nabla^2 f)^{n-1}$ is element-wise positive,

$-\nabla^2 f$ has a positive real eigenvalue λ_{pf} equal to its spectral radius (which is the largest magnitude of its eigenvalues) and an entry-wise positive eigenvector $v_{pf} \succ 0$ corresponding to λ_{pf} and therefore, we can set $L = \lambda_{pf}$. In order to check irreducibility of the symmetric matrix $-\nabla^2 f$, we can associate with it an undirected graph G with n vertices labeled $\{1, \ldots, n\}$ where there is an edge between vertices i and j if $-\nabla^2_{ij} f = -\nabla^2_{ji} f \gneq 0$. $-\nabla^2 f$ is irreducible if and only if its associated graph G is connected. According to a result in the theory of non-negative matrices, the Perron-Frobenius eigenvalue is the solution of the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \sum_{j=1}^{n} -\nabla^2_{ij} f(x) v_j \leq \lambda v_i \ \forall i \in [n], \end{array} \tag{7.4}$$

where the variables are $x \succ 0$, $v \succ 0$ and $\lambda > 0$. We show how the above problem could be transformed into a convex optimization problem in many cases.

A function $h : \mathbb{R}^n_{++} \to \mathbb{R}$ defined as $h(x) = c x_1^{a_1} \ldots x_n^{a_n}$, where $c > 0$ and $a_i \in \mathbb{R}\ \forall i \in [n]$, is called a monomial. A sum of m monomials, i.e., a function of the form $h(x) = \sum_{s=1}^{m} c_s x_1^{a_{1s}} \ldots x_n^{a_{ns}}$ where $c_s > 0\ \forall s \in [m]$ and $a_{is} \in \mathbb{R}\ \forall i \in [n], s \in [m]$, is called a posynomial. Consider the following optimization problem with variable $x \in \mathbb{R}^n_{++}$:

$$\begin{array}{ll} \text{minimize} & h_0(x) \\ \text{subject to} & h_{1i}(x) \leq 1 \ \forall i \in [n] \\ & h_{2j}(x) = 1 \ \forall j \in [p]. \end{array}$$

If $h_0, h_{11}, h_{12}, \ldots, h_{1n}$ are posynomials and $h_{21}, \ldots, h_{2p}$ are monomials, the above problem is called a Geometric Program (GP). While GPs are not generally convex optimization problems, they can be transformed into convex problems (see [31, Section 4.5.3] for details). For all $i \in [n]$, we can rewrite the constraints of problem (7.4) in the following equivalent way:

$$(\lambda^{-1} v_i^{-1}) \sum_{j=1}^{n} -\nabla^2_{ij} f(x) v_j \leq 1.$$

If the non-zero entries of $-\nabla^2 f(x)$ are posynomial functions of the variable x, the constraints of problem (7.4) are posynomial inequalities and therefore, the optimization problem for computing the smoothness parameter $L = \lambda_{pf}$ can be expressed as a GP.

As an example, consider the problem of computing the stability number of an undirected graph G with adjacency matrix A where $f(x) = x^T(-A - I)x + 21^T x$ (introduced in Section 7.2). Without loss of generality, assume that G is connected (otherwise, we can consider each connected component of G separately and use the fact that the stability number of a graph equals the sum of the stability numbers of its connected components). Since G is connected, $-\nabla^2 f(x) = 2A + 2I$ is an entry-wise non-negative and irreducible matrix, and therefore, computing the smoothness parameter L of f could be formulated as a GP.

More generally, consider the class of concave functions with negative dependence. Let $d \geq 2$. If $h_i : \mathbb{R}_+ \to \mathbb{R}$ is μ-strongly concave for all $i \in [n]$ and $\theta_{i_1,\ldots,i_r} \leq 0$ for all $r \in [d]$ and $(i_1, \ldots, i_r) \subseteq [n]$, the following function $f : \mathbb{R}_+^n \to \mathbb{R}$ is μ-strongly DR-submodular:

$$f(x) = \sum_{i=1}^{n} h_i(x_i) + \sum_{(i,j):i\neq j} \theta_{ij} x_i x_j + \cdots + \sum_{(i_1,\ldots,i_d):i_r \neq i_s \ \forall r,s\in[d]} \theta_{i_1,\ldots,i_d} x_{i_1} \ldots x_{i_d}.$$

It is easy to see that all off-diagonal entries of $-\nabla^2 f(x)$ are posynomials. If for all $i \in [n]$, $-h_i''(x_i)$ is a posynomial as well, the smoothness parameter of f could be computed as the solution of a GP.

7.7 Numerical examples

For the first experiment, we set $n = 25$ and chose $\mathcal{K} = \{x \in \mathbb{R}^n : 1^T x \leq s, 0 \preceq x \preceq 1\}$. We considered the class of indefinite quadratic functions and defined $f(x) = (\frac{1}{2}x - \mathbf{1})^T Hx$, where $H \in \mathbb{R}^{n\times n}$ is a matrix whose entries are uniformly distributed in the range $[-10, -5]$. Therefore, $\mu = 5$ in this setting. We computed L using the technique described in Section 7.6 and set $K = \lceil \frac{L}{\mu} \rceil$. We ran Algorithm 1, Frank-Wolfe variant of [25] and PGA for K iterations using different values of s in the range $[2, 20]$ (using $x_1 = 0$ for PGA) and plotted the utility of the output of all three algorithms in each setting. The plot is depicted in Figure 7.1. This plot shows that Algorithm 1 obtains slightly higher utilities, followed by PGA and the Frank-Wolfe variant of [25].

In the second experiment, we studied the problem of computing the stability number $s(G)$

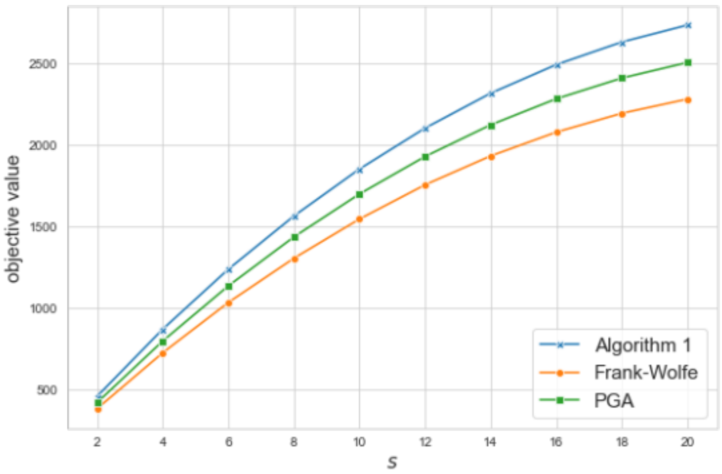


Figure 7.1: Comparison of algorithms for a class of strongly DR-submodular quadratic functions.

for two graphs. The first graph is provided in Figure 7.2. It is easy to see that $s(G) = 4$ for this graph (e.g., vertices $\{3, 5, 6, 8\}$ form a maximum stable set of G). As it was mentioned earlier in Section 7.2, we set $\mathcal{K} = \{x \in \mathbb{R}^n : 1^T x = 1, x_i \geq 0 \ \forall i \in [n]\}$ where $n = 10$. Also, we defined $f(x) = x^T(-A - I)x + 21^T x$ and using the formula $s(G)^{-1} = 2 - \max_{x \in \mathcal{K}} f(x)$, we set $\frac{1}{2-f(x)}$ as the estimate of the stability number at $x \in \mathcal{K}$. Since all the diagonal entries of $\nabla^2 f(\cdot)$ are equal to -2, we have $\mu = 2$. We also computed L using the method presented in Section 7.6. We ran PGA for this problem and plotted $\frac{1}{2-f(x_k)}$ versus the iteration number k in Figure 7.3(a). As the plot shows, PGA converges to the optimal value 4 after only

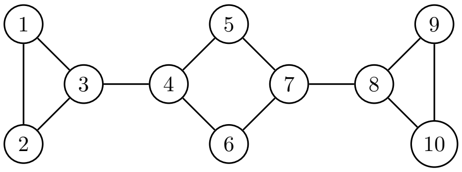
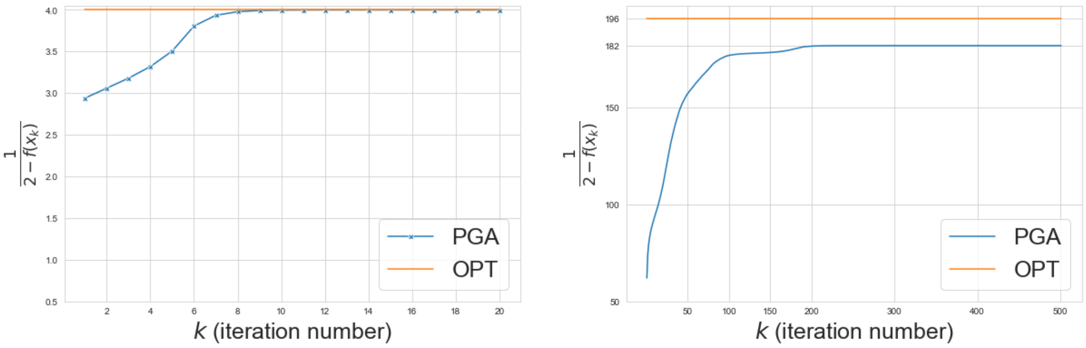


Figure 7.2: Graph G with $n = 10$ used in Experiment 2.



(a)

(b)

Figure 7.3: Computing the stability number $s(G)$ for a graph G with (a) $n = 10$ vertices and $s(G) = 4$, (b) $n = 1024$ vertices and $s(G) = 196$.

10 iterations. As the second example, we considered a graph with $n = 1024$ vertices from `https://oeis.org/A265032/a265032.html` that contains a collection of graph instances commonly used in coding theory. Using an algorithm with a running time of 200 hours, [164] managed to show that the stability number of this graph is 196. The performance of PGA for this problem is plotted in Figure 7.3(b). The algorithm converges to the value 182 as the estimate of the stability number and therefore, the performance of PGA for this problem is significantly better than the approximation guarantee proved in Theorem 7.5.1. Note that the domain in the second experiment does not contain the origin and thus, Algorithm 1 does not apply to this problem.

7.8 Conclusion and future work

In this chapter, we considered the class of monotone, smooth, and strongly DR-submodular functions with bounded curvature and we proposed many first-order gradient methods for this problem along with their approximation guarantees and convergence rates.

This work could be extended in several directions. Throughout this chapter, we assumed that we have access to the exact gradient of the objective function. However, in many applications, it is difficult to compute the gradient exactly, but an unbiased estimate of the gradient can be easily obtained. It is interesting to study stochastic gradient methods for this setting and provide performance guarantees. Secondly, we introduced the class of strongly DR-submodular functions with respect to the norm $\|\cdot\|_2$. As we showed in Lemma 7.3.1, strongly DR-submodular functions are strongly concave along non-negative directions with respect to $\|\cdot\|_2$. This definition could be easily extended to other norms as well. Similarly, while the smoothness of f over the domain $\mathcal{K}$ was defined with respect to the Euclidean norm, in some cases, f and $\mathcal{K}$ are not well-behaved in the $\|\cdot\|_2$ norm and L scales with the ambient dimension n leading to slower convergence rates for our proposed algorithms in large-scale applications. In such cases, one can study mirror ascent methods that are designed to adapt to smoothness in general norms.

Chapter 8

ONLINE STRONGLY DR-SUBMODULAR MAXIMIZATION

8.1 Chapter overview

This chapter is based on joint work with Maryam Fazel. A shorter version of this work appeared at the International Conference on Machine Learning (ICML) 2021 Workshop on Subset Selection in Machine Learning: From Theory to Applications. In this chapter, we consider an online optimization problem over T rounds where at each step $t \in [T]$, the algorithm chooses an action x_t from the fixed convex and compact domain set $\mathcal{K}$. A continuous utility function $f_t(\cdot)$ is then revealed and the algorithm receives the payoff $f_t(x_t)$. This problem has been previously studied under the assumption that the utilities are adversarially chosen monotone DR-submodular functions and $\mathcal{O}(\sqrt{T})$ regret bounds have been derived. We first characterize the class of strongly DR-submodular functions and then, we derive regret bounds for the following new online settings: (1) $\{f_t\}_{t=1}^T$ are monotone strongly DR-submodular and chosen adversarially, (2) $\{f_t\}_{t=1}^T$ are monotone DR-submodular (while the average $\frac{1}{T}\sum_{t=1}^T f_t$ is strongly DR-submodular) and chosen by an adversary but they arrive in a uniformly random order, (3) $\{f_t\}_{t=1}^T$ are drawn i.i.d. from some unknown distribution $f_t \sim \mathcal{D}$ where the expected function $f(\cdot) = \mathbb{E}_{f_t \sim \mathcal{D}}[f_t(\cdot)]$ is monotone DR-submodular. For (1), we obtain the first logarithmic regret bounds. In terms of the second framework, we show that it is possible to obtain similar logarithmic bounds with high probability. Finally, for the i.i.d. model, we provide algorithms with $\tilde{\mathcal{O}}(\sqrt{T})$ stochastic regret bound, both in expectation and with high probability. Experimental results demonstrate that our algorithms outperform the previous techniques in the aforementioned three settings.

8.2 Introduction

Online Convex Optimization (OCO) is a well-studied framework for sequential prediction in the face of uncertainty inherent in the arriving data [218, 104, 189, 35]. OCO can be interpreted as a repeated game between a learner and an adversary in which at each round $t \in [T]$, the player chooses an action x_t from a fixed convex and compact domain $\mathcal{K}$ and upon committing to this action, a convex loss function $f_t : \mathcal{K} \to \mathbb{R}$ is revealed and the learner incurs a loss of $f_t(x_t)$. In the non-stochastic adversary model, the sequence of loss functions $\{f_t\}_{t=1}^T$ is assumed to be selected by an adversary who knows the learner's algorithm (but not the potential randomness used for prediction). The goal is to choose $\{x_t\}_{t=1}^T$ such that the *regret* of the learner defined as $R_T = \sum_{t=1}^T f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^T f_t(x)$ is minimized. For this setting, several algorithms have been proposed to obtain the provably optimal $\mathcal{O}(\sqrt{T})$ regret bound [167, 104, 189, 35]. Similarly, for the case where $\{f_t\}_{t=1}^T$ is a sequence of monotone DR-submodular utility functions, [195], [95] and [61] obtained $\mathcal{O}(\sqrt{T})$ regret bounds against the $(1 - \frac{1}{e})$-approximation to the best fixed decision in hindsight, i.e., regret being defined as $R_T = (1 - \frac{1}{e}) \max_{x \in \mathcal{K}} \sum_{t=1}^T f_t(x) - \sum_{t=1}^T f_t(x_t)$. If the sequence of loss functions is strongly convex, algorithms with logarithmic regret bounds have been introduced [103]. However, even though strongly DR-submodular functions were introduced by [183], the study of online strongly DR-submodular maximization problems is missing in the literature.

In some scenarios, despite the adversarial nature of the sequence of arriving functions, the online input does not have a temporal structure, and online data is streamed without particular order [82, 70, 7]. For instance, consider the problem of bidding in repeated auctions for online advertising in which at each round $t \in [T]$, an impression (e.g., online viewer) arrives and becomes available for sale through an auction. The advertiser needs to bid to win the auction and the corresponding impression. In this example, the viewers do not arrive in a particular order and therefore, their order of arrival could be assumed to be random.

Moreover, in many applications, the sequence of arriving functions $\{f_t\}_{t=1}^T$ is not chosen arbitrarily (adversarially). For instance, in empirical risk minimization, the learner wants to minimize a loss function f which is the expectation of empirical loss functions

$f_t(\cdot) = \tilde{f}(\cdot;\omega_t)\ \forall t \in [T]$, where ω_t is drawn i.i.d. from a fixed unknown distribution $\mathcal{D}$, i.e., $f(\cdot) = \mathbb{E}_{\omega\sim\mathcal{D}}[\tilde{f}(\cdot;\omega)]$. In this setting, we aim to minimize the *stochastic regret* defined as $SR_T = \sum_{t=1}^{T} f(x_t) - T\min_{x\in\mathcal{K}} f(x)$ $(T(1-\frac{1}{e})\max_{x\in\mathcal{K}} f(x) - \sum_{t=1}^{T} f(x_t)$ for the DR-submodular setting).

In this chapter, we fill the gaps in the online DR-submodular maximization literature by studying online strongly DR-submodular maximization problems in the adversarial setting, and also online DR-submodular problems against the aforementioned two stochastic adversary models.

8.3 Preliminaries

For a DR-submodular function f, f is L-smooth over non-negative directions with respect to $\|\cdot\|$ if for all x, y such that $x \preceq y$, we have $f(y) - f(x) \geq \langle \nabla f(x), y - x\rangle - \frac{L}{2}\|y-x\|^2$.

Example 8.3.1 *Let* $f(x) = \frac{1}{2}x^T Ax + a^T x + c$ *where* A *is a symmetric matrix. If for all* $i, j \in [n]$, $A_{ij} \geq -L$ *holds, the function* f *is* L*-smooth with respect to* $\|\cdot\|_1$.

8.3.1 Related work

Online optimization in the i.i.d. model. For the setting where the objective functions $\{f_t\}_{t=1}^{T}$ are drawn i.i.d. from an unknown distribution $\mathcal{D}$ with a convex mean $f(\cdot) = \mathbb{E}_{\mathcal{D}}[f_t(\cdot)]$, several algorithms have been proposed that could be divided into two categories of projection-based and projection-free algorithms where the latter is most relevant to our work. In particular, [105] proposed the Online Frank-Wolfe (OFW) algorithm which achieves a nearly optimal $\tilde{\mathcal{O}}(\sqrt{T})$ stochastic regret bound with high probability. The OFW algorithm requires to access exact gradient of $\{f_t\}_{t=1}^{T}$ and has an average $\mathcal{O}(T)$ per-iteration computational cost. To remedy this issue, more recently, [60] proposed the One-Shot Frank-Wolfe (OSFW) algorithm with $\mathcal{O}(T^{2/3})$ stochastic regret bound in expectation. Note that the OSFW algorithm obtains similar bounds for the setting where f is monotone continuous DR-submodular. The OSFW algorithm only uses unbiased stochastic gradient estimates of loss functions and has an $\mathcal{O}(1)$ cost per iteration. However, the derived $\mathcal{O}(T^{2/3})$ stochastic regret bound is sub-optimal. To bridge this gap, [208] introduced the Online stochastic Recursive Gradient-based Frank-Wolfe (ORGFW) algorithm which not only has a nearly optimal $\tilde{O}(\sqrt{T})$ stochastic regret bound with high probability, but it also maintains a low $\mathcal{O}(1)$ computational cost per round.

Online convex optimization in the random order model. [92] studied OCO in a weaker model where the loss functions are still chosen adversarially but their order of arrival is random. This model is termed Random Order Online Convex Optimization (ROOCO). ROOCO is a natural middle ground between the standard adversarial OCO setting and the model where the loss functions are drawn i.i.d. from some unknown underlying distribution. In [92], the authors assumed that the average of the sequence of loss functions is α-strongly convex while each individual loss function may not even be convex. They showed that if all the loss functions are quadratic, with probability at least $1-\delta$, Online Gradient Descent (OGD) with step size $\eta_t = \frac{1}{\alpha t}$ $\forall t \in [T]$ obtains a regret bound of $\mathcal{O}(\frac{\beta^2}{\alpha^3}\log^2 T)$, where β is the Lipschitz constant of the loss functions. Moreover, for general loss functions, OGD with the same choice of step sizes obtains a regret bound of $\mathcal{O}(\frac{n\beta^2}{\alpha^3}\log^2 T)$ (n is the dimension of the domain space) with probability at least $1-\delta$. More recently, [191] provided a new algorithm with an improved $\mathcal{O}(\frac{\beta^2}{\alpha}\log T)$ for the problem.

See the Appendix for a more detailed review of prior works.

8.3.2 Contributions

In this chapter, we study online continuous DR-submodular maximization in three different online settings. Our contributions are listed as follows:

- In Section 8.4, we propose Algorithm 1 which achieves logarithmic regret bound in the adversarial setting with strongly DR-submodular utility functions $\{f_t\}_{t=1}^T$.
- We consider the random order model in Section 8.5, where the utility functions are DR-submodular and chosen adversarially but they arrive in a uniformly random order. For this setting, we leverage concentration inequalities for sampling without replacement as our

Paper	Setting	Approx. Ratio	Regret	Guarantee
[61]	adversarial, DR-submodular	$1 - 1/e$	$\mathcal{O}(\sqrt{T})$	deterministic
Algorithm 1	adversarial, strongly DR-submodular	$1 - 1/e$	$\mathcal{O}(\ln T)$	deterministic
Algorithm 1	random order, DR-submodular$^{(a)}$	$1 - 1/e$	$\mathcal{O}(\ln^2 T)$	w.h.p.
[60]	i.i.d., DR-submodular	$1/e$	$\mathcal{O}(T^{2/3})$	in expectation
Algorithm 2	i.i.d., DR-submodular	$1 - 1/e$	$\mathcal{O}(\sqrt{T})$	in expectation, w.h.p.
Algorithm 3	i.i.d., DR-submodular	$1/e$	$\mathcal{O}(\sqrt{T})$	in expectation, w.h.p.

Table 8.1: Comparison of online DR-submodular algorithms. Note that in (a), we also assume that $\frac{1}{T}\sum_{t=1}^{T} f_t$ is strongly DR-submodular.

main tool to show that if the average of utility functions is strongly DR-submodular, we can exploit Algorithm 1 to achieve logarithmic regret bound even if each utility function is only DR-submodular.

- In Section 8.6, we study the i.i.d. model where the sequence of generally non DR-submodular utility functions $\{f_t\}_{t=1}^{T}$ are drawn from an unknown distribution $\mathcal{D}$ with DR-submodular mean f, i.e., $\mathbb{E}_{\mathcal{D}}[f_t(\cdot)] = f(\cdot)$, and we only have access to unbiased stochastic gradient estimates of $\{f_t\}_{t=1}^{T}$. We propose Algorithm 2 and Algorithm 3 for this setting, and we obtain nearly optimal $\tilde{\mathcal{O}}(\sqrt{T})$ stochastic regret bounds, both in expectation and with high probability for each algorithm. Algorithm 2 manages to obtain the optimal approximation ratio $1 - \frac{1}{e}$, however it requires $\mathcal{O}(T^{5/2})$ overall gradient evaluations. On the other hand, Algorithm 3 only achieves a $\frac{1}{e}$ approximation ratio while requiring a much lower $\mathcal{O}(T)$ total gradient evaluations.

Finally, in Section 8.7, we test our online algorithms through numerical experiments on synthetic and real-world datasets and highlight their superiority compared to previous techniques for the aforementioned three online frameworks. A summary of the results of this chapter in comparison with prior works is presented in Table 8.1. All missing proofs are provided in the Appendix.

In this chapter, we have assumed that the horizon T is known in advance. However, had we not known T beforehand, we could have used the well-known doubling trick [10] to obtain the same regret bounds with slightly worse constants.

8.4 Online strongly DR-submodular maximization in the adversarial setting

We first define strongly DR-submodular functions below.

Definition 8.4.1 ***(Strong DR-submodularity [183])*** *A twice differentiable function* $f : \mathcal{K} \to \mathbb{R}$, $\mathcal{K} \subset \mathbb{R}^n_+$*, is* μ*-strongly DR-submodular if for all* $x \in \mathcal{K}$*, we have* $\nabla^2_{ii} f(x) \leq -\mu \ \forall i \in [n]$ *and* $\nabla^2_{ij} f(x) \leq 0 \ \forall i \neq j$*.*

Note that if $\mu = 0$, f is DR-submodular. Definition 8.4.1 states that for a twice differentiable DR-submodular function, if $\nabla^2_{ii} f(x) \leq -\mu$ holds for all $i \in [n]$ and $x \in \mathcal{K}$, the function is μ-strongly DR-submodular. For instance, consider a graph $G = (V, E)$ with adjacency matrix A. Computing the stability number $s(G)$ of the graph could be formulated as $s(G)^{-1} = \min_{x \in \Delta} x^T(A + I)x$ where Δ is the simplex (Section 6.1 of [27]). Since the Hessian of $x^T(-A - I)x + 21^T x$ is $-2A - 2I$, the function is 2-strongly DR-submodular and therefore, the problem of finding the stability number of the graph could be formulated as a convex-constrained monotone strongly DR-submodular maximization problem.

Now, consider the following adversarial online optimization problem over T rounds: At each step $t \in [T]$, the algorithm chooses a decision variable $x_t \in \mathcal{K}$, where $\mathcal{K}$ is the fixed convex and compact domain with diameter R and $0 \in \mathcal{K}$. Once the algorithm commits to the action x_t, the adversary reveals a monotone utility function f_t which is μ-strongly DR-submodular and L-smooth. Without loss of generality and for ease of notation, we assume the utility functions are normalized, i.e., $f_t(0) = 0 \ \forall t \in [T]$. The overall goal is to maximize the cumulative utility $\sum_{t=1}^T f_t(x_t)$ or equivalently, minimize the $(1 - \frac{1}{e})$-regret R_T defined as:

$$R_T = (1 - \frac{1}{e}) \max_{x \in \mathcal{K}} \sum_{t=1}^T f_t(x) - \sum_{t=1}^T f_t(x_t),$$

where $1 - \frac{1}{e}$ is the optimal polynomial time approximation ratio for offline monotone DR-submodular maximization. In other words, the regret compares the decisions of the algorithm

Algorithm 1 Online strongly DR-submodular maximization algorithm

```
Input: $K > 0$ is the number of inner loops, $T$ is the horizon, and $\mu > 0$.
Output: $\{x_t : 1 \leq t \leq T\}$.
Choose an off-the-shelf online strongly concave maximization algorithm and initialize $K$ instances $\{\mathcal{E}_k\}_{k=1}^K$ of it for online maximization of $\mu$-strongly concave utility functions over $\mathcal{K}$.
for $t = 1$ to $T$ do
  Set $x_t^{(1)} = 0$.
  for $k = 1$ to $K$ do
    Let $v_t^{(k)}$ be the vector selected by $\mathcal{E}_k$.
    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  end for
  Play $x_t = x_t^{(K+1)}$, observe the function $f_t$ and the reward $f_t(x_t)$.
  Feedback $\langle v_t^{(k)}, \nabla f_t(x_t^{(k)}) \rangle - \frac{\mu}{2} \|v_t^{(k)}\|^2$ as the payoff to be received by $\mathcal{E}_k$.
end for
```

with the $(1 - \frac{1}{e})$-approximation to the optimal solution in hindsight.

For the setting with $\mu = 0$, i.e., when all utility functions are just DR-submodular, [61] proposed the Meta-Frank-Wolfe algorithm with a provably optimal $\mathcal{O}(\sqrt{T})$ regret. Since for all $t \in [T]$, f_t remains unknown until the algorithm chooses the action x_t, the Meta-Frank-Wolfe algorithm runs $K = \sqrt{T}$ instances of a no-regret online linear maximization algorithm (such as Follow the Regularized Leader (FTRL)) to mimic the Frank-Wolfe variant of [25] for offline DR-submodular maximization in the online setting, and outputs the average of the decisions of these instances at each round. We now propose a modified version of the Meta-Frank-Wolfe algorithm of [61] for strongly DR-submodular utility functions to be able to obtain improved logarithmic regret bound in this case. The algorithm is presented in Algorithm 1. The algorithm runs K instances $\{\mathcal{E}_k\}_{k=1}^K$ of no-regret online strongly concave maximization algorithms (such as Follow the Leader (FTL)) where at each round $t \in [T]$, the instance $\mathcal{E}_k$ chooses the action $v_t^{(k)}$ and upon committing to this decision, it receives a μ-strongly concave payoff of $\langle v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu}{2}\|v_t^{(k)}\|^2$. Algorithm 1 outputs $x_t = \frac{1}{K}\sum_{k=1}^{K} v_t^{(k)}$. Note that the FTL update rule for $v_t^{(k)}$ is computationally efficient in many cases. To be precise, we can write:

$$\begin{aligned} v_t^{(k)} &= \arg\max_{x \in \mathcal{K}} \ \langle x, \sum_{s=1}^{t-1} \nabla f_s(x_s^{(k)})\rangle - \frac{\mu(t-1)}{2}\|x\|_2^2 \\ &= \mathrm{Proj}_{\mathcal{K}}\big(\frac{1}{\mu(t-1)}\sum_{s=1}^{t-1}\nabla f_s(x_s^{(k)})\big), \end{aligned}$$

where $\mathrm{Proj}_{\mathcal{K}}$ denotes Euclidean projection onto set $\mathcal{K}$.

The regret guarantee of Algorithm 1 is provided below.

Theorem 8.4.1 *Assume that $\{f_t\}_{t=1}^T$ are normalized (i.e., $f_t(0) = 0$), monotone, μ-strongly DR-submodular, L-smooth and β-Lipschitz with respect to $\|\cdot\|_2$. Using Algorithm 1 with $K = \lceil \frac{L}{\mu} \rceil$, we have:*

$$R_T = (1 - \frac{1}{e})\sum_{t=1}^{T} f_t(x^*) - \sum_{t=1}^{T} f_t(x_t) \leq \mathcal{O}(\ln T),$$

where $x^* = \arg\max_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$.

Note that while the Meta-Frank-Wolfe algorithm of [61] requires $K = \sqrt{T}$ parallel online algorithms to achieve an $O(\sqrt{T})$ regret bound, Algorithm 1 only requires $\lceil \frac{L}{\mu} \rceil$ (independent of T) such subroutines to obtain an improved logarithmic regret bound. In other words, Algorithm 1 has a better utility performance while being computationally more efficient as well. Moreover, compared to the $(\frac{1}{1+c})$-regret guarantee of OGA provided in the previous chapter, Algorithm 1 achieves a higher approximation ratio of $1 - \frac{1}{e}$ for general strongly DR-submodular functions.

8.5 Online DR-submodular maximization in the random order model

In this section, we focus on the random order adversary model where the sequence of objective functions $\{f_t\}_{t=1}^T$ is still chosen adversarially, but they arrive in a uniformly random order.

In other words, each of the $T!$ permutations of the ordering of arbitrarily chosen utility functions is equally likely to happen. Specifically, we first consider an online DR-submodular maximization problem in which the DR-submodular utility function at step $t \in [T]$ is quadratic, $f_t(x) = \frac{1}{2}x^T A^{(t)} x + (a^{(t)})^T x$ where $A^{(t)}$ is symmetric, every off-diagonal entry $A_{ij}^{(t)}$ $\forall i \neq j \in [n]$ is in the range $[-L, 0]$, and every diagonal entry of the average matrix $\frac{1}{T}\sum_{t=1}^{T} A^{(t)}$ is in the range $[-L, -\mu]$ while $|A_{ii}^{(t)}| \leq L$ for all $i \in [n]$. In other words, each utility function $\{f_t\}_{t=1}^{T}$ is DR-submodular, and the average of utility functions $\frac{1}{T}\sum_{t=1}^{T} f_t$ is μ-strongly DR-submodular with respect to $\|\cdot\|_2$ and L-smooth with respect to $\|\cdot\|_1$. We first provide the following concentration inequality for randomly permuted sums.

Theorem 8.5.1 *(Theorem 4.3 of [20]) Let* $\{c_{r,s}\}_{r,s\in[T]}$ *be an array of real numbers from the range* $[-m_c, m_c]$*. Let* $Z_T = \sum_{r=1}^{T} c_{r,\prod_T(r)}$ *where* $\prod_T$ *is drawn from the uniform distribution over the set of all permutations of* $\{1, \ldots, T\}$*. Then, for any* $\lambda > 0$*, we have:*

$$\mathbb{P}\big(|Z_T - \mathbb{E}[Z_T]| \geq \lambda\big) \leq 4\exp\big(-\frac{\lambda^2}{16(\frac{\theta}{T}\sum_{r,s=1}^{T} c_{r,s}^2 + \frac{m_c\lambda}{3})}\big),$$

where $\theta = \frac{5}{2}\ln 3 - \frac{2}{3}$*.*

Now, we apply the result of Theorem 8.5.1 to our problem. Let $W \in [1, T]$. For a fixed $i \in [n]$, set:

$$c_{r,s} = \begin{cases} A_{ii}^{(s)} - \frac{1}{T}\sum_{t=1}^{T} A_{ii}^{(t)} & \text{if } 1 \leq r \leq W \\ 0 & \text{o.w.} \end{cases}$$

We can write:

$$Z_T = \sum_{r=1}^{T} c_{r,\Pi_T(r)} = \sum_{r=1}^{W} A_{ii}^{(\Pi_T(r))} - \frac{W}{T}\sum_{t=1}^{T} A_{ii}^{(t)}.$$

Taking the expectation of the above equality, we obtain:

$$\begin{aligned} \mathbb{E}[Z_T] &= \sum_{r=1}^{W} \mathbb{E}[A_{ii}^{(\Pi_T(r))}] - \frac{W}{T}\sum_{t=1}^{T} A_{ii}^{(t)} \\ &= \sum_{r=1}^{W}(\frac{1}{T}\sum_{t=1}^{T} A_{ii}^{(t)}) - \frac{W}{T}\sum_{t=1}^{T} A_{ii}^{(t)} \\ &= 0. \end{aligned}$$

Since $m_c = 2L$ in our setting, we have $\frac{\theta}{T}\sum_{r,s=1}^{T} c_{r,s}^2 = \frac{\theta}{T}\sum_{r=1}^{W}\sum_{s=1}^{T}\big(A_{ii}^{(s)} - \frac{1}{T}\sum_{t=1}^{T} A_{ii}^{(t)}\big)^2 \leq 4\theta L^2 W$. Plugging in $\lambda = W\epsilon$, we obtain:

$$\mathbb{P}\big(|\frac{1}{W}\sum_{r=1}^{W} A_{ii}^{(\Pi_T(r))} - \frac{1}{T}\sum_{t=1}^{T} A_{ii}^{(t)}| \geq \epsilon\big) = \mathbb{P}\big(|\sum_{r=1}^{W} A_{ii}^{(\Pi_T(r))} - \frac{W}{T}\sum_{t=1}^{T} A_{ii}^{(t)}| \geq W\epsilon\big)$$
$$\leq 4\exp\big(-\frac{W^2\epsilon^2}{16(4\theta L^2 W + \frac{2LW\epsilon}{3})}\big).$$

$\mathbb{P}\big(|\frac{1}{W}\sum_{r=1}^{W} A_{ii}^{(\Pi_T(r))} - \frac{1}{T}\sum_{t=1}^{T} A_{ii}^{(t)}| \geq \epsilon\big)$ is then bounded from above by

$$\leq \begin{cases} 4\exp\big(-\frac{W\epsilon^2}{128\theta L^2}\big) & \epsilon \leq 6\theta L \\ 4\exp\big(-\frac{3W\epsilon}{64L}\big) & \text{o.w.} \end{cases}.$$

If we choose $\epsilon \leq \frac{\mu}{2}$, the first case happens. Therefore, if we set $\frac{\delta}{nT} = 4\exp\big(-\frac{W\epsilon^2}{128\theta L^2}\big)$, we can use the union bound to conclude that for every $i \in [n]$ and $\tau \in \{0, \dots, \lceil\frac{T}{W}\rceil - 1\}$, with probability at least $1-\delta$, it holds that $\frac{1}{W}\sum_{t=\tau W+1}^{(\tau+1)W} A_{ii}^{(t)} \leq -(\mu - \epsilon) \leq -\frac{\mu}{2}$ for any $W \geq W_0$ where $W_0 = \frac{128\theta L^2}{\epsilon^2}\ln(\frac{4nT}{\delta})$. In other words, for large enough W, the average of each of the consecutive blocks of size W of the utility functions is $(\frac{\mu}{2})$-strongly DR-submodular.

Now, consider general monotone DR-submodular utility functions $\{f_t\}_{t=1}^{T}$. In this case, the Hessian of the utility functions is not fixed and therefore, we have to ensure the strong DR-submodularity of every block of size W at every point $x \in \mathcal{K}$. The analysis in this setting is provided in the Appendix.

Considering that the strong DR-submodularity property propagates to the average of utility functions over sufficiently large blocks, we can apply Algorithm 1 to these blocks to obtain logarithmic regret bounds. We summarize this result in the following theorem.

Theorem 8.5.2 *Assume that for all $t \in [T]$, the utility function f_t is normalized, monotone and DR-submodular, and the average of utility functions $\frac{1}{T}\sum_{t=1}^{T} f_t$ is μ-strongly DR-submodular and L-smooth. Then, with probability at least $1-\delta$, if we apply Algorithm 1 to the utility functions $\{\frac{1}{W}\sum_{t=\tau W+1}^{(\tau+1)W} f_t\}_{\tau=0}^{\lceil\frac{T}{W}\rceil-1}$ for $W \geq W_0 = \mathcal{O}(\frac{L^2}{\mu^2}\ln(\frac{nT}{\delta}))$, we have:*

$$\frac{1}{W}\sum_{\tau=0}^{\lceil\frac{T}{W}\rceil-1}\sum_{t=\tau W+1}^{(\tau+1)W}\big((1-\frac{1}{e})f_t(x^*) - f_t(z_\tau)\big) \leq \mathcal{O}(\ln\frac{T}{W}).$$

Therefore, if we set $x_t = z_\tau$ *for all* $\tau \in \{0, \ldots, \lceil \frac{T}{W} \rceil - 1\}$ *and* $t \in \{\tau W + 1, \ldots, (\tau+1)W\}$*, the following holds with probability at least* $1 - \delta$*:*

$$R_T \leq \mathcal{O}(W \ln \frac{T}{W}) = \mathcal{O}(\ln(\frac{nT}{\delta}) \ln T).$$

It is noteworthy that while the idea of propagation of the strong convexity property of the overall average function to the average over sufficiently large blocks in [92] is similar to the idea used here for the DR-submodular setting, the techniques and *matrix* concentration inequalities used to guarantee strong convexity of the Hessian matrix in [92] is quite different than the *entry-wise* concentration inequalities used in our work to ensure that entries of the Hessian matrix satisfy the box constraints enforced by the strong DR-submodularity and smoothness properties.

8.6 Online DR-submodular maximization in the i.i.d. model

In this section, we focus on the setting where the sequence of utility functions $\{f_t\}_{t=1}^T$ is drawn i.i.d. from some fixed unknown distribution $f_t \sim \mathcal{D}$ where $\mathbb{E}_\mathcal{D}[f_t(\cdot)] = f(\cdot)$. In this framework, the performance is measured via the notion of stochastic regret defined as $\alpha\text{-SR}_T = T\alpha f(x^*) - \sum_{t=1}^T f(x_t)$, where $x^* = \arg\max_{x \in \mathcal{K}} f(x)$.

Assumption 1. We make the following assumptions on the utility functions:

- For all $t \in [T]$ and $x \in \mathcal{K}$, we only have access to the unbiased gradient oracle $\tilde{\nabla} f_t(x)$, i.e., $\mathbb{E}[\tilde{\nabla} f_t(x)] = \nabla f(x)$.
- There exists $\sigma > 0$ such that for any $x \in \mathcal{K}$ and $t \in [T]$, $\|\tilde{\nabla} f_t(x) - \nabla f(x)\|_2 \leq \sigma$ holds.
- f is monotone DR-submodular and L-smooth.
- $\{f_t\}_{t=1}^T$ are L-smooth.

Note that there are several differences between the adversarial and i.i.d. settings that are as follows: 1) In the i.i.d. setting, it is assumed that only the expected function $f(\cdot) = \mathbb{E}[f_t(\cdot)]$ is DR-submodular while each utility function $f_t(\cdot)$ may not be necessarily DR-submodular. However, in the adversarial setting, all utility functions $\{f_t\}_{t=1}^T$ are assumed to be DR-submodular. 2) The regret metric used in the adversarial and i.i.d. settings are quite different

Algorithm 2

```
Input: $\{K_t = \sqrt{t}\}_{t=1}^{T}$ and $T > 0$ is the horizon.
for $t = 1, \dots, T$ do
  Play $x_t = x_t^{(K_{t-1}+1)}$ and observe $f_t$.
  Set $x_{t+1}^{(1)} = 0$.
  for $k = 1, 2, \dots, K_t$ do
    $d_t^{(k)} = \frac{1}{t}\sum_{\tau=1}^{t} \tilde{\nabla} f_\tau(x_{t+1}^{(k)})$.
    $v_t^{(k)} = \arg\max_{x\in\mathcal{K}} \langle x, d_t^{(k)}\rangle$.
    Set $x_{t+1}^{(k+1)} = x_{t+1}^{(k)} + \frac{1}{K_t} v_t^{(k)}$.
  end for
end for
```

from each other. The adversarial regret is defined with respect to the utility functions $\{f_t\}_{t=1}^{T}$. On the other hand, the stochastic regret metric is in terms of the expected function f. If f was known in advance, we would use the Frank-Wolfe variant of [25] for offline DR-submodular maximization. To be precise, for all $t+1 \in [T]$, starting from $x_{t+1}^{(1)} = 0$, we would perform K_t Frank-Wolfe updates where at each iteration $k \in [K_t]$, we choose $v_t^{(k)}$ according to $v_t^{(k)} = \arg\max_{x\in\mathcal{K}} \langle x, \nabla f(x_{t+1}^{(k)})\rangle$, and perform the update $x_{t+1}^{(k+1)} = x_{t+1}^{(k)} + \frac{1}{K_t} v_t^{(k)}$. However, f is not available in advance. To tackle this problem, at round $t+1 \in [T]$, we can use the average of utility functions $\{f_s\}_{s=1}^{t}$ observed so far as an estimate of f. Using this technique, we propose Algorithm 2 and provide its stochastic regret guarantee in the theorem below.

Theorem 8.6.1 *For online DR-submodular maximization in the i.i.d. model, if Assumption* 1 *holds, the expected stochastic regret of Algorithm 2 is as follows:*

$$\mathbb{E}[(1-\frac{1}{e})\text{-}SR_T] \leq \sum_{t=1}^{T} \frac{LR^2}{2K_t} + \mathcal{O}(\sigma\sqrt{T}).$$

Moreover, with probability at least $1-\delta$*, we have:*

$$(1-\frac{1}{e})\text{-}SR_T \leq \sum_{t=1}^{T} \frac{LR^2}{2K_t} + \mathcal{O}(\sigma\sqrt{T\ln(\frac{T^{3/2}}{\delta})}).$$

Algorithm 3

```
Input: $\{\rho_t = \frac{1}{t+1}\}_{t=1}^{T}$ and $T > 0$ is the horizon.
for $t = 1, \dots, T$ do
    Play $x_t$ and observe $f_t$.
    if $t = 1$ then
        $d_t = \tilde{\nabla} f_t(x_t)$.
    else
        $d_t = \tilde{\nabla} f_t(x_t) + (1 - \rho_t)\big(d_{t-1} - \tilde{\nabla} f_t(x_{t-1})\big)$.
    end if
    $v_t = \arg\max_{x \in \mathcal{K}} \langle x, d_t \rangle$.
    Set $x_{t+1} = x_t + \frac{1}{T} v_t$.
end for
```

Therefore, if we set $K_t = \sqrt{t}\ \forall t \in [T]$*, Algorithm 2 obtains an* $\tilde{O}(\sqrt{T})$ *stochastic regret bound, both in expectation and with high probability.*

In the convex setting, $\mathcal{O}(\sqrt{T})$ stochastic regret bound is optimal and the same lower bound extends to the DR-submodular framework as well. Algorithm 2 manages to obtain the nearly optimal $\tilde{\mathcal{O}}(\sqrt{T})$ stochastic regret bound. However, in each step $t \in [T]$, the algorithm requires gradient oracle access at $\mathcal{O}(t^{3/2})$ points which leads to $\mathcal{O}(T^{5/2})$ overall gradient evaluations. Therefore, in applications where computing the gradient of utility functions is computationally expensive, Algorithm 2 has a high computational cost and may not be suitable.

To remedy this issue, inspired by the variance-reduction technique of [208], we propose Algorithm 3. In this algorithm, for all $t \in [T]$, we estimate $\nabla f(x_t)$ using the recursive estimator $d_t = \tilde{\nabla} f_t(x_t) + (1 - \rho_t)\big(d_{t-1} - \tilde{\nabla} f_t(x_{t-1})\big)$. The regret bound is provided below.

Theorem 8.6.2 *For online DR-submodular maximization in the i.i.d. model, if Assumption*

1 *holds, Algorithm 3 obtains the following stochastic regret bound in expectation:*

$$\mathbb{E}[(\frac{1}{e})\text{-}SR_T] \leq \mathcal{O}(\sigma\sqrt{T}).$$

Also, the following holds with probability at least $1-\delta$*:*

$$(\frac{1}{e})\text{-}SR_T \leq \mathcal{O}(\sigma\sqrt{T\ln(\frac{T}{\delta})}).$$

At each round $t \in [T]$, Algorithm 2 requires computing the gradient at $\mathcal{O}(t^{3/2})$ points whereas Algorithm 3 requires only 2 gradient evaluations per step. Therefore, the overall number of gradient evaluations of Algorithm 2 and Algorithm 3 are $\mathcal{O}(T^{5/2})$ and $\mathcal{O}(T)$ respectively. Moreover, both algorithms require solving just 1 linear optimization problem over the constraint set $\mathcal{K}$ per round. However, despite the lower computational complexity of Algorithm 3, this algorithm only manages to obtain an $\mathcal{O}(\sqrt{T})$ bound for the $(\frac{1}{e})$-stochastic regret while Algorithm 2 obtains similar bounds for the $(1-\frac{1}{e})$-regret where $1-\frac{1}{e}$ is the optimal approximation ratio for offline DR-submodular maximization.

The only prior study of online DR-submodular maximization in the i.i.d. model was done by [60] in which the authors proposed the OSFW algorithm with a sub-optimal $\mathcal{O}(T^{2/3})$ $(1-\frac{1}{e})$-stochastic regret bound in expectation. The OSFW algorithm requires only 1 gradient evaluation per round, however, there is a mistake in the analysis of the regret bound (using the inequality $(1-\frac{1}{T})^t \leq \frac{1}{e}$ for all $t \in [T]$ which is incorrect) and consequently, the approximation ratio is $\frac{1}{e}$. Therefore, while Algorithm 3 and the OSFW algorithm of [60] both perform just $O(1)$ gradient evaluations per step, the variance reduction technique used in Algorithm 3 leads to an improved $O(\sqrt{T})$ bound for the $(\frac{1}{e})$-regret.

8.7 Numerical examples

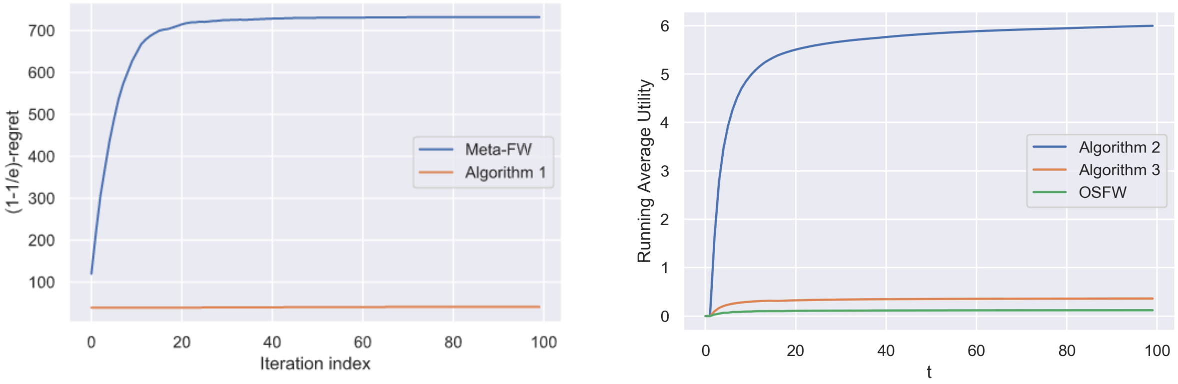
For the first experiment, we used the MovieLens dataset [100], containing anonymous ratings of approximately 3900 movies made by 6040 MovieLens users, and we studied a movie recommendation problem. We extracted 17 movies and 100 users with the most number of ratings. Therefore, $n = 17$ and $T = 100$. For all $t \in [T]$, the utility function f_t is defined as

$f_t(x) = 5\sum_{i=1}^{n} \ln(1 + R_i^{(t)} x_i) + \sum_{i,j:i<j} \theta_{ij}^{(t)} x_i x_j$ where $R_i^{(t)} \in [0, 1]$ is the rescaled rating of t-th online user for the i-th movie, and $\theta_{ij}^{(t)}$ is uniformly distributed in the range $[-1, 0]$ if the i-th and j-th movies are from the same genre. In other words, the utility function captures the diversity of the recommended movies. We chose $\mathcal{K} = \{x \in \mathbb{R}^n : 1^T x \leq 4, 0 \preceq x \preceq 1\}$, i.e., the algorithm has to recommend 4 movies to each arriving user. We ran Algorithm 1 and the Meta-Frank-Wolfe algorithm of [61], and plotted the $(1 - \frac{1}{e})$-regret versus the number of iterations in Figure 8.1(a). This plot verifies that the regret of Algorithm 1 is significantly smaller than the Meta-Frank-Wolfe algorithm for strongly DR-submodular functions.

For the second experiment, we studied online DR-submodular maximization in the i.i.d. model. We set $m = 2$, $n = 4$ and $T = 100$. We defined $\mathcal{K} = \{x \in \mathbb{R}^n : Cx \preceq b, 0 \preceq x \preceq 1\}$, where $C \in \mathbb{R}^{m \times n}$ is a matrix whose entries are uniformly distributed in the range $[0, 1]$ and $b \in \mathbb{R}^m$ is the vector of all ones. For the underlying utility function $f(x) = (\frac{1}{2}x - \mathbf{1})^T Ax$, we chose A to be a random matrix whose entries are sampled uniformly at random from the range $[-1, 0]$. Therefore, f is DR-submodular and 1-smooth. For all $t \in [T]$, we set $A^{(t)} = A + N^{(t)}$, where $N^{(t)}$ is a matrix with entries uniformly distributed in the range $[-4, 4]$. Thus, $\{A^{(t)}\}_{t=1}^{T}$ is an unbiased estimator of A. Note that since $\{A^{(t)}\}_{t=1}^{T}$ may have positive entries, each individual utility function $\{f_t\}_{t=1}^{T}$ is not necessarily DR-submodular. The running average of the utility of Algorithm 2, Algorithm 3, and the OSFW algorithm of [60] is depicted in Figure 8.1(b). It can be observed that the regret of Algorithm 2 is smaller than the other two algorithms because the average of the utility functions observed so far is a more accurate estimation of f and has a lower variance at the expense of higher computational complexity.

8.8 Conclusion and future work

We studied online DR-submodular maximization in three different settings. First, we focused on the adversarial setting and provided Algorithm 1 with logarithmic regret bound. Next, we considered online DR-submodular maximization in the random order model and showed how we can use Algorithm 1 to obtain similar logarithmic regret bounds with high probability



(a)

(b)

Figure 8.1: (a) Experiment 1, and (b) Experiment 2.

despite each utility function not being DR-submodular. Finally, we focused on online DR-submodular maximization in the i.i.d. model, and we provided two algorithms with nearly optimal $\tilde{\mathcal{O}}(\sqrt{T})$ regret bounds, both in expectation and with high probability. This work could be extended in several directions. It is interesting to see whether Algorithm 1 could be used in a best-of-both-worlds fashion for the adversarial and random order frameworks, i.e., if we can obtain logarithmic regret bounds in the random order model by simply applying Algorithm 1 on the individual utility functions (instead of blocks of functions). Moreover, providing an algorithm for the i.i.d. setting with optimal $\mathcal{O}(\sqrt{T})$ $(1-\frac{1}{e})$-stochastic regret bound and $\mathcal{O}(T)$ overall gradient evaluations is yet to be done.

Part III

SUBMODULAR MAXIMIZATION UNDER SOCIAL AND ECONOMIC CONSIDERATIONS

Chapter 9

DIFFERENTIALLY PRIVATE MONOTONE SUBMODULAR MAXIMIZATION UNDER MATROID AND KNAPSACK CONSTRAINTS

9.1 Chapter overview

This chapter presents the joint work with Maryam Fazel that was published at the International Conference on Artificial Intelligence and Statistics (AISTATS) 2021. Numerous tasks in machine learning and artificial intelligence have been modeled as submodular maximization problems. These problems usually involve sensitive data about individuals, and in addition to maximizing the utility, privacy concerns should be considered. In this chapter, we study the general framework of non-negative monotone submodular maximization subject to matroid or knapsack constraints in both offline and online settings. For the offline setting, we propose a differentially private $(1 - \frac{\kappa}{e})$-approximation algorithm, where $\kappa \in [0, 1]$ is the total curvature of the submodular set function, which improves upon prior works in terms of approximation guarantee and query complexity under the same privacy budget. In the online setting, we propose the first differentially private algorithm, and we specify the conditions under which the regret bound scales as $\mathcal{O}(\sqrt{T})$, i.e., privacy could be ensured while maintaining the same regret bound as the optimal regret guarantee in the non-private setting.

9.2 Introduction

The submodularity property of set functions has profound theoretical consequences and far-reaching applications. Submodular set functions play a significant role in combinatorial optimization as many well-known combinatorial functions are indeed submodular. Cut functions of graphs and hypergraphs, rank functions of matroids and covering functions are a few examples of submodular set functions. Moreover, submodularity has been identified and utilized in applications such as viral marketing [118], feature selection for classification [127], image segmentation [123, 32] and document summarization [136, 120]. The multilinear extension $f : [0,1]^V \to \mathbb{R}$ of F is defined as [46]

$$f(x) = \sum_{S \subset V} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) = \mathbb{E}_{S \sim x}[F(S)].$$

Multilinear extensions coupled with lossless rounding techniques are extensively used for maximizing the corresponding submodular set functions. In particular, for submodular maximization subject to matroid constraints, [46] and [58] proposed the pipage rounding and swap rounding schemes respectively to round the fractional solution without losing in terms of the objective function. [128, 129] provided lossless rounding techniques for knapsack constraints. It has been shown that multilinear extensions can be efficiently computed for a large class of submodular set functions, for example, the weighted matroid rank function, set cover function, probabilistic coverage function, and graph cut function.

In applications where the submodular function involves sensitive data about individuals, privacy concerns should be addressed as well as obtaining good approximation guarantees. For instance, consider the following feature selection problem [127, 149]:

Example 9.2.1 *Let $D = \{(x_t, C_t)\}_{t=1}^T$ be a sensitive dataset consisting of a feature vector $x_t \in \mathbb{R}^m$ for each individual $t \in [T]$ along with a binary class label C_t. The goal is to select a small subset of the m features that provide a good classifier for the labels. In particular, determining the likeliness of an individual having a certain disease using a representative collection of his or her features (such as height, age, and weight) could be cast in this framework.*

In order to solve this problem, [127] proposed a non-private algorithm based on maximizing a submodular function capturing the mutual information between a subset of the features and the class label of interest. However, in this setting, along with obtaining the most relevant subset of features, it is crucial to ensure that the privacy of any individual included in the dataset is not compromised. See [148] for more applications (such as personal data summarization) that could be cast as submodular problems under more general constraints in which privacy concerns should be addressed.

9.2.1 Preliminaries

Matroids and matroid polytopes. A matroid $\mathcal{M}$ is a pair $\mathcal{M} = (V, \mathcal{I})$, where V is a finite ground set and $\mathcal{I}$ is a collection of subsets of V called the independent sets, that satisfies the following properties: 1) $\emptyset \in \mathcal{I}$. 2) For $S' \subset S \subset V$, if $S \in \mathcal{I}$, then $S' \in \mathcal{I}$ holds. 3) For $S, S' \in \mathcal{I}$, if $|S| > |S'|$, there exists $v \in S \setminus S'$ such that $S' \cup \{v\} \in \mathcal{I}$. The matroid polytope corresponding to the matroid $\mathcal{M} = (V, \mathcal{I})$ is defined as

$$P(\mathcal{M}) = \mathrm{conv}\{1_I : I \in \mathcal{I}\} = \{x \succeq 0 : \sum_{s \in S} x_s \leq r_{\mathcal{M}}(S),\ \forall S \subset V\},$$

where the rank function $r_{\mathcal{M}} : 2^V \to \mathbb{Z}_+$ is $r_{\mathcal{M}}(S) = \max\{|I| : I \subseteq S, I \in \mathcal{I}\}$ and conv denotes the convex hull. We define $\mathrm{rank}(\mathcal{M}) = r_{\mathcal{M}}(V)$ as the rank of the matroid $\mathcal{M}$.

Knapsack constraints and knapsack polytopes. Given a ground set V, a positive vector $c \in \mathbb{R}_{++}^{|V|}$ and a collection $\mathcal{I} = \{S \subseteq V : \sum_{s \in S} c_s \leq 1\}$ of subsets of V, $S \in \mathcal{I}$ is called a knapsack constraint. The natural continuous relaxation $\{x \in [0,1]^{|V|} : c^T x \leq 1\}$ is the knapsack polytope corresponding to $\mathcal{I}$.

9.2.2 Related work

Non-private submodular maximization. Maximizing non-negative monotone submodular set functions under a certain constraint has been extensively studied in the literature in both offline and online settings. Consider the problem of maximizing the monotone submodular function $F(S)$ subject to a cardinality constraint $|S| \leq k$. For offline monotone submodular set function maximization subject to a cardinality constraint, [161] proposed a simple greedy algorithm that obtains the provably optimal approximation ratio of $1 - \frac{1}{e}$. At each round $i \in [k]$, the greedy algorithm constructs S_i from S_{i-1} by adding the element $v_i \in V \setminus S_{i-1}$ which maximizes the marginal gain $F(S_{i-1} \cup \{v_i\}) - F(S_{i-1})$. However, if

$\mathcal{M} = (V, \mathcal{I})$ is a matroid, the greedy algorithm applied to the submodular maximization problem subject to the matroid constraint $S \in \mathcal{I}$ achieves a sub-optimal $\frac{1}{2}$ approximation ratio. [46] proposed the continuous greedy algorithm for this problem which achieves the optimal $1 - \frac{1}{e}$ approximation ratio. The continuous greedy algorithm is applied to the multilinear extension of the submodular set function under the matroid polytope and is as follows:

$$\begin{aligned} dy/dt &= v_{\max}(y) \\ v_{\max}(y) &= \arg\max_{v \in P(\mathcal{M})} \langle v, \nabla f(y) \rangle \end{aligned},$$

where f is the multilinear extension of the submodular set function F and $P(\mathcal{M})$ is the matroid polytope corresponding to the matroid $\mathcal{M}$. In this algorithm, $y(1) = \int_0^1 v_{\max}(y(\tau))d\tau$ is the output. More recently, [197] introduced a modification of the continuous greedy algorithm with an approximation ratio of $1 - \frac{\kappa}{e}$, where $\kappa \in [0, 1]$ is the total curvature of the submodular set function, and proved that the derived approximation ratio is indeed optimal. Later on, [83] managed to obtain the same $1 - \frac{\kappa}{e}$ approximation ratio with an algorithm that is much faster than the one proposed by [197].

In the online setting, [61] proposed an online variant of the continuous greedy algorithm, called the Meta Frank-Wolfe algorithm, with a provably optimal $\mathcal{O}(\sqrt{T})$ regret bound, where T is the length of the horizon.

Offline differentially private submodular maximization. Let D be a sensitive dataset associated with a monotone submodular set function $F_D : 2^V \to \mathbb{R}_+$. Offline submodular maximization in the context of differential privacy has been studied in two different settings:

- F_D is Δ-decomposable [97, 55]: In this setting, it is assumed that $D = (F_1, \ldots, F_T)$, where for all $t \in [T]$, $F_t : 2^V \to [0, \Delta]$ is a *private* monotone submodular set function and $F_D(\cdot) = \frac{1}{T} \sum_{t=1}^{T} F_t(\cdot)$.
- F_D is Δ-sensitive [149, 173]: For this framework, we have $D = (F_1, \ldots, F_T)$, where for all $t \in [T]$, $F_t : 2^V \to \mathbb{R}_+$ is a *private* monotone submodular set function, however, the submodular objective function F_D depends on the dataset D in ways that could be much more complicated than simply averaging the private submodular functions $F_1, \ldots, F_T$ (e.g.,

Example 1). Two datasets D and D' are neighboring ($D \sim D'$) if all but one of the T submodular functions in the datasets are equal. It is assumed that F_D is Δ-sensitive, i.e., $\Delta = \max_{D':D'\sim D} \max_{S\subseteq V} |F_D(S) - F_{D'}(S)|$ holds.

Note that if F_D is Δ-decomposable, the sensitivity parameter is bounded above by $\frac{\Delta}{T}$, implying F_D is ($\frac{\Delta}{T}$)-sensitive. In this chapter, we focus on the more general setting where F_D is Δ-sensitive, and we review the prior work in this setting below. For submodular maximization subject to cardinality or matroid constraints, [149] combined the greedy algorithm with the exponential mechanism of [145] for differential privacy as follows: At round $i \in [k]$ of the greedy algorithm, define a quality function q_i via $q_i(v, D) = F_D(S_{i-1} \cup \{v\}) - F_D(S_{i-1})$, and select every $v \in V \setminus S_{i-1}$ with probability proportional to $\exp(\epsilon q(v, D)/2\lambda)$ where λ is the sensitivity of the quality function q, i.e., for all $v \in V$ and two neighboring datasets D and D', we have $|q(v, D) - q(v, D')| \leq \lambda$. [149] showed that this algorithm is ϵ-differentially private (see Section 9.3 for a formal definition of differential privacy) and obtained an expected utility bound of $(1 - \frac{1}{e})\mathrm{OPT} - \mathcal{O}(\frac{\Delta k^2 \ln(|V|)}{\epsilon})$ and $\frac{1}{2}\mathrm{OPT} - \mathcal{O}(\frac{\Delta(\mathrm{rank}(\mathcal{M}))^2 \ln(|V|)}{\epsilon})$ for submodular maximization subject to cardinality constraint $|S| \leq k$ and matroid constraint $\mathcal{M} = (V, \mathcal{I})$ respectively. However, the result of [149] for the setting with matroid constraints fails to achieve the optimal approximation ratio of $1 - \frac{1}{e}$. More recently, for submodular maximization subject to a matroid constraint $\mathcal{M} = (V, \mathcal{I})$, [173] combined the discretized version of the continuous greedy algorithm with the exponential mechanism in the following way: Let C_ρ be a ρ-covering of the matroid polytope $P(\mathcal{M})$, i.e., for any $x \in P(\mathcal{M})$, there exists $y \in C_\rho$ such that $\|x - y\|_2 \leq \rho$. At each round $k \in [K]$, where $K = \mathrm{rank}(\mathcal{M})$ is the rank of the matroid, the algorithm samples $y_k \in C_\rho$ with probability proportional to $\exp(\epsilon\langle y_k, \nabla f_D(x_k)\rangle/2\Delta)$, and sets $x_{k+1} = x_k + \frac{1}{K} y_k$ (where $x_1 = 0$). [173] showed that the output of this algorithm (x_{K+1}) obtains the utility bound $(1 - \frac{1}{e})\mathrm{OPT} - \mathcal{O}(\sqrt{\epsilon} + \frac{\Delta(\mathrm{rank}(\mathcal{M}))^7 |V| \ln(|V|)}{\epsilon^3})$ with high probability while ensuring ϵ-differential privacy. Although the result in [173] has the optimal $1 - \frac{1}{e}$ approximation ratio, it has several major drawbacks that are as follows:

- The $\mathcal{O}(\sqrt{\epsilon})$ term in the approximation guarantee is unusual, i.e., if $\epsilon \to \infty$ (no differential privacy), the approximation guarantee is vacuous.

- In order to ensure differential privacy, $\rho = \frac{\epsilon}{|V|}$ should hold and thus, $|C_\rho| = \mathcal{O}(|V|^{1+(\frac{\mathrm{rank}(\mathcal{M})}{\epsilon})^2})$. Therefore, discretization of the matroid polytope and implementing the exponential mechanism over such a large domain requires $\mathcal{O}(\mathrm{rank}(\mathcal{M})|V|^{1+(\frac{\mathrm{rank}(\mathcal{M})}{\epsilon})^2})$ gradient evaluation of the multilinear extension, and is not computationally efficient (although [173] provided a second algorithm with improved query complexity, it is still computationally expensive).
- The dependence of the additive factor $\mathcal{O}(\frac{\Delta(\mathrm{rank}(\mathcal{M}))^7|V|\ln(|V|)}{\epsilon^3})$ in the approximation guarantee on ϵ and $\mathrm{rank}(\mathcal{M})$ is not optimal.

After the publication of our work, [204] and [56] have studied the differentially private submodular maximization problem in the federated and streaming settings respectively.

Differentially private online learning. Online submodular maximization had not been studied under the context of differential privacy before the publication of our work (since then, [187] has studied this problem under a cardinality constraint of size k and proposed a (ϵ, δ)-differentially private algorithm with a $(1-\frac{1}{e})$-regret bound of $\mathcal{O}(\frac{k^2 \ln|V| \sqrt{T \ln(K/\delta)}}{\epsilon}))$. Nonetheless, the problem of differentially private online learning is extensively studied for linear and convex objective functions [77, 112, 96, 3]. In particular, [3] considered an online linear optimization problem over T rounds where at each step $t \in [T]$, the algorithm first chooses a point $x_t \in \mathcal{X}$ in the convex and compact domain set $\mathcal{X}$ and subsequently, it observes the loss vector ℓ_t and incurs a loss of $\langle \ell_t, x_t \rangle$. [3] proposed an ϵ-differentially private modification of the well-known Follow the Regularized Leader (FTRL) scheme for linear objectives with a regret bound that scales as $\mathcal{O}(\sqrt{T}) + \tilde{\mathcal{O}}(\frac{1}{\epsilon})$. Therefore, if $\epsilon \geq \Omega(\frac{1}{\sqrt{T}})$, the regret incurred by the differentially private algorithm matches the optimal $\mathcal{O}(\sqrt{T})$ regret in the non-private setting, i.e., differential privacy could be ensured for free. We use this algorithm as a sub-routine in our proposed algorithm for differentially private online submodular maximization.

9.2.3 Contributions

In this chapter, we study the general framework of differentially private monotone submodular maximization subject to matroid or knapsack constraints in both offline and online settings. Specifically, we make the following contributions:

- In Section 9.4.1, we propose the Differentially Private Continuous Greedy (DPCG) algorithm for offline monotone submodular maximization subject to matroid or knapsack constraints and we analyze its performance in both settings. The DPCG algorithm is ϵ-differentially private under both constraints. For matroid constraints, we obtain a utility bound of $(1-\frac{1}{e})\mathrm{OPT}-\mathcal{O}(\sqrt{\frac{\Delta(\mathrm{rank}(\mathcal{M}))^3|V|\ln(|V|)}{\epsilon}})$ with $\mathcal{O}(\sqrt{\frac{\epsilon.\mathrm{rank}(\mathcal{M})}{|V|\ln(|V|)\Delta}})$ multilinear extension evaluations which is a significant improvement over the $(1-\frac{1}{e})\mathrm{OPT}-\mathcal{O}(\sqrt{\epsilon}+\frac{\Delta(\mathrm{rank}(\mathcal{M}))^7|V|\ln(|V|)}{\epsilon^3})$ bound in [173] with $\mathcal{O}(\mathrm{rank}(\mathcal{M})|V|^{1+(\frac{\mathrm{rank}(\mathcal{M})}{\epsilon})^2})$ multilinear extension evaluations. Also, we obtain the first approximation guarantee for knapsack constraint $\{S\subseteq V:\sum_{s\in S}c_s\leq 1\}$ which is $(1-\frac{1}{e})\mathrm{OPT}-\mathcal{O}(\sqrt{\frac{\Delta|V|\ln(|V|)}{(c_{\min})^3\epsilon}})$, where $c_v\geq c_{\min},\ \forall v\in V$.
- For submodular functions with bounded curvature, we propose a modification of the DPCG algorithm, called the κ-DPCG algorithm, in Section 9.4.2 which has a utility bound of $(1-\frac{\kappa}{e})\mathrm{OPT}-\mathcal{O}(\sqrt{\frac{\Delta(\mathrm{rank}(\mathcal{M}))^3|V|\ln(|V|)}{\epsilon}})$ and $(1-\frac{\kappa}{e})\mathrm{OPT}-\mathcal{O}(\sqrt{\frac{\Delta|V|\ln(|V|)}{(c_{\min})^3\epsilon}})$ for matroid and knapsack constraints respectively, where $\kappa\in[0,1]$ is the total curvature of the submodular set function. In other words, compared to the DPCG algorithm, the κ-DPCG algorithm maintains the same additive factor in its utility bound while its $1-\frac{\kappa}{e}$ approximation ratio is strictly better than $1-\frac{1}{e}$ for submodular functions with curvature $\kappa<1$.
- In the online setting, we propose the first algorithm for (ϵ,δ)-differentially private (defined in Section 9.3) submodular maximization, namely the Differentially Private Meta Frank-Wolfe (DPMFW) algorithm, whose regret bound scales as $\mathcal{O}(\sqrt{T})+\mathcal{O}(\frac{T^{1/4}\sqrt{\ln(1/\delta)}}{\epsilon})$. Therefore, if $\frac{\sqrt{\ln(1/\delta)}}{\epsilon}\leq\mathcal{O}(T^{1/4})$, the regret bound of the DPMFW algorithm matches the provably optimal $\mathcal{O}(\sqrt{T})$ bound in the non-private setting, i.e., privacy could be guaranteed for free.

Note that although all our proposed algorithms are applied to the multilinear extension of the discrete submodular objective function, we can couple the algorithms with the lossless rounding schemes of [46, 58] for matroid constraints and [128, 129] for knapsack constraints to obtain discrete solutions with similar guarantees for the original submodular set function.

9.3 Differential privacy

In this work, a sensitive dataset D consists of *private* non-negative monotone submodular set functions $F_1, \ldots, F_T : 2^V \to \mathbb{R}_+$. In the offline setting, the non-negative monotone submodular objective function $F_D : 2^V \to \mathbb{R}_+$ depends on $\{F_t\}_{t=1}^T$ and is given in advance. Two datasets D and D' are neighboring ($D \sim D'$) if all but one of the T submodular functions in the datasets are equal. We define the sensitivity of the submodular set function F_D via

$$\Delta = \max_{D':D'\sim D} \max_{S\subseteq V} |F_D(S) - F_{D'}(S)|.$$

For the online setting, at each step $t \in [T]$ (where T is the length of the horizon), the private submodular function F_t arrives after committing to an action S_t leading to a utility $F_t(S_t)$ for the algorithm.

Definition 9.3.1 *[78] For $\epsilon, \delta \in \mathbb{R}_+$, a randomized algorithm $\mathcal{A}$ is called (ϵ, δ)-differentially private if for any two neighboring datasets D and D' and any set of possible outcomes S, the following holds:*

$$\mathbb{P}[\mathcal{A}(D) \in S] \leq \exp(\epsilon)\mathbb{P}[\mathcal{A}(D') \in S] + \delta.$$

If $\delta = 0$, we say that $\mathcal{A}$ is ϵ-differentially private.

Theorem 9.3.1 (Basic composition theorem) *[78] Let $\mathcal{A}_k$ be an (ϵ_k, δ_k)-differentially private algorithm for $k \in [K]$. Then, if algorithm $\mathcal{A}$ is defined to be $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_K)$, $\mathcal{A}$ is $(\sum_{k=1}^K \epsilon_k, \sum_{k=1}^K \delta_k)$-differentially private.*

K-fold adaptive composition. Let $\{(\epsilon_k, \delta_k)\}_{k=1}^K$ be a sequence of privacy parameters and let $\mathcal{A}$ be an algorithm that works as follows on a dataset D: At each round $k \in [K]$, the algorithm chooses an (ϵ_k, δ_k)-differentially private algorithm $\mathcal{A}_k$ and releases the output of $\mathcal{A}_k$, where $\mathcal{A}_k$ depends on the output of the previous algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_{k-1}$ but not on the dataset D itself. The output of $\mathcal{A}$ is called the *K-fold adaptive composition* of (ϵ_k, δ_k)-differentially private algorithms $\mathcal{A}_k$. The following privacy guarantee holds for the composite algorithm $\mathcal{A}$.

Theorem 9.3.2 (Advanced composition theorem) *[78] Given target privacy parameters* $0 < \epsilon' < 1$ *and* $\delta' > 0$*, in order to ensure* $(\epsilon', K\delta + \delta')$ *cumulative privacy loss for the composite algorithm* $\mathcal{A}$*, it suffices that each algorithm is* (ϵ, δ)*-differentially private, where* $\epsilon = \frac{\epsilon'}{2\sqrt{2K\ln(1/\delta')}}$.

9.4 Differentially private offline submodular maximization

In this section, we first introduce the Differentially Private Continuous Greedy (DPCG) algorithm and analyze its approximation and privacy guarantees in Section 9.4.1. Then, in Section 9.4.2, we consider the setting where the submodular objective function has a bounded curvature, and we introduce the κ-DPCG algorithm with improved approximation ratio.

9.4.1 Differentially Private Continuous Greedy (DPCG) algorithm

We propose the Differentially Private Continuous Greedy (DPCG) algorithm in Algorithm 1. The algorithm performs K Frank-Wolfe iterations to obtain $\{v_k\}_{k=1}^K$ and outputs the average $x = \frac{1}{K}\sum_{k=1}^K v_k$. Note that the output is the average of K points in the convex constraint set P (the matroid or knapsack polytope) and hence, $x \in P$ holds. The privacy is ensured by adding noise sampled from the distribution $\mathcal{D}$ to the gradients $\{\nabla f(x^{(k)})\}_{k=1}^K$. We first provide three useful Lemmas below.

Lemma 9.4.1 *If the multilinear extension* f *is* L*-smooth with respect to* $\|\cdot\|_1$ *and the diameter of* P *is denoted by* $R = \max_{x\in P}\|x\|_1$*, Algorithm 1 outputs* $x = x^{(K+1)}$ *such that the following holds:*

$$\mathbb{E}[f(x)] \geq (1 - \frac{1}{e})f(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K},$$

where expectation is taken with respect to the noise distribution $\mathcal{D}$ *and* $G_{\mathcal{D}} := \mathbb{E}_{Y\sim\mathcal{D}}\big[\max_{x\in P}\langle Y, x\rangle - \min_{x\in P}\langle Y, x\rangle\big]$ *measures the width of* P *under* $\mathcal{D}$.

Algorithm 1 Differentially Private Continuous Greedy (DPCG) algorithm

```
**Input:** $K$, the constraint set $P$, the multilinear extension $f : [0,1]^{|V|} \to \mathbb{R}$ of the monotone submodular set function $F : 2^V \to \mathbb{R}$, and the noise distribution $\mathcal{D}$.
**Initialization:** $x^{(1)} = 0$.
**for** $k = 1, 2, \ldots, K$ **do**
    Draw $Y^{(k)} \sim \mathcal{D}$.
    Set $v_k = \arg\max_{v \in P} \langle v, \nabla f(x^{(k)}) + Y^{(k)} \rangle$.
    Set $x^{(k+1)} = x^{(k)} + \frac{1}{K} v_k$.
**end for**
**Output:** $x = x^{(K+1)}$.
```

Proof For $k \in [K]$, we can write:

$$\begin{aligned} f(x^{(k+1)}) - f(x^{(k)}) &\overset{(a)}{\geq} \frac{1}{K}\langle v_k, \nabla f(x^{(k)})\rangle - \frac{L}{2K^2}\|v_k\|_1^2 \\ &\overset{(b)}{\geq} \frac{1}{K}\langle x^*, \nabla f(x^{(k)})\rangle + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\ &\overset{(c)}{\geq} \frac{1}{K}\langle (x^* - x^{(k)}) \vee 0, \nabla f(x^{(k)})\rangle + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\ &\overset{(d)}{\geq} \frac{1}{K}\big(f(x^* \vee x^{(k)}) - f(x^{(k)})\big) + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\ &\overset{(e)}{\geq} \frac{1}{K}\big(f(x^*) - f(x^{(k)})\big) + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2}, \end{aligned}$$

where (a) is due to the L-smoothness of f, (b) follows from the update rule of the algorithm, (c) and (e) use the monotonicity of f, and (d) exploits concavity of f along non-negative directions. Taking the expectation of the above inequality and rearranging the terms, we have:

$$\mathbb{E}[f(x^{(k+1)})] - f(x^*) \geq (1 - \frac{1}{K})\big(\mathbb{E}[f(x^{(k)})] - f(x^*)\big) - \frac{1}{K}G_{\mathcal{D}} - \frac{LR^2}{2K^2}.$$

Applying the inequality recursively for all $k \in [K]$, we obtain:

$$\mathbb{E}[f(x^{(K+1)})] - f(x^*) \geq (1 - \frac{1}{K})^K\big(\mathbb{E}[f(\underbrace{x^{(1)}}_{=0})] - f(x^*)\big) - G_{\mathcal{D}} - \frac{LR^2}{2K}.$$

Rearranging the terms and using the inequality $(1-\frac{1}{K})^K \leq \frac{1}{e}$, we obtain the desired result. ■

Lemma 9.4.2 *L-smoothness parameter of the multilinear extension* f *of the submodular set function* $F : 2^V \to \mathbb{R}$ *is bounded as* $L \leq m_F$, *where* $m_F = \max_{i \in V} F(\{i\})$. *Moreover, for submodular maximization over the matroid polytope* $P(\mathcal{M})$, $R \leq \text{rank}(\mathcal{M})$ *holds, and for submodular maximization subject to a knapsack constraint, we have* $R \leq \frac{1}{c_{\min}}$.

Assume that the submodular objective function is Δ-sensitive (defined in Section 9.3). The following lemma provides the performance guarantees of the DPCG algorithm under Laplace noise distribution.

Lemma 9.4.3 *If* $\mathcal{D} = \text{Lap}^{|V|}(\lambda)$, *where* $\text{Lap}^{|V|}(\lambda)$ *denotes a distribution over* $\mathbb{R}^{|V|}$ *such that each coordinate is drawn i.i.d. from the Laplace distribution with p.d.f.* $f(z|\lambda) = \frac{1}{2\lambda}\exp(-\frac{|z|}{\lambda})$, $\forall z \in \mathbb{R}$, *setting* $\lambda = \frac{2K|V|\Delta}{\epsilon}$, *the following holds in expectation:*

$$\mathbb{E}[f(x)] \geq (1 - \frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{RK|V|\ln(|V|)\Delta}{\epsilon}),$$

where $R = \text{rank}(\mathcal{M})$ *and* $R = \frac{1}{c_{\min}}$ *for matroid and knapsack constraints respectively. Also, with probability at least* $1 - \frac{1}{K}$, *we have:*

$$f(x) \geq (1 - \frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{RK|V|\ln(K|V|)\Delta}{\epsilon}).$$

Moreover, Algorithm 1 preserves ϵ*-differential privacy.*

Proof In order to analyze the differential privacy of the proposed algorithm, let f_D and $f_{D'}$ be the multilinear extension of monotone submodular set functions F_D and $F_{D'}$ associated with neighboring datasets D and D'. Using the definition of multilinear extension, we can write:

$$
\begin{aligned}
\|\nabla f_D(x) - \nabla f_{D'}(x)\|_1 &= \sum_{i=1}^{|V|} |\nabla_i f_D(x) - \nabla_i f_{D'}(x)| \\
&= \sum_{v\in V} |\mathbb{E}_{S\sim x}\big[F_D(S\cup\{v\}) - F_D(S\setminus\{v\}) - F_{D'}(S\cup\{v\}) + F_{D'}(S\setminus\{v\})\big]| \\
&\leq \sum_{v\in V} \mathbb{E}_{S\sim x}\big[|F_D(S\cup\{v\}) - F_{D'}(S\cup\{v\})| + |F_{D'}(S\setminus\{v\}) - F_D(S\setminus\{v\})|\big] \\
&\leq 2|V|\Delta.
\end{aligned}
$$

Let $\lambda = \frac{2K|V|\Delta}{\epsilon}$. We show that for each $k \in [K]$, $\nabla f(x^{(k)}) + Y^{(k)}$ is $(\frac{\epsilon}{K})$-differentially private. Considering the immunity of differential privacy to post-processing and using the basic composition theorem, we can conclude that the proposed algorithm is ϵ-differentially private. We have:

$$
\begin{aligned}
\frac{\mathbb{P}(\nabla f_D(x) + Y^{(k)} = z)}{\mathbb{P}(\nabla f_{D'}(x) + Y^{(k)} = z)} &= \prod_{i=1}^{|V|} \frac{\exp\big(-\frac{\epsilon|z_i - \nabla_i f_D(x)|}{2K|V|\Delta}\big)}{\exp\big(-\frac{\epsilon|z_i - \nabla_i f_{D'}(x)|}{2K|V|\Delta}\big)} \\
&= \prod_{i=1}^{|V|} \exp\big(\frac{\epsilon(|z_i - \nabla_i f_{D'}(x)| - |z_i - \nabla_i f_D(x)|)}{2K|V|\Delta}\big) \\
&\leq \exp\big(\frac{\epsilon\|\nabla f_{D'}(x) - \nabla f_D(x)\|_1}{2K|V|\Delta}\big) \\
&\leq \exp(\frac{\epsilon}{K}).
\end{aligned}
$$

Hence, $\nabla f(x^{(k)}) + Y^{(k)}$ is $(\frac{\epsilon}{K})$-differentially private. We now find an upper bound for $G_{\mathcal{D}}$. Using the definition of the matroid polytope $P(\mathcal{M})$, for all $x \in P(\mathcal{M})$, we have $\|x\|_1 \leq \text{rank}(\mathcal{M})$, where $\text{rank}(\mathcal{M})$ is the rank of the matroid constraint. Therefore, we have:

$$
\begin{aligned}
\max_{x\in P(\mathcal{M})} \langle Y, x\rangle - \min_{x\in P(\mathcal{M})} \langle Y, x\rangle &\leq \|x\|_1\|Y\|_\infty + \|x\|_1\|Y\|_\infty \\
&= 2\|x\|_1\|Y\|_\infty \\
&\leq 2\text{rank}(\mathcal{M})\|Y\|_\infty.
\end{aligned}
$$

Thus, we have $G_{\mathcal{D}} \leq 2\text{rank}(\mathcal{M})\mathbb{E}_{Y\sim\text{Lap}^{|V|}(\lambda)}\|Y\|_\infty$. Note that in the case of knapsack constraint $c^T x \leq 1$, we have $c_{\min}\|x\|_1 \leq c^T x \leq 1$ and thus, $\|x\|_1 \leq \frac{1}{c_{\min}}$ holds. Therefore, we have $G_{\mathcal{D}} \leq \frac{2}{c_{\min}} \mathbb{E}_{Y \sim \mathrm{Lap}^{|V|}(\lambda)} \|Y\|_{\infty}$ under the knapsack constraint. For the Laplace random vector $Y \sim \mathrm{Lap}^{|V|}(\lambda)$, we have:

$$\mathbb{E}\|Y\|_{\infty} \leq \mathcal{O}\big(\lambda \ln(|V|)\big),$$
$$\mathbb{P}\big(\|Y\|_{\infty} \leq \sqrt{10}\lambda \ln(K|V|)\big) \geq 1 - \frac{1}{K^2}.$$

Therefore, using the union bound over $k \in [K]$, we can obtain the result as stated. ■

Alternatively, we can use the Gaussian noise to ensure differential privacy. The analysis for this case is provided in the Appendix. Compared to the Laplace noise, the additive factor in the approximation guarantee using the Gaussian noise is smaller by an order of $\sqrt{|V| \ln(|V|)}$. However, this improved accuracy comes at the price of achieving (ϵ, δ)-differential privacy as opposed to ϵ-differential privacy using the Laplace noise.

We can also use the advanced composition theorem to ensure differential privacy. In particular, for $0 < \epsilon < 1$ and $\delta > 0$, if $\nabla f(x^{(k)}) + Y^{(k)},\ \forall k \in [K]$ is $(\frac{\epsilon}{2\sqrt{2K \ln(1/\delta)}})$-differentially private using the Laplace noise, we can use the advanced composition theorem for all $k \in [K]$ and ensure (ϵ, δ)-differential privacy of Algorithm 1. The result is summarized in the lemma below.

Lemma 9.4.4 *If* $\mathcal{D} = \mathrm{Lap}^{|V|}(\lambda)$*, setting* $\lambda = \frac{4\sqrt{2K \ln(1/\delta)}|V|\Delta}{\epsilon}$*, the following holds in expectation:*

$$\mathbb{E}[f(x)] \geq (1 - \frac{1}{e}) f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{R\sqrt{K \ln(1/\delta)}|V| \ln(|V|)\Delta}{\epsilon}).$$

where $R = \mathrm{rank}(\mathcal{M})$ *and* $R = \frac{1}{c_{\min}}$ *for matroid and knapsack constraints respectively. Also, with probability at least* $1 - \frac{1}{K}$*, we have:*

$$f(x) \geq (1 - \frac{1}{e}) f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{R\sqrt{K \ln(1/\delta)}|V| \ln(K|V|)\Delta}{\epsilon}).$$

Moreover, Algorithm 1 preserves (ϵ, δ)*-differential privacy.*

Combining the result of Lemma 9.4.2 and 9.4.3, we provide the approximation and privacy guarantee of Algorithm 1 below.

Theorem 9.4.1 *Setting* $K = \mathcal{O}(\sqrt{\frac{\epsilon\ \mathrm{rank}(\mathcal{M})}{|V|\ln(|V|)\Delta}})$*, Algorithm 1 is* ϵ*-differentially private and has the following approximation guarantees for matroid and knapsack constraints respectively:*

$$\mathbb{E}[f(x)] \geq (1-\frac{1}{e})f(x^*) - \mathcal{O}(\sqrt{\frac{\Delta(\mathrm{rank}(\mathcal{M}))^3|V|\ln(|V|)}{\epsilon}}),$$
$$\mathbb{E}[f(x)] \geq (1-\frac{1}{e})f(x^*) - \mathcal{O}(\sqrt{\frac{\Delta|V|\ln(|V|)}{(c_{\min})^3\epsilon}}).$$

9.4.2 κ-Differentially Private Continuous Greedy (κ-DPCG) algorithm

For a general monotone continuous function $f : \mathbb{R}^m \to \mathbb{R}$, we define:

$$c_f = 1 - \min_{i\in[m]}\min_{x,y}\frac{\nabla_i f(y)}{\nabla_i f(x)}.$$

Note that $c_f \leq 1$ due to monotonicity of f and $c_f \geq 0$ by setting $x = y$ in the definition. If $f : [0,1]^V \to \mathbb{R}_+$ is the multilinear extension of a normalized monotone submodular set function $F : 2^V \to \mathbb{R}_+$, we can write:

$$\begin{aligned} c_f &= 1 - \min_{i\in V}\min_{x,y}\frac{\nabla_i f(y)}{\nabla_i f(x)} \\ &\overset{(a)}{=} 1 - \min_{i\in V}\min_{y}\frac{\nabla_i f(y)}{\nabla_i f(0)} \\ &\overset{(b)}{=} 1 - \min_{i\in V}\min_{y}\frac{\mathbb{E}_{S\sim y}[F(S\cup\{i\}) - F(S\setminus\{i\})]}{F(\{i\}) - F(\{\})} \\ &\overset{(c)}{=} 1 - \min_{i\in V}\frac{F(V) - F(V\setminus\{i\})}{F(\{i\}) - F(\{\})} \\ &= \kappa_F, \end{aligned}$$

where κ_F is the total curvature of the submodular set function F. (a) follows from the DR-submodularity of f, (b) uses the definition of the multilinear extension and (c) is due to the submodularity of F. Therefore, the parameter c_f extends the notion of curvature from submodular set functions to general monotone continuous functions and could be of independent interest. We propose the κ-DPCG algorithm in Algorithm 2. Let $\lambda = \ell^T x^*$ where $\ell_i = \min_x \nabla_i f(x) = F(V) - F(V\setminus\{i\})$ and $x^* = \mathbf{1}_{S^*}$ be the optimal point corresponding to the optimal solution $S^* \subseteq V$. Compared to the DPCG algorithm, the κ-DPCG algorithm is different in two important respects:

Algorithm 2 κ-Differentially Private Continuous Greedy (κ-DPCG) algorithm

```
**Input:** $K$, $\lambda > 0$, the constraint set $P$, the multilinear extension $f : [0,1]^{|V|} \to \mathbb{R}$ of the monotone submodular set function $F : 2^V \to \mathbb{R}$, and the noise distribution $\mathcal{D}$.
**Initialization:** $x^{(1)} = 0$.
**for** $k = 1, 2, \ldots, K$ **do**
    Draw $Y^{(k)} \sim \mathcal{D}$.
    Set $v_k = \arg\max_{v \in P : \ell^T v \geq \lambda} \langle v, \nabla g(x^{(k)}) + Y^{(k)} \rangle$.
    Set $x^{(k+1)} = x^{(k)} + \frac{1}{K} v_k$.
**end for**
**Output:** $x = x^{(K+1)}$.
```

1. The DPCG algorithm is applied to the function $g(x) = f(x) - \ell^T x$. Note that the function g is normalized monotone DR-submodular as well.

2. The linear maximization step is performed over the intersection of the constraint set P and $\{x : \ell^T x \geq \lambda\}$.

Similar to Lemma 9.4.1, We provide the approximation guarantee of the κ-DPCG algorithm below.

Lemma 9.4.5 *If the multilinear extension f is L-smooth with respect to $\|\cdot\|_1$ and the diameter of P is denoted by $R = \max_{x \in P} \|x\|_1$, Algorithm 2 outputs $x = x^{(K+1)}$ such that the following holds:*

$$\mathbb{E}[f(x)] \geq (1 - \frac{\kappa_F}{e}) f(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K},$$

where $G_{\mathcal{D}} := \mathbb{E}_{Y \sim \mathcal{D}}\big[\max_{x \in P} \langle Y, x \rangle - \min_{x \in P} \langle Y, x \rangle\big]$ and κ_F is the total curvature of the submodular objective function F.

Therefore, all the analysis from Section 9.4.1 can be performed here as well and thus, the κ-DPCG algorithm maintains the same privacy and utility guarantees as the DPCG algorithm except for its improved approximation ratio $1 - \frac{\kappa_F}{e}$.

Note that although we have used $\lambda = \ell^T x^*$ in Algorithm 2, the optimal value x^* is generally unknown and therefore, we have to guess the value of λ in practice. Using the definition of ℓ and submodularity of F, we have $\ell_i \leq m_F \; \forall i \in V$. Therefore, $\lambda \leq |V| m_F$ holds. We discretize the interval $[0, m_F]$ with $\mathcal{O}(\frac{1}{\gamma})$ points of the form $i\gamma m_F$ for $0 \leq i \leq \frac{1}{\gamma}$, along with $\mathcal{O}(\frac{1}{\gamma}|V| \ln(|V|))$ points of the form $(1 + \frac{\gamma}{|V|})^i m_F$ for $0 \leq i \leq \log_{1+\frac{\gamma}{|V|}} |V|$ to fill the interval $[m_F, |V| m_F]$. We then run Algorithm 2 using each point as a guess for λ and we output the best solution found, i.e., the solution with the highest utility. If $\lambda \in [0, m_F]$, we should have $\lambda \geq \hat{\lambda} \geq \lambda - \gamma m_F$ using one of the guesses $\hat{\lambda}$ in the interval $[0, m_F]$. Otherwise, if $\lambda \in [m_F, |V| m_F]$, consider the largest guess $\hat{\lambda}$ in the interval $[m_F, |V| m_F]$ satisfying $\lambda \geq \hat{\lambda}$. We have:

$$\lambda \geq \hat{\lambda} \geq \lambda(1 + \frac{\gamma}{|V|})^{-1} \geq \lambda(1 - \frac{\gamma}{|V|}) \geq \lambda - \gamma m_F,$$

where the last inequality uses $\lambda \leq |V| m_F$. Therefore, in practice, the approximation guarantee of the κ-DPCG algorithm has an additional γm_F error term which can be tuned by choosing γ.

The κ-DPCG algorithm is inspired by the non-private algorithm of [197] that achieves the optimal $1 - \frac{\kappa}{e}$ approximation ratio for offline submodular maximization. To avoid the issue of estimating the optimal value x^*, we can adapt the $(1 - \frac{\kappa}{e})$-approximate non-private algorithm of [83] instead to design our private algorithm.

9.5 Differentially private online submodular maximization

In this section, we study the following general protocol of online submodular maximization in the context of differential privacy: There is a fixed constraint set $(V, \mathcal{I})$ which could either be a matroid or a knapsack constraint. At each iteration $t \in [T]$, the online algorithm chooses $S_t \in \mathcal{I}$. Upon committing to this choice, a normalized monotone submodular set function F_t is revealed and the algorithm receives the payoff $F_t(S_t)$. The goal is to minimize the $(1 - \frac{1}{e})$-regret defined below:

$$R_T = (1 - \frac{1}{e}) \max_{S \in \mathcal{I}} \sum_{t=1}^{T} F_t(S) - \sum_{t=1}^{T} F_t(S_t),$$

Algorithm 3 Differentially Private Meta Frank-Wolfe (DPMFW) algorithm

```
**Input:** $K$, the constraint set $P$.
**Output:** $\{x_t\}_{t=1}^T$
Initialize $K$ instances $\{\mathcal{E}_k\}_{k=1}^K$ of the $(\frac{\epsilon}{2\sqrt{2K\ln(1/\delta)}})$-differentially private algorithm of [3] with noise distribution $\mathcal{D}$ and regularizer $g(x) = \sum_{i=1}^{|V|} x_i \ln(x_i)$ for online linear optimization over $P$.
**for** $t = 1, \dots, T$ **do**
  $x_t^{(1)} = 0$.
  **for** $k = 1, 2, \dots, K$ **do**
    Let $v_t^{(k)}$ be the output of $\mathcal{E}_k$ for round $t$.
    Set $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  **end for**
  Play $x_t = x_t^{(K+1)}$.
  **for** $k = 1, 2, \dots, K$ **do**
    Feedback $\nabla f_t(x_t^{(k)})$ to $\mathcal{E}_k$ as the linear utility vector observed at round $t$.
  **end for**
**end for**
```

where $1-\frac{1}{e}$ is the optimal polynomial time approximation ratio for offline monotone submodular maximization subject to matroid or knapsack constraints. We propose the Differentially Private Meta Frank-Wolfe (DPMFW) algorithm for online submodular maximization which exploits Algorithm 1 of [3] for differentially private online linear optimization as a subroutine. The algorithm is presented in Algorithm 3. The DPMFW algorithm is applied to the multilinear extensions $\{f_t\}_{t=1}^T$ of the discrete monotone submodular utility functions $\{F_t\}_{t=1}^T$. At round $t \in [T]$, similar to the DPCG algorithm, the DPMFW algorithm outputs the average of K points $\{v_t^{(k)}\}_{k=1}^K$ in the constraint set and hence, $x_t \in P$ holds. However, since the utility function F_t remains unknown until the algorithm commits to a choice S_t, we instead run K instances $\{\mathcal{E}_k\}_{k=1}^K$ of the differentially private online linear optimization algorithm of [3] to mimic the K Frank-Wolfe updates of the DPCG algorithm. $\{\mathcal{E}_k\}_{k=1}^K$ combine the well-known Follow the Regularized Leader (FTRL) algorithm for online linear optimization with the Tree-Based Aggregation Protocol (TBAP) of [77, 112] for maintaining differentially private partial sums of linear utility vectors arriving online. See the Appendix for a more detailed presentation of Algorithm 1 of [3].

We provide the regret bound and privacy guarantee of the DPMFW algorithm below.

Theorem 9.5.1 *Let* $0 < \epsilon < 1$ *and* $\delta > 0$*. If* $\mathcal{D} = \mathrm{Lap}^{|V|}(\lambda)$*, where* $\mathrm{Lap}^{|V|}(\lambda)$ *is a distribution over* $\mathbb{R}^{|V|}$ *such that each coordinate is drawn i.i.d. from the Laplace distribution with p.d.f.* $f(z|\lambda) = \frac{1}{2\lambda}\exp(-\frac{|z|}{\lambda})\ \forall z \in \mathbb{R}$*, setting* $\lambda = \frac{2m_F|V|\ln T\sqrt{2K\ln(1/\delta)}}{\epsilon}$ *and* $K = \mathcal{O}(\sqrt{T})$*, Algorithm 3 is* (ϵ, δ)*-differentially private and has the following expected regret bound for matroid and knapsack constraints respectively:*

$$\mathbb{E}[R_T] \leq \mathcal{O}(\mathrm{rank}(\mathcal{M})\sqrt{T\ln|V|}) + \tilde{\mathcal{O}}(\frac{(\mathrm{rank}(\mathcal{M}))^{3/2}|V|T^{1/4}\sqrt{\ln(1/\delta)}}{\epsilon}),$$

$$\mathbb{E}[R_T] \leq \mathcal{O}(\frac{\sqrt{T\ln|V|}}{c_{\min}}) + \tilde{\mathcal{O}}(\frac{|V|T^{1/4}\sqrt{\ln(1/\delta)}}{(c_{\min})^{3/2}\epsilon}).$$

The above theorem shows that if $\frac{\sqrt{\ln(1/\delta)}}{\epsilon} \leq \mathcal{O}(T^{1/4})$ holds, we can obtain a regret bound of $\mathcal{O}(\sqrt{T})$ which matches the optimal regret bound for online submodular maximization in the non-private setting.

We can alternatively use the Gaussian noise as the noise distribution $\mathcal{D}$. The analysis in this setting is provided in the Appendix.

9.6 Conclusion

In this chapter, we studied the differentially private maximization of non-negative monotone submodular set functions, subject to matroid or knapsack constraints, in both offline and online settings. We proposed differentially private algorithms with optimal approximation ratios which are faster (i.e., less query complexity) and have improved accuracy compared to the prior works.

Chapter 10

M-EXPERTS: NO-REGRET ONLINE PREDICTION WITH STRATEGIC EXPERTS AND SUBMODULAR UTILITIES

10.1 Chapter overview

In this chapter, we present joint work with Maryam Fazel that was published at the Conference on Neural Information Processing Systems (NeurIPS) 2023. We study a generalization of the online binary prediction with expert advice framework, called the m-experts problem, where at each round, the learner is allowed to pick $m \geq 1$ experts from a pool of K experts and the overall utility is a modular or submodular function of the chosen experts. We focus on the setting in which experts act strategically and aim to maximize their influence on the algorithm's predictions by potentially misreporting their beliefs about the events. Among others, this setting finds applications in forecasting competitions where the learner seeks not only to make predictions by aggregating different forecasters but also to rank them according to their relative performance. Our goal is to design algorithms that satisfy the following two requirements: 1) *Incentive-compatible*: Incentivize the experts to report their beliefs truthfully, and 2) *No-regret*: Achieve sublinear regret with respect to the true beliefs of the best-fixed set of m experts in hindsight. Prior works have studied this framework when $m = 1$ and provided incentive-compatible no-regret algorithms for the problem. We first show that a simple reduction of our problem to the $m = 1$ setting is neither efficient nor effective. Then, we provide algorithms that utilize the specific structure of the utility functions to achieve the two desired goals. We also study the 1-expert problem in the setting where predictions are performative, i.e., they can influence the state of the world and the binary events. We show that if the influence of the expert's predictions on outcomes is bounded, we can design no-regret algorithms where the regret is defined with respect to the true beliefs of the experts.

10.2 Introduction

Learning from a constant flow of information is one of the most prominent challenges in machine learning. In particular, online learning requires the learner to iteratively make decisions and at the time of making each decision, the outcome associated with it is unknown to the learner. The *experts problem* is perhaps the most well-known problem in online learning [203, 52, 137, 121]. In this problem, the learner aims to make predictions about a sequence of T binary events. To do so, the learner has access to the advice of K experts who each have internal beliefs about the likelihood of each event. At each round $t \in [T]$, the learner has to choose one among the advice of K experts and upon making her choice, the t-th binary event is realized and a loss bounded between zero and one is revealed. The goal of the learner is to have no regret, i.e., to perform as well as the best-fixed expert in hindsight.

In many applications, however, the experts are strategic and wish to be selected by the learner as often as possible. To this end, they may strategically misreport their beliefs about the events. For instance, FiveThirtyEight[1] aggregates different pollsters according to their past performance to make a single prediction for elections and sports matches. To do so, FiveThirtyEight maintains publicly available pollster ratings[2]. A low rating can be harmful to the pollster's credibility and adversely impact their revenue opportunities in the future. Therefore, instead of maximizing their expected performance by reporting their predictions truthfully, the pollsters may decide to take risks and report more extreme beliefs to climb higher on the leaderboard. Therefore, it is important to design algorithms that not only achieve *no-regret* but also motivate the experts to report their true beliefs (*incentive-compatible*). Otherwise, the quality of the learner's predictions may be harmed.

As we have mentioned in Section 10.2.1, all the previous works on this topic have focused on the standard experts problem where the goal is to choose a *single* expert among the K

---

[1] https://fivethirtyeight.com/

[2] https://projects.fivethirtyeight.com/pollster-ratings/

experts. In the offline setting, this is equivalent to a forecasting competition in which only the single highest-ranked forecaster wins and receives prizes. However, in many applications, a *set* of top-performing forecasters are awarded perks and benefits. For instance, in the Good Judgement Project, a recent geopolitical forecasting tournament, the top 2% of forecasters were given the "superforecaster" status and received benefits such as paid conference travel and employment opportunities [198]. Similarly, in the latest edition of Kaggle's annual machine learning competition to predict the match outcomes of the NCAA March Madness college basketball tournament (called "March Machine Learning Mania 2023"[3]), the top 8 forecasters on the leaderboard received monetary prizes.

In this chapter, we initiate the study of the m-experts problem with strategic experts. Variants of the m-experts problem have been previously studied in [205, 124, 67, 156], however, all of these works focused only on providing no-regret algorithms for the problem and the incentive compatibility considerations were not taken into account. To the best of our knowledge, this is the first work that focuses on the strategic m-experts problem where the experts may misreport their true beliefs to increase their chances of being chosen by the learner.

For the setting with modular utilities, perhaps the simplest approach to learning well compared to the best-fixed set of m experts while maintaining incentive compatibility is to run the incentive-compatible WSU algorithm of [89] for the standard experts problem (the setting with $m = 1$) over the set of $\binom{K}{m}$ meta-experts where each meta-expert corresponds to one of the sets of size m. This approach has two major drawbacks: 1) There are exponentially many meta-experts and maintaining weights per each meta-expert and running the WSU algorithm is computationally expensive, and 2) The dependence of the regret bound on m is sub-optimal. Therefore, for our setting, it is preferable to design algorithms that are tailored for the m-experts problem.

---

[3] https://www.kaggle.com/competitions/march-machine-learning-mania-2023/

10.2.1 Related work

Prior works have studied the experts problem under incentive compatibility considerations for two feedback models: In the full information setting, the learner observes the reported prediction of all experts at each round. In the partial information setting, however, the learner is restricted to choosing a single expert at each round and does not observe the prediction of other experts. [178] considered algorithms that maintain weights over the experts and choose experts according to these weights. They assumed that experts' incentives are only affected by the unnormalized weights of the algorithm over the experts. However, since the probability of an expert being chosen by the learner equals her normalized weight, the aforementioned assumption might not be suitable. Later on, [89] assumed that at each round $t \in [T]$, incentives are tied to the expert's normalized weight (i.e., the probability of being chosen at round $t+1$) and studied this problem under both feedback models. They proposed the WSU and WSU-UX algorithms for the full information and partial information settings respectively where both algorithms are incentive-compatible and they obtained $\mathcal{O}(\sqrt{T\text{ln}K})$ and $\mathcal{O}(T^{2/3}(K\text{ln}K)^{1/3})$ regret bounds for the algorithms. Then, [90] considered *non-myopic* strategic experts where the goal of each expert is to maximize a conic combination of the probabilities of being chosen in all subsequent rounds (not just the very next round). They showed that the well-known Follow the Regularized Leader (FTRL) algorithm with the negative entropy regularizer obtains a regret bound of $\mathcal{O}(\sqrt{T \ln K})$ while being $\Theta(\frac{1}{\sqrt{T}})$ approximately incentive-compatible, i.e., it is a strictly dominated strategy for any expert to make reports $\Theta(\frac{1}{\sqrt{T}})$ distant from their true beliefs.

For the non-strategic m-experts problem with modular utilities, [124] proposed the Component Hedge (CH) algorithm and obtained a regret bound of $\sqrt{2m\ell^* \ln(\frac{K}{m})} + m \ln(\frac{K}{m})$ where ℓ^* is the cumulative loss of the best-chosen set in hindsight. They also gave a matching lower bound for this problem. [67] studied the FTPL algorithm with Gaussian noise distribution and provided an $\mathcal{O}(m\sqrt{T \ln(\frac{K}{m})})$ regret bound for this setting. For the non-strategic m-experts problem with submodular utility functions, [101] proposed the online distorted greedy algorithm (with dual averaging algorithm as the algorithm $\mathcal{A}_i$ for $i = 1, \ldots, m$)

whose regret bound is $\mathcal{O}(\sqrt{mT\ln(\frac{K}{m})})$. More recently, [156] studied the m-experts problem under various choices of the utility function (sum-reward, max-reward, pairwise-reward, and monotone reward). In particular, for the setting with modular utilities (sum-reward), they proposed an algorithm that matches the optimal regret bound of the CH algorithm while being computationally more efficient.

10.2.2 Contributions

In this chapter, we focus on a generalization of the experts problem (called "m-experts problem") where at each round, instead of picking a single expert, we are allowed to pick $m \geq 1$ experts and our utility is either a modular or submodular function of the chosen experts. In particular, at round $t \in [T]$ and for a set of experts $S_t \subseteq [K]$, the utility function is defined as $f_t(S_t) = \frac{|S_t|}{m} - \frac{1}{m}\sum_{i \in S_t} \ell_{i,t}$ and $f_t(S_t) = 1 - \prod_{i \in S_t} \ell_{i,t}$ in the modular and submodular cases respectively where $\ell_{i,t} \in [0,1]$ is the loss of expert i at round t. The goal is to design algorithms that perform as well as the best-fixed set of m experts in hindsight (*no-regret*) and incentivize the experts to report their beliefs about the events truthfully (*incentive-compatible*).

Towards this goal, we build upon the study of the Follow the Perturbed Leader (FTPL) algorithm for the m-experts problem with modular utility functions by [67] and derive a sufficient condition for the perturbation distribution to guarantee approximate incentive compatibility. Furthermore, we show how this condition is related to the commonly used bounded hazard rate assumption for noise distribution. In particular, we show that while FTPL with Gaussian perturbations is not incentive-compatible, choosing Laplace or hyperbolic noise distribution guarantees approximate incentive compatibility.

Moreover, inspired by Algorithm 1 of [101] for online monotone submodular maximization subject to a matroid constraint, we first introduce a simpler algorithm (called the "online distorted greedy algorithm") for the special case of cardinality constraints. This algorithm utilizes m incentive-compatible algorithms for the standard experts problem (i.e., $m = 1$ setting) and outputs their combined predictions. We provide $(1 - \frac{c}{e})$-regret bounds for the algorithm where $c \in [0,1]$ is the average curvature of the submodular utility functions. Therefore, applying the algorithm to the setting where the utility functions are modular (i.e., $c = 0$), the approximation ratio is 1. For submodular utility functions, the algorithm achieves the optimal $1 - \frac{c}{e}$ approximation ratio.

We also study the 1-expert problem in the setting where predictions are performative, i.e., they can influence the state of the world and the binary events. We show that if the influence of the expert's predictions on outcomes is bounded, we can design no-regret algorithms where the regret is defined with respect to the true beliefs of the experts.

Finally, we validate our theoretical results through experiments on data gathered from a forecasting competition run by FiveThirtyEight in which forecasters make predictions about the match outcomes of the recent 2022–2023 National Football League (NFL).

10.3 Preliminaries

As mentioned in Section 10.2.2, at round $t \in [T]$ and for a set of experts $S \subseteq [K]$, the submodular utility function is defined as $f_t(S_t) = 1 - \prod_{i \in S_t} \ell_{i,t}$ where $\ell_{i,t} \in [0,1]$ is the loss of expert i at round t. To show that this function is submodular, note that for all $A \subset B \subset [K]$ and $j \in [K] \setminus B$, we have:

$$f(A \cup \{j\}) - f(A) = (1 - \ell_{j,t}) \prod_{i \in A} \ell_{i,t} \geq (1 - \ell_{j,t}) \prod_{i \in B} \ell_{i,t} = f(B \cup \{j\}) - f(B),$$

where the inequality follows from $\ell_{i,t} \in [0,1]$ for $i \in B \setminus A$.

10.4 m-experts problem

We introduce the m-experts problem in this section. In this problem, there are K experts available and each expert makes probabilistic predictions about a sequence of T binary outcomes. At round $t \in [T]$, expert $i \in [K]$ has a private belief $b_{i,t} \in [0,1]$ about the outcome $r_t \in \{0,1\}$, where r_t and $\{b_{i,t}\}_{i=1}^{K}$ are chosen arbitrarily and potentially adversarially. Expert i reports $p_{i,t} \in [0,1]$ as her prediction to the learner. Then, the learner chooses a set S_t containing m of the experts. Upon committing to this action, the outcome r_t is revealed, and expert i incurs a loss of $\ell_{i,t} = \ell(b_{i,t}, r_t)$ where $\ell : [0,1] \times \{0,1\} \to [0,1]$ is a bounded loss function. In this chapter, we focus on the quadratic loss function defined as $\ell(b,r) = (b-r)^2$. The utility of the learner at round t is one of the following:

- **Modular utility function:** $f_t(S_t) = \frac{|S_t|}{m} - \frac{1}{m}\sum_{i\in S_t} \ell_{i,t} = \frac{|S_t|}{m} - \frac{1}{m}\sum_{i\in S_t} \ell(b_{i,t}, r_t)$.
- **Submodular utility function:** $f_t(S_t) = 1 - \prod_{i\in S_t} \ell_{i,t} = 1 - \prod_{i\in S_t} \ell(b_{i,t}, r_t)$.

It is easy to see that f_t is monotone in both cases and $f_t(S_t) \in [0,1]$ holds. Note that the utility at each round is defined with respect to the true beliefs of the chosen experts rather than their reported beliefs.

The goal of the learner is twofold:

1) Minimize the α-regret defined as $\alpha\text{-}R_T = \mathbb{E}\big[\alpha \max_{S\subseteq[K]:|S|=m} \sum_{t=1}^T f_t(S) - \sum_{t=1}^T f_t(S_t)\big]$, where the expectation is taken with respect to the potential randomness of the algorithm. For the modular utility function, $\alpha = 1$ and for the submodular setting, we set $\alpha = 1 - \frac{c_f}{e}$ (where $f = \sum_{t=1}^T f_t$) which is the optimal approximation ratio for any algorithm making polynomially many queries to the objective function.

2) Incentivize experts to report their private beliefs truthfully. To be precise, at each round $t \in [T]$, each expert $i \in [K]$ acts strategically to maximize their probability of being chosen at round $t+1$ and the learner's algorithm is called incentive-compatible if expert i maximizes this probability by reporting $p_{i,t} = b_{i,t}$. To be precise, we define the incentive compatibility property below.

Definition 10.4.1 *An online learning algorithm is incentive-compatible if for every $t \in [T]$, every expert $i \in [K]$ with belief $b_{i,t}$, every report $p_{i,t}$, reports of other experts $p_{-i,t}$, every history of reports $\{p_{j,s}\}_{j\in[K],s<t}$, and outcomes $\{r_s\}_{s<t}$, we have:*

$$\begin{aligned}
&\mathbb{E}_{r_t\sim Bern(b_{i,t})}\big[\pi_{i,t+1} \mid b_{i,t}, p_{-i,t}, \{p_{j,s}\}_{j\in[K],s<t}, \{r_s\}_{s<t}\big] \\
&\geq \mathbb{E}_{r_t\sim Bern(b_{i,t})}\big[\pi_{i,t+1} \mid p_{i,t}, p_{-i,t}, \{p_{j,s}\}_{j\in[K],s<t}, \{r_s\}_{s<t}\big],
\end{aligned}$$

where $Bern(b)$ denotes a Bernoulli distribution with probability of success b and $\pi_{i,t+1}$ is the probability of expert i being chosen at round $t+1$.

In other words, an online learning algorithm is incentive-compatible if it is in the best interest of the experts to report their private beliefs truthfully to maximize the probability of being chosen at the next round.

As mentioned earlier, we focus on the quadratic loss function in this chapter. The quadratic loss function is an instance of proper loss functions [175], i.e., the following holds:

$$\mathbb{E}_{r\sim\text{Bern}(b)}[\ell(p,r)] \geq \mathbb{E}_{r\sim\text{Bern}(b)}[\ell(b,r)] \ \forall p \neq b,$$

i.e., each expert minimizes her expected loss (according to their true belief b) by reporting truthfully.

10.4.1 Motivating applications

There are several interesting motivating applications that could be cast into our framework. We mention two classes of such applications below.

- *Forecasting competitions*: In this problem, there are a set of K forecasters who aim to predict the outcome of sports games or elections (between two candidates). At each round $t \in [T]$, information on the past performance of the two opposing teams or candidates is revealed and forecasters provide a probabilistic prediction (as a value between $[0, 1]$) for which team or candidate will win. The learner can choose up to m forecasters at each round and her utility is simply the average of the utilities of chosen experts.
- *Online paging problem with advice* [139]: There is a library $\{1, \ldots, N\}$ of N distinct files. A cache with limited storage capacity can store at most m files at any time. At each round $t \in [T]$, a user arrives and requests one file. The learner has access to a pool of K experts where each expert $i \in [K]$ observes the user history and makes a probabilistic prediction $p_{i,t} \in [0, 1]^N$ for the next file request (where $1^T p_{i,t} = 1$). For instance, $p_{i,t} = e_j$ if expert i predicts the file $j \in [N]$ where e_j is the j-th standard basis vector. Also, $r_t = e_j$ if the j-th file is requested at round $t \in [T]$. The learner can choose m of these experts at each round and put their predictions in the cache. The learner's prediction for round t is correct if and only if one of the m chosen experts has correctly predicted the file. Thus, the loss of

expert i can be formulated as $\ell_{i,t} = \|p_{i,t} - r_t\|_2^2$ and the utility at round t could be written as $f_t(S_t) = 2 - \prod_{i \in S_t} \ell_{i,t}$ which is exactly our submodular utility function. Note that this is a slight generalization of our framework where instead of binary outcomes, we consider nonbinary (categorical) outcomes. All our results could be easily extended to this setting.

10.4.2 Naive approach

The WSU algorithm of [89] for the standard experts problem is derived by drawing a connection between online learning and wagering mechanisms. The framework of one-shot wagering mechanisms was introduced by [132] and is as follows: There are K experts and each expert $i \in [K]$ holds a belief $b_i \in [0, 1]$ about the likelihood of an event. Expert i reports a probability $p_i \in [0, 1]$ and a wager $\omega_i \geq 0$. A wagering mechanism Γ is a mapping from the reports $p = (p_1, \ldots, p_K)$, wagers $\omega = (\omega_1, \ldots, \omega_K)$ and the realization r of the binary event to the payments $\Gamma_i(p, \omega, r)$ to expert i. It is assumed that $\Gamma_i(p, \omega, r) \geq 0 \ \forall i \in [K]$, i.e., no expert loses more than her wager. A wagering mechanism is called budget-balanced if $\sum_{i=1}^{K} \Gamma_i(p, \omega, r) = \sum_{i=1}^{K} \omega_i$. [132, 131] introduced a class of incentive-compatible budget-balanced wagering mechanisms called the Weighted Score Wagering Mechanisms (WSWMs) which is defined as follows: For a fixed proper loss function ℓ bounded in $[0, 1]$, the payment to expert i is

$$\Gamma_i(p, \omega, r) = \omega_i \big(1 - \ell(p_i, r) + \sum_{j=1}^{K} \omega_j \ell(p_j, r)\big).$$

The proposed algorithm in [89] is called Weighted-Score Update (WSU) and the update rule for the weights of the experts $\{\pi_t\}_{t=1}^{T}$ is the following:

$$\pi_{i,t+1} = \eta \Gamma_i(p_t, \pi_t, r_t) + (1 - \eta)\pi_{i,t},$$

where $\pi_{i,1} = \frac{1}{K} \ \forall i \in [K]$. In other words, the normalized weights of the experts at round t are interpreted as the wager of the corresponding expert, and the normalized weights at round $t + 1$ are derived using a convex combination of the weights at the previous round and the payments in WSWM. Note that since WSWM is budget-balanced, the derived weights at each round automatically sum to one and there is no need to normalize the weights (which might break the incentive compatibility). Also, considering the incentive compatibility of WSWM, the WSU algorithm is incentive-compatible as well.

The update rule of WSU could be rewritten as follows:

$$\pi_{i,t+1} = \eta\pi_{i,t}\big(1 - \ell_{i,t} + \sum_{j=1}^{K} \pi_{j,t}\ell_{j,t}\big) + (1-\eta)\pi_{i,t} = \pi_{i,t}(1 - \eta L_{i,t}),$$

where $L_{i,t} = \ell_{i,t} - \sum_{j=1}^{K} \pi_{j,t}\ell_{j,t}$. Therefore, the WSU update rule is similar to that of the Multiplicative Weights Update (MWU) algorithm [9] with the relative loss $L_{i,t}$ instead of $\ell_{i,t}$ in the formula.

A simple approach to solving the m-experts problem with modular utilities ($\ell_{S,t} = \frac{1}{m}\sum_{j \in S} \ell_{j,t}$) is to define an "expert" for each of the possible $\binom{K}{m}$ sets of size m and apply the incentive-compatible WSU algorithm of [89] for the standard experts problem to this setting. Note that we still define incentive compatibility with respect to individual experts (instead of the $\binom{K}{m}$ meta-experts). To be precise, we define $\pi_{i,t} = \sum_{S:|S|=m, i \in S} \pi_{S,t}$. We can show the following:

$$\pi_{i,t+1} = \sum_{S:|S|=m, i\in S} \pi_{S,t+1} = \pi_{i,t}(1 - \frac{\eta}{m}L_{i,t}) - \frac{\eta}{m}\sum_{s \neq i}\Big(\sum_{S:|S|=m, \{i,s\}\subseteq S} \pi_{S,t}\Big)\ell_{s,t}.$$

Given that $\pi_{i,t+1}$ is linear in $L_{i,t}$, $L_{i,t} = \ell_{i,t} - \sum_{j=1}^{K} \pi_{j,t}\ell_{j,t}$ is linear in $\ell_{i,t}$, and the loss function is proper, we can conclude that incentive compatibility holds in this setting as well. We summarize the result of this approach in the theorem below.

Theorem 10.4.1 *If we apply the WSU algorithm of [89] to a standard experts problem with* $\binom{K}{m}$ *experts corresponding to each* S *with* $|S| = m$*, and set* $\eta = \sqrt{\frac{m \ln(\frac{Ke}{m})}{T}}$*, the algorithm is incentive-compatible and its regret is bounded as follows:*

$$\mathbb{E}[1\text{-}R_T] \leq \mathcal{O}(\sqrt{mT\ln(\frac{K}{m})}).$$

This approach has two major drawbacks:

1) Computational complexity of maintaining weights for each $\binom{K}{m}$ feasible sets. In particular, we have to do exponentially many queries to the objective function at each round to update

these weights.

2) The regret bound has a $\sqrt{m}$ dependence on the number of experts m that is suboptimal. In the subsequent sections, we propose two efficient algorithmic frameworks that exploit the modular or submodular structure of the utility function and obtain the desired regret and incentive compatibility guarantees.

10.5 Follow the Perturbed Leader (FTPL) algorithm

In this section, we study the well-known Follow the Perturbed Leader (FTPL) algorithm for the m-experts problem with modular utility functions and study its regret and incentive compatibility guarantees. The algorithm is as follows: At each round $t \in [T]$, we first take K i.i.d. samples $\{\gamma_{i,t}\}_{i=1}^{K}$ from the noise distribution $\mathcal{D}$. In particular, we focus on zero-mean symmetric noise distributions from the exponential family, i.e., $f(\gamma_{i,t}) \propto \exp(-\nu(\gamma_{i,t}))$ where $\nu : \mathbb{R} \to \mathbb{R}_+$ is symmetric about the origin. At round t, we simply keep track of $\sum_{s=1}^{t-1} \ell_{i,s} + \eta\gamma_{i,t}$ for each i (where η is the step size) and pick the m experts for whom this quantity is the smallest. [67] previously studied the FTPL algorithm for a class of problems that includes the m-experts problem. However, they only focused on the setting with zero-mean Gaussian perturbations. In contrast, we not only extend this analysis to all zero-mean symmetric noise distributions from the exponential family, but we also analyze the incentive compatibility guarantees of the algorithm and determine a sufficient condition for the perturbation distribution under which the algorithm is approximately incentive-compatible. This condition is provided below.

Condition 10.5.1 *For all $z \in \mathbb{R}$, $|\nu'(z)| \leq B$ holds for some constant $B > 0$.*

We have $\nu(z) = |z|$ for Laplace distribution. Therefore, $\nu'(z) = \text{sign}(z)$ and $B = 1$. For symmetric hyperbolic distribution, $\nu(z) = \sqrt{1+z^2}$ holds. So, $|\nu'(z)| = \frac{|z|}{\sqrt{1+z^2}} \leq 1$ and $B = 1$.

Condition 10.5.1 is closely related to a boundedness assumption on the hazard rate of the perturbation distribution. We first define the hazard rate below.

Definition 10.5.1 *The hazard rate of* $\mathcal{D}$ *at* $z \in \mathbb{R}$ *is defined as*

$$haz_{\mathcal{D}}(z) = \frac{f_{\mathcal{D}}(z)}{1 - F_{\mathcal{D}}(z)},$$

where $f_{\mathcal{D}}$ *and* $F_{\mathcal{D}}$ *are the probability density function (pdf) and the cumulative density function (cdf) of the noise distribution* $\mathcal{D}$*. The maximum hazard rate of* $\mathcal{D}$ *is* $haz_{\mathcal{D}} = \sup_{z \in \mathbb{R}} haz_{\mathcal{D}}(z)$*.*

The hazard rate is a statistical tool used in survival analysis that measures how fast the tail of a distribution decays. In the theorem below, we show the connection between Condition 10.5.1 and the bounded hazard rate assumption.

Theorem 10.5.1 *If Condition 10.5.1 holds for the perturbation distribution* $\mathcal{D}$ *with the constant* $B > 0$*, we have* $haz_{\mathcal{D}} \leq B$*.*

However, there are distributions with bounded hazard rates for which $\max_z |\nu'(z)|$ is unbounded (i.e., Condition 10.5.1 does not hold). For instance, consider the standard Gumbel distribution. In this case, $\nu(z) = z + \exp(-z)$. Therefore, we have $\nu'(z) = 1 - \exp(-z)$. So, if $z \to -\infty$, $|\nu'(z)| \to \infty$. Therefore, Condition 10.5.1 is strictly stronger than the bounded hazard rate assumption for the noise distribution $\mathcal{D}$.

We show how Condition 10.5.1 guarantees an approximate notion of incentive compatibility for FTPL.

Theorem 10.5.2 *For the FTPL algorithm with a noise distribution satisfying Condition 10.5.1 with a constant* $B > 0$*, at round* $t \in [T]$*, for an expert* $i \in [K]$*, the optimal report from the expert's perspective* $p^*_{i,t}$ *is at most* $\frac{2B}{\eta - 2B}$ *away from her belief* $b_{i,t}$*, i.e., the following holds:*

$$|p^*_{i,t} - b_{i,t}| \leq \frac{2B}{\eta - 2B}.$$

Note that while we focused on the incentive structure in which at each round $t \in [T]$, experts wish to maximize their probability of being chosen at round $t + 1$, the same argument could be applied to a more general setting where the goal is to maximize a conic combination of probabilities of being chosen at all subsequent round $s > t$. Therefore, FTPL is approximately incentive compatible with respect to this more general incentive structure as well. Theorem 10.5.2 allows us to bound the regret of the FTPL algorithm with respect to the true beliefs of the experts. First, note that FTPL obtains the following bound with respect to the reported beliefs of the experts.

Theorem 10.5.3 *For the FTPL algorithm with noise distribution $\mathcal{D}$ satisfying Condition 10.5.1 with the constant $B > 0$, if we set $\eta = \sqrt{\frac{BT}{\ln(\frac{K}{m})}}$, the following holds:*

$$\mathbb{E}[\frac{1}{m}\sum_{t=1}^{T}\sum_{i\in S_t}\ell(p_{i,t}, r_t) - \min_{S:|S|=m}\frac{1}{m}\sum_{t=1}^{T}\sum_{j\in S}\ell(p_{j,t}, r_t)] \leq \mathcal{O}(\sqrt{BT\ln(\frac{K}{m})}).$$

Using the result of Theorem 10.5.2, we have $|p_{i,t} - b_{i,t}| = |p^*_{i,t} - b_{i,t}| \leq \frac{2B}{\eta-2B}$. Moreover, one can easily show that the quadratic loss function is 2-Lipschitz. Therefore, for all $t \in [T]$ and $i \in [K]$, we have:

$$|\ell(p_{i,t}, r_t) - \ell(b_{i,t}, r_t)| \leq \frac{4B}{\eta - 2B}.$$

Putting the above results together, we can obtain the following regret bound for the FTPL algorithm.

$$\mathbb{E}[1\text{-}R_T] = \mathbb{E}[\frac{1}{m}\sum_{t=1}^{T}\sum_{i\in S_t}\ell(b_{i,t}, r_t) - \min_{S:|S|=m}\frac{1}{m}\sum_{t=1}^{T}\sum_{j\in S}\ell(b_{j,t}, r_t)] \leq \mathcal{O}(\sqrt{BT\ln(\frac{K}{m})}) + \frac{8BT}{\eta - 2B}.$$

Given that $\eta = \sqrt{\frac{BT}{\ln(\frac{K}{m})}}$ in Theorem 10.5.3, the expected regret bound is $\mathcal{O}(\sqrt{BT\ln(\frac{K}{m})})$. This result is summarized in the following theorem.

Theorem 10.5.4 *For the FTPL algorithm with noise distribution $\mathcal{D}$ satisfying Condition 10.5.1 with the constant $B > 0$, if we set $\eta = \sqrt{\frac{BT}{\ln(\frac{K}{m})}}$, the regret bound is $\mathcal{O}(\sqrt{BT\ln(\frac{K}{m})})$.*

In order to ensure approximate incentive compatibility, the probability density function of the noise distribution f needs to be such that $\frac{f(z)}{f(z+1)}$ does not grow to infinity for very large z. One way to enforce this condition is via a Lipschitzness assumption on $\ln f$. Condition 10.5.1 implies that $\ln f$ is B-Lipschitz. That is why smaller values of B lead to better approximate incentive compatibility which in turn results in smaller regret bounds (given that the term $4TC$ appears in the regret bound where C is the bound on the approximate incentive-compatibility derived in Theorem 10.5.2).

We can use the FTPL algorithm to obtain results for the partial information setting as well. [2] showed that if the hazard rate of the noise distribution is bounded by B, applying the FTPL algorithm to the partial information setting for the 1-expert problem leads to $\mathcal{O}(\sqrt{BKT \ln K})$ regret bounds. Using the result of Theorem 10.5.1, we know that if Condition 10.5.1 holds, the hazard rate is bounded. Therefore, if the noise distribution satisfies Condition 10.5.1, FTPL applied to the m-experts problem is approximately incentive-compatible and achieves $\mathcal{O}(\sqrt{BKT \ln(\frac{K}{m})})$ regret bound.

10.6 Online distorted greedy algorithm

In this section, we study the setting where the utility function is submodular. In this case, we have $f_t(S_t) = 1 - \prod_{i \in S_t} \ell_{i,t} = 1 - \prod_{i \in S_t} \ell(b_{i,t}, r_t)$. The problem in this setting could be written as an online monotone submodular maximization problem subject to a cardinality constraint of size m. [195] proposed the online greedy algorithm for this problem whose $(1 - \frac{1}{e})$-regret is bounded by $\mathcal{O}(\sqrt{mT \ln K})$. The algorithm works as follows: There are m instantiations $\mathcal{A}_1, \dots, \mathcal{A}_m$ of no-regret algorithms for the 1-expert problem. At each round $t \in [T]$, $\mathcal{A}_i$ selects an expert $v_{i,t} \in [K]$ and the set $S_t = \{v_{1,t}, \dots, v_{m,t}\}$ is selected. $\mathcal{A}_i$ observes the reward $f_t(v_{i,t} | S_{i-1,t})$ where $S_{j,t} = \{v_{1,t}, \dots, v_{j,t}\}$.

Inspired by Algorithm 1 of [101] for online monotone submodular maximization subject to a matroid constraint, we propose the online distorted greedy in Algorithm 1 for the special case of a cardinality constraint. The algorithm is similar to the online greedy algorithm of [195] discussed above. However, in the online distorted greedy algorithm, after choosing the set S_t and observing the function f_t, we first compute the modular lower bound h_t defined as $h_t(S) = \sum_{i \in S} f_t(i | [K] \setminus \{i\})$. We define $g_t = f_t - h_t$. Note that g_t is monotone submodular as well. The reward of $\mathcal{A}_i$ for choosing $v_{i,t}$ at round t is $(1 - \frac{1}{m})^{m-i} g_t(v_{i,t} | S_{i-1,t}) + h_t(v_{i,t})$ (in case $v_{i,t} \in S_{i-1,t}$, we repeatedly take samples from the weight distribution of $\mathcal{A}_i$ over the experts until we observe an expert not in the set $S_{i-1,t}$). This technique was first introduced

Algorithm 1 Online distorted greedy algorithm

```
**Initialization:** Initialize $m$ instances $\mathcal{A}_1, \dots, \mathcal{A}_m$ of online algorithms for the 1-expert problem.
**for** $t = 1, \dots, T$ **do**
  **for** $i = 1, \dots, m$ **do**
    $\mathcal{A}_i$ chooses the expert $v_{i,t}$ and $S_{i,t} = \{v_{1,t}, \dots, v_{i,t}\}$.
  **end for**
  Play the set $S_t = S_{m,t} = \{v_{1,t}, \dots, v_{m,t}\}$ and observe $f_t$.
  Compute the modular function $h_t(S) = \sum_{i \in S} f_t(i|[K] \setminus \{i\})$ and set $g_t = f_t - h_t$.
  **for** $i = 1, \dots, m$ **do**
    Feedback the cost $-(1 - \frac{1}{m})^{m-i} g_t(v_{i,t}|S_{i-1,t}) - h_t(v_{i,t})$ to $\mathcal{A}_i$.
  **end for**
**end for**
```

by [83] for the corresponding offline problem and it allows us to obtain $(1 - \frac{c_f}{e})$-regret bounds (where $f = \sum_{t=1}^{T} f_t$) with the optimal approximation ratio (optimality was shown by [197]) instead of the $(1 - \frac{1}{e})$-regret bounds for the online greedy algorithm.

One particular choice for $\{\mathcal{A}_i\}_{i=1}^{m}$ is the WSU algorithm of [89]. We summarize the result for this choice in the theorem below.

Theorem 10.6.1 *For all $i \in [m]$, let $\mathcal{A}_i$ be an instantiation of the WSU algorithm of [89] for the 1-expert problem and denote $f = \sum_{t=1}^{T} f_t$. The online distorted greedy algorithm applied to the m-experts problem obtains the following regret bound:*

$$\mathbb{E}[(1 - \frac{c_f}{e})\text{-}R_T] \leq \sum_{i=1}^{m} R_T^{(i)},$$

where $R_T^{(i)}$ is the regret of algorithm $\mathcal{A}_i$. If $c_f = 0$, the algorithm is incentive-compatible.

If $c_f \in (0, 1]$, we can use the online greedy algorithm of [195] instead to ensure incentive compatibility while maintaining similar bounds for the $(1 - \frac{1}{e})$-regret. [89] provided $\mathcal{O}(\sqrt{T \ln K})$

regret bounds for the WSU algorithm. If we plug in this bound in the result of Theorem 10.6.1, the regret bound of the online distorted greedy algorithm is $\mathcal{O}(m\sqrt{T\ln K})$. However, this bound depends linearly on m which is suboptimal. To remedy this issue, we first provide an adaptive regret bound for the WSU algorithm below.

Theorem 10.6.2 *The regret of the WSU algorithm of [89] is bounded by* $\mathcal{O}(\sqrt{|L_T|\ln K} + \ln K)$ *where* $|L_T|$ *is the cumulative absolute loss of the algorithm.*

Note that the bound in Theorem 10.6.2 adapts to the hardness of the problem. In the worst case, we have $|L_T| = T$ and recover the $\mathcal{O}(\sqrt{T\ln K})$ bound proved in [89]. However, for smaller values of $|L_T|$, the bound in Theorem 10.6.2 improves that of [89]. For the non-strategic setting with modular utilities, [124] proposed the Component Hedge (CH) algorithm and obtained a regret bound of $\sqrt{2m\ell^* \ln(\frac{K}{m})} + m\ln(\frac{K}{m})$ where ℓ^* is the cumulative loss of the best-chosen set in hindsight. They also gave a matching lower bound for this problem. Applying the same analysis as in Theorem 10.6.2 to the setting of the naive approach with $\binom{K}{m}$ meta-experts, we can show that the regret bound of WSU matches the above lower bound.

We can use the above adaptive regret bound to improve the regret bound of the online distorted greedy algorithm. First, note that at round $t \in [T]$, the sum of the absolute value of losses incurred by $\{\mathcal{A}_i\}_{i=1}^m$ is bounded as follows:

$$\sum_{i=1}^m \left((1-\frac{1}{m})^{m-i} g_t(v_{i,t}|S_{i-1,t}) + h_t(v_{i,t})\right) \leq \sum_{i=1}^m \underbrace{\left(g_t(v_{i,t}|S_{i-1,t}) + h_t(v_{i,t})\right)}_{=f_t(v_{i,t}|S_{i-1,t})} = f_t(S_t) \leq 1.$$

Therefore, if we denote the cumulative absolute losses incurred by $\mathcal{A}_i$ with $|L_T^{(i)}|$, we have:

$$\sum_{i=1}^m |L_T^{(i)}| \leq T.$$

Using the result of Theorem 10.6.2, we know that the regret bound of the online distorted greedy algorithm is $\sum_{i=1}^m \sqrt{|L_T^{(i)}|\ln K} + m\ln K$. $\sum_{i=1}^m \sqrt{|L_T^{(i)}|}$ is maximized when $|L_T^{(i)}| = \frac{T}{m}$ for all $i \in [m]$. Thus, in the worst case, the expected $(1-\frac{c_f}{e})$-regret bound is $\mathcal{O}(\sqrt{mT\ln K} + m\ln K)$.

While we focused on submodular utility functions in this section, we can also apply the online distorted greedy algorithm to the setting with modular utilities. In this case, we have $c_f = 0$, and therefore, the algorithm is incentive-compatible and its 1-regret is bounded by $\mathcal{O}(\sqrt{mT \ln K} + m \ln K)$. Unlike the FTPL algorithm which is only approximately incentive-compatible, the online distorted greedy algorithm applied to modular utility functions is incentive-compatible. However, this comes at the price of an extra $\sqrt{m}$ term in the regret bound.

10.7 Performative prediction with strategic experts

in real-world applications, the predictions can influence the state of the world, i.e., the predictions are *performative*. For instance, the prediction of a high rate of inflation might result in people buying goods before their cash reserves depreciate too much, thereby causing inflation.

[166] considered an offline problem with a principal and an expert. The expert could be either a human or an AI system. The principal aims to elicit honest predictions about a binary event from the expert. The expert's goal is to report a prediction p such that the expected loss given by a proper loss function ℓ is minimized. In order to model the performative predictions, [166] assumed that there is a function $f : [0,1] \to [0,1]$ such that $b = f(p)$ where b is the expert's belief about the binary outcome. In other words, given the prediction p, the expert's belief over the outcome varies according to the function f. They assumed that f is only known to the expert (not the principal).

The loss function ℓ is called proper if $\mathbb{E}_{r \sim \text{Bern}(b)} \ell(b, r) \leq \mathbb{E}_{r \sim \text{Bern}(b)} \ell(p, r)$ for all $b \in [0,1], r \in \{0,1\}, p \neq b$ (Bern(b) denotes a Bernoulli distribution with probability of success b). It is called strictly proper if the inequality is strict for all $p \neq b$. [94] showed that ℓ is (strictly) proper if and only if there exists a (strictly) convex function $G : [0,1] \to \bar{\mathbb{R}}$ such that the following holds for all $p \in [0,1], r \in \{0,1\}$:

$$\ell(p, r) = -G(p) + (p - r)G'(p).$$

A prediction p is performatively optimal if $p \in \arg\min_{p\in[0,1]} \mathbb{E}_{r\sim \mathrm{Bern}(f(p))}\ell(p,r)$. The expert aims to report a performatively optimal prediction.

A point p is a fixed point if $f(p) = p$. From the principal's perspective, fixed points (or approximately fixed points) are standards of honesty. If $|p - f(p)|$ is small, the principal can draw useful conclusions from the reports. We will use this notion for the online problem. If f is continuous, a fixed point p always exists. Moreover, if f is Lipschitz continuous with parameter $L_f < 1$, then the fixed point is unique.

[166] showed that fixed points are in general not performatively optimal. Moreover, they showed that if f is L_f-Lipschitz, G is L_G-Lipschitz, and G is γ-strongly convex, we have $|p - f(p)| \leq \frac{L_f L_G}{\gamma}$ where p is the performatively optimal prediction. Also, if $L_f < 1$, the fixed point p^* is unique and $|p - p^*| \leq \frac{L_f L_G}{\gamma(1-L_f)}$.

[53] also studied the binary prediction problem with performative predictions. In particular, they focused on two choices of f (p^* is the prior belief): 1) *Drift* model: $f(p) = \alpha p + (1-\alpha)p^*$, and 2) *Reversion* model: $f(p) = \big(4(p-0.5)^2\big)0.5 + \big(1-4(p-0.5)^2\big)p^*$. The drift model applies to settings such as the prediction of inflation rate where predicting a high inflation rate results in people buying goods before their cash reserves depreciate too much and this in turn leads to inflation itself. In particular, large values of α correspond to self-fulfilling prophecies. The reversion model applies to settings where predicting an extreme value (e.g., close to 0 or 1) leads to the probability of outcome closer to 0.5, i.e., the event becomes less predictable. For instance, a prediction that one election candidate will win with near certainty might lead her fans not to show up to vote, resulting in a tighter election result.

[53] showed that under proper scoring rules and the above two choices of f, the fixed point p^* is not the expert's performatively optimal report.

In our work, we study the 1-expert problem with performative predictions, and we use the incentive-compatible WSU algorithm of [89] and obtain $\mathcal{O}(\sqrt{T\ln K})$ regret bounds under some assumptions on ℓ and f.

10.7.1 Problem formulation

In this problem, there are K experts available and each expert makes probabilistic predictions about a sequence of T binary outcomes. At round $t \in [T]$, each expert $i \in [K]$ has a private prior belief $b_{i,t} \in [0,1]$ about the outcome $r_t \in \{0,1\}$, where r_t and $\{b_{i,t}\}_{i=1}^{K}$ are chosen adversarially. Moreover, each expert i has a function $f_{i,t} : [0,1] \to [0,1]$ that maps the prediction to the outcome, i.e., $f_{i,t}(p) = b$. Expert i reports $p_{i,t} \in [0,1]$ as her prediction to the learner. Then, the learner makes her prediction $\sum_{i=1}^{K} \pi_{i,t} p_{i,t}$ (where $\sum_{i=1}^{K} \pi_{i,t} = 1$ and $\pi_{i,t} \geq 0 \ \forall i \in [K]$) and upon committing to this prediction, the outcome $r_t \in \{0,1\}$ is revealed, and the learner and expert i incur a loss of $\ell_t = \ell\big(\sum_{j=1}^{K} \pi_{j,t} f_{j,t}(\pi_t^T p_t), r_t\big)$ and $\ell_{i,t} = \ell\big(f_{i,t}(\pi_t^T p_t), r_t\big)$ respectively where $\ell : [0,1] \times \{0,1\} \to [0,1]$ is a proper loss function (note that the losses are defined with respect to the true beliefs of the experts). From the perspective of expert i, the outcome r_t is sampled according to a Bernoulli distribution with probability of success $f_{i,t}(\pi_t^T p_t)$. The goal of the learner is to minimize the regret and also incentivize experts to (approximately) report their private beliefs at each round, i.e., the learner's algorithm should be designed such that the reported prediction of expert i, $p_{i,t} = \arg\min_{p \in [0,1]} \Big(f_{i,t}\big(\sum_{j \neq i}^{K} \pi_{j,t} p_{j,t} + \pi_{i,t} p\big) \ell(p,1) + \big(1 - f_{i,t}\big(\sum_{j \neq i}^{K} \pi_{j,t} p_{j,t} + \pi_{i,t} p\big)\big) \ell(p,0) \Big)$ is close to $f_{i,t}(\pi_t^T p_t)$.

To define the regret metric, we also need to specify the benchmark. Consider the algorithms $\{\mathcal{A}_i\}_{i=1}^{K}$ where algorithm $\mathcal{A}_i$ always chooses expert i at all rounds. At round $t \in [T]$, expert i aims to report $p_{i,t}^*$ to algorithm $\mathcal{A}_i$ to minimize the following:

$$\mathbb{E}_{r_t \sim \text{Bern}\left(f_{i,t}(p_{i,t})\right)} \ell(p_{i,t}, r_t) = f_{i,t}(p_{i,t}) \ell(p_{i,t}, 1) + (1 - f_{i,t}(p_{i,t})) \ell(p_{i,t}, 0).$$

Now, we can define the regret metric as follows:

$$R_T = \sum_{t=1}^{T} \ell\big(\sum_{j=1}^{K} \pi_{j,t} f_{j,t}(\pi_t^T p_t), r_t \big) - \min_{i \in [K]} \sum_{t=1}^{T} \ell(f_{i,t}(p_{i,t}^*), r_t).$$

[166] studied an offline framework, called the "prediction markets", that is very similar to our setting. To be precise, they considered a setting with K traders where each trader submits a single prediction and gets scored according to a proper scoring rule. Each player

$i \in [K]$ has an associated number $\omega_i \in [0,1]$ (such that $\sum_{i=1}^{K} \omega_i = 1$) that represents the fraction of overall capital in the market provided by player i (similar to the expert weights $\pi_{i,t}$ in our framework). Each player i is scored according to a strictly proper loss function ℓ and the player i aims to minimize their expected loss $\ell(p_i, f(\omega^T p))$. In the game, all players simultaneously provide their prediction $\{p_i\}_{i\in[K]}$ which is the pure strategy Nash equilibrium. Then, the binary event is sampled according to a Bernoulli distribution with success probability $b = f(\omega^T p)$. The authors proved that if f is L_f-Lipschitz, G is L_G-Lipschitz, and G is γ-strongly convex, we have $|f(\omega^T p) - p_i| \leq \frac{\omega_i L_f L_G}{\gamma}$ for all $i \in [K]$.
In our setting, each expert $i \in [K]$ has separate belief functions $\{f_{i,t}\}_{t\in[T]}$ that map the learner's prediction to their belief. At round $t \in [T]$, expert $i \in [K]$ aims to minimize her expected loss defined as follows:

$$\mathbb{E}_{r_t \sim \mathrm{Bern}\left(f_{i,t}(\pi_t^T p_t)\right)} \ell(p_{i,t}, r_t) = f_{i,t}(\pi_t^T p_t)\ell(p_{i,t}, 1) + (1 - f_{i,t}(\pi_t^T p_t))\ell(p_{i,t}, 0).$$

Because the loss of each player depends on the actions of others, this could be characterized as a game between the K experts where each expert aims to minimize her expected loss defined above. We first show the following result.

Theorem 10.7.1 *For quadratic loss function (i.e., $G(p) = p^2 - p$) and the drift model for $\{f_{i,t}\}_{i\in[K],t\in[T]}$ (i.e., $f_{i,t}(p) = \alpha p + (1-\alpha)b_{i,t}$) with $\alpha < 0.5$, the aforementioned K-person game is convex.*

Therefore, we can use the result of [177] to conclude that the pure strategy Nash equilibrium $\{p_{i,t}\}_{i\in[K]}$ exists at each round $t \in [T]$ and is unique.
We provide the main result of the paper in the following theorem.

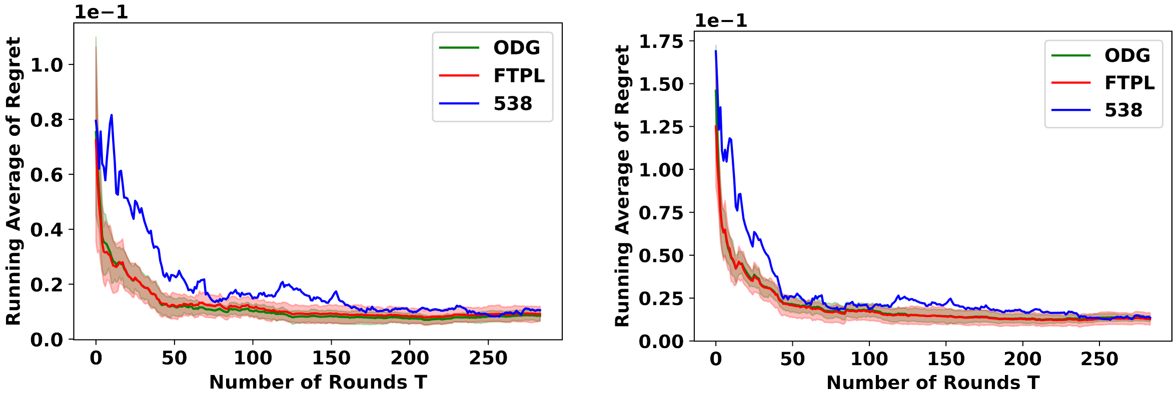
Theorem 10.7.2 *Let the loss function be quadratic (i.e., $G(p) = p^2 - p$) and assume the drift model for $\{f_{i,t}\}_{i\in[K],t\in[T]}$ (i.e., $f_{i,t}(p) = \alpha p + (1-\alpha)b_{i,t}$). Using the WSU algorithm of [89], if $\alpha = \mathcal{O}(\sqrt{\frac{\ln K}{T}})$, the regret bound is $\mathcal{O}(\sqrt{T \ln K})$, i.e., performative predictions do not lead to worse regret guarantees.*

10.8 Experiments

In this section, we evaluate the performance of our proposed algorithms for modular utility functions on a publicly available dataset from a FiveThirtyEight forecasting competition[4] in which forecasters make predictions about the match outcomes of the 2022–2023 National Football League (NFL). Before each match, FiveThirtyEight provides information on the past performance of the two opposing teams. Forecasters observe this information and make probabilistic predictions about the likelihood of each team winning the match. Considering that there are 284 different matches in the dataset, we set $T = 284$. Out of the 9982 forecasters who participated in this competition, only 274 made predictions for every single match. We consider two cases: $K = 20$ and $K = 100$. To reduce variance, for each case, we sample 5 groups of K forecasters from the 274 and run FTPL and Online Distorted Greedy (ODG) 10 times. We set $m = 5$. Given that FTPL is only approximately incentive-compatible and according to the result of Theorem 10.5.2, the reported beliefs could be $\frac{2B}{\eta-2B}$ distant from the true beliefs, we add a uniformly random value in the range $[\frac{-2B}{\eta-2B}, \frac{2B}{\eta-2B}]$ to the true beliefs to model this fact. We use the standard Laplace distribution as the perturbation for FTPL. Hence, we set $B = 1$. For both algorithms, the step size η is chosen according to our theoretical results. In Figure 10.1, we plot the average regret $\frac{1}{t}\mathbb{E}\big[\max_{S\subseteq[K]:|S|=m} \sum_{\tau=1}^{t} f_\tau(S) - \sum_{\tau=1}^{t} f_\tau(S_\tau)\big]$ of the two algorithms over time (along with error bands corresponding to 20th and 80th percentiles) along with that of the FiveThirtyEight aggregated predictions for $K = 20$ and $K = 100$ settings. Note that while our proposed algorithms choose m predictions at each round $t \in [T]$, the FiveThirtyEight aggregated prediction is a single scalar value. The plots suggest that while the regret of all three algorithms converges to zero as t gets larger, both our proposed algorithms have superior performance compared to that of the FiveThirtyEight predictions.

---

[4] https://github.com/fivethirtyeight/nfl-elo-game



(a)

(b)

Figure 10.1: Running average of regret over time for (a) $K = 20$ and (b) $K = 100$.

10.9 Conclusion and future directions

In this chapter, we studied the *m-experts* problem, a generalization of the standard binary prediction with expert advice problem where at each round $t \in [T]$: 1) The algorithm is allowed to pick $m \geq 1$ experts and its utility is a modular or submodular function of the chosen experts, and 2) The experts are strategic and may misreport their true beliefs about the t-th event to increase their probability of being chosen at the next round (round $t+1$). The goal is to design algorithms that incentivize experts to report truthfully (i.e., incentive-compatible) and obtain sublinear regret bounds with respect to the true beliefs of the experts (i.e., no-regret). We proposed two algorithmic frameworks for this problem. In Section 10.5, we introduced the Follow the Perturbed Leader (FTPL) algorithm. Under a certain condition for the noise distribution (Condition 10.5.1), this algorithm is approximately incentive-compatible and achieves sublinear regret bounds for modular utility functions. Moreover, in Section 10.6, we proposed the online distorted greedy algorithm that applies to both modular and submodular utility functions. This algorithm is incentive-compatible but its regret bound is slightly worse than that of FTPL. Moreover, we studied the 1-expert problem under the assumption that experts are strategic and the predictions are performative. We showed that for a particular choice of the loss function and the belief mapping function, if the influence of the predictions on outcomes is bounded, the standard $\mathcal{O}(\sqrt{T \ln K})$ regret bound could be obtained, i.e., the performative prediction setting does not lead to worse regret guarantees.

This work could be extended in several interesting directions. First, none of the algorithms discussed here or in prior works have taken into account the properties of the quadratic loss function. In particular, this loss function is exp-concave, and [121] showed that for exp-concave loss functions in the 1-expert problem, the regret bound could be improved to $\mathcal{O}(\ln K)$ using the Hedge algorithm without the incentive compatibility property. Designing incentive-compatible algorithms with similarly improved regret bounds for the 1-expert and m-experts problems is yet to be done. To obtain the $O(\ln K)$ regret bound for the 1-expert problem (with squared loss) using the Hedge algorithm, the algorithm makes a single prediction $\sum_{i=1}^{K} \pi_{i,t} p_{i,t}$ at round $t \in [T]$ and its loss is $(\sum_{i=1}^{K} \pi_{i,t} p_{i,t} - r_t)^2$. In other words, choosing an expert $i \in [K]$ with probability $\pi_{i,t}$ at round t is not good enough to obtain the improved $\mathcal{O}(\ln K)$ regret bound. Moving on to the m-expert problem, the main challenge for obtaining regret bounds better than $\mathcal{O}(\sqrt{T})$ is to decide how to aggregate the K predictions as m scalar values. Second, while we focused on the particular choice of quadratic loss functions, the setting could be extended to other loss functions as well. It is not clear to what extent our results hold when moving beyond the quadratic loss function. Moreover, [139] introduced a framework for augmenting online algorithms for various online problems with predictions or pieces of advice. An interesting future research direction is to extend this setting to the case where the predictions are given by strategic experts and study incentive compatibility guarantees for online problems beyond the m-experts problem. Finally, it is interesting to study the performative binary prediction problem with strategic experts under more general choices of the loss function and the belief mapping function and see if our results could be extended.

BIBLIOGRAPHY

[1] Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823. PMLR, 2014.

[2] Jacob D Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

[3] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *International Conference on Machine Learning*, pages 32–40. PMLR, 2017.

[4] Akshay Agrawal, Steven Diamond, and Stephen Boyd. Disciplined geometric programming. *Optimization Letters*, 13(5):961–976, 2019.

[5] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[6] Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, page 989–1006, New York, NY, USA, 2014. Association for Computing Machinery.

[7] Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1405–1424. SIAM, 2014.

[8] Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. Structured robust submodular maximization: Offline and online algorithms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3128–3137. PMLR, 2019.

[9] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

[10] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[11] Y. Azar, N. Buchbinder, T. H. Chan, S. Chen, I. R. Cohen, A. Gupta, Z. Huang, N. Kang, V. Nagarajan, J. Naor, and D. Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 148–157, 2016.

[12] Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalya Panigrahi. Online mixed packing and covering. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 85–100. Society for Industrial and Applied Mathematics, 2013.

[13] Yossi Azar, Ilan Reuven Cohen, and Debmalya Panigrahi. Online covering with convex objectives and applications. *arXiv preprint arXiv:1412.3507*, 2014.

[14] Yossi Azar, Ilan Reuven Cohen, and Alan Roytman. Online lower bounds via duality. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1050. Society for Industrial and Applied Mathematics, 2017.

[15] Francis Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, pages 1–41, 2016.

[16] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018.

[17] Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1497–1514. SIAM, 2014.

[18] Santiago R. Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.

[19] MohammadHossein Bateni, Mohammadtaghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Transactions on Algorithms (TALG)*, 9(4):32, 2013.

[20] Bernard Bercu, Bernard Delyon, and Emmanuel Rio. *Concentration inequalities for sums and martingales.* Springer, 2015.

[21] Omar Besbes and Assaf Zeevi. Blind network revenue management. *Operations research*, 60(6):1537–1550, 2012.

[22] Kshipra Bhawalkar, Sreenivas Gollapudi, and Debmalya Panigrahi. Online set cover with set requests. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 28. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[23] An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. *Advances in Neural Information Processing Systems*, 30, 2017.

[24] Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 498–507. JMLR. org, 2017.

[25] Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120. PMLR, 2017.

[26] 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*, pages 644–653, 2019.

[27] Yatao Bian, Joachim M Buhmann, and Andreas Krause. Continuous submodular function maximization. *arXiv preprint arXiv:2006.13474*, 2020.

[28] Jeffrey Bilmes and Wenruo Bai. Deep submodular functions. *arXiv preprint arXiv:1701.08939*, 2017.

[29] JR Birge and F Louveaux. Springer series in operations research, introduction to stochastic programming, 1997.

[30] Avrim Blum, Anupam Gupta, Yishay Mansour, and Ankit Sharma. Welfare and profit maximization with production costs. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 77–86. IEEE, 2011.

[31] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[32] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 105–112. IEEE, 2001.

[33] Peter Brucker. An o(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, 1984.

[34] Simon Bruggmann and Rico Zenklusen. Submodular maximization through the lens of linear programming. *Mathematics of Operations Research*, 2019.

[35] Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, 2, 2011.

[36] Niv Buchbinder, Shahar Chen, Anupam Gupta, Viswanath Nagarajan, et al. Online packing and covering framework with convex objectives. *arXiv preprint arXiv:1412.8347*, 2014.

[37] Niv Buchbinder and Moran Feldman. Deterministic algorithms for submodular maximization problems. *ACM Transactions on Algorithms (TALG)*, 14(3):32, 2018.

[38] Niv Buchbinder and Moran Feldman. Submodular functions maximization problems. In *Handbook of Approximation Algorithms and Metaheuristics*, pages 753–788. Chapman and Hall/CRC, 2018.

[39] Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a nonsymmetric technique. *Mathematics of Operations Research*, 2019.

[40] Niv Buchbinder, Moran Feldman, Yuval Filmus, and Mohit Garg. Online submodular maximization: Beating 1/2 made simple. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 101–114. Springer, 2019.

[41] Niv Buchbinder, Moran Feldman, and Roy Schwartz. Online submodular maximization with preemption. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1202–1216. Society for Industrial and Applied Mathematics, 2015.

[42] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

[43] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

[44] Niv Buchbinder, Joseph Seffi Naor, et al. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3(2–3):93–263, 2009.

[45] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

[46] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM Journal on Computing*, 40(6):1740–1766, January 2011.

[47] X. Cao and K. J. R. Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on Automatic Control*, 64(7):2665–2680, 2019.

[48] Xuanyu Cao and K. J. Ray Liu. On the Time-Varying Constraints and Bandit Feedback of Online Convex Optimization. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, Kansas City, MO, May 2018. IEEE.

[49] Xuanyu Cao, Junshan Zhang, and H. Vincent Poor. A Novel Online Convex Optimization Algorithm Based on Virtual Queues. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, Kansas City, MO, May 2018. IEEE.

[50] Xuanyu Cao, Junshan Zhang, and H. Vincent Poor. A Virtual-Queue-Based Algorithm for Constrained Online Convex Optimization With Applications to Data Center Resource Allocation. *IEEE Journal of Selected Topics in Signal Processing*, 12(4):703–716, August 2018. ZSCC: 0000004.

[51] Francesco Cellinese, Gianlorenzo D'Angelo, Gianpiero Monaco, and Yllka Velaj. Generalized budgeted submodular set function maximization. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[52] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

[53] Alan Chan. Scoring rules for performative binary prediction. *arXiv preprint arXiv:2207.02847*, 2022.

[54] TH Chan, Zhiyi Huang, and Ning Kang. Online convex covering and packing problems. *arXiv preprint arXiv:1502.01802*, 2015.

[55] Anamay Chaturvedi, Huy Lê Nguyen, and Lydia Zakynthinou. Differentially private decomposable submodular maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6984–6992, 2021.

[56] Anamay Chaturvedi, Huy Nguyen, and Thy Dinh Nguyen. Streaming submodular maximization with differential privacy. In *International Conference on Machine Learning*, pages 4116–4143. PMLR, 2023.

[57] Chandra Chekuri, TS Jayram, and Jan Vondrák. On multiplicative weight updates for concave and submodular function maximization. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 201–210. ACM, 2015.

[58] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010.

[59] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

[60] Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2018.

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

[62] Tianyi Chen and Georgios B. Giannakis. Bandit Convex Optimization for Scalable and Dynamic IoT Management. *IEEE Internet of Things Journal*, 6(1):1276–1286, February 2019.

[63] Tianyi Chen, Qing Ling, and Georgios B Giannakis. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24):6350–6364, 2017.

[64] Tianyi Chen, Yanning Shen, Qing Ling, and Georgios B. Giannakis. Online learning for “thing-adaptive” Fog Computing in IoT. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 664–668, Pacific Grove, CA, October 2017. IEEE. ZSCC: 0000006.

[65] Xinjia Chen. Concentration Inequalities for Bounded Random Vectors. *arXiv:1309.0003 [cs, math, stat]*, August 2013. ZSCC: 0000004.

[66] Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran. Scalable and distributed submodular maximization with matroid constraints. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 435–442, Mumbai, India, May 2015. IEEE. ZSCC: 0000007.

[67] Alon Cohen and Tamir Hazan. Following the perturbed leader for online structured learning. In *International Conference on Machine Learning*, pages 1034–1042. PMLR, 2015.

[68] Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.

[69] Abhimanyu Das and David Kempe. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1057–1064, 2011.

[70] Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78, 2009.

[71] Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 137–144. ACM, 2012.

[72] Gianni Di Pillo. Exact penalty methods. In *Algorithms for Continuous Optimization*, pages 209–253. Springer, 1994.

[73] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[74] MV Dolgopolik. A unifying theory of exactness of linear penalty functions. *Optimization*, 65(6):1167–1202, 2016.

[75] Brian W Dolhansky and Jeff A Bilmes. Deep submodular functions: Definitions and learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[76] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[77] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 715–724, New York, NY, USA, 2010. Association for Computing Machinery.

[78] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[79] Reza Eghbali and Maryam Fazel. Designing smoothing functions for improved worst-case competitive ratio in online optimization. In *Advances in Neural Information Processing Systems*, pages 3287–3295, 2016.

[80] Reza Eghbali, James Saunderson, and Maryam Fazel. Competitive online algorithms for resource allocation over the positive semidefinite cone. *Mathematical Programming*, pages 1–26, 2018.

[81] Alina Ene and Huy L Nguyen. Constrained submodular maximization: Beyond 1/e. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 248–257. IEEE, 2016.

[82] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In Mark de Berg and Ulrich Meyer, editors, *Algorithms – ESA 2010*, pages 182–194, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

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

[84] Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 570–579. IEEE, 2011.

[85] Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Improved competitive ratios for submodular secretary problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 218–229. Springer, 2011.

[86] Junkai Feng, Ruiqi Yang, Yapu Zhang, and Zhenning Zhang. Online weakly dr-submodular optimization with stochastic long-term constraints. In *International Conference on Theory and Applications of Models of Computation*, pages 32–42. Springer, 2022.

[87] Kris Johnson Ferreira, David Simchi-Levi, and He Wang. Online network revenue management using thompson sampling. *Operations research*, 66(6):1586–1602, 2018.

[88] Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014.

[89] Rupert Freeman, David Pennock, Chara Podimata, and Jennifer Wortman Vaughan. No-regret and incentive-compatible online learning. In *International Conference on Machine Learning*, pages 3270–3279. PMLR, 2020.

[90] Rafael Frongillo, Robert Gomez, Anish Thilagar, and Bo Waggoner. Efficient competitions and online learning with strategic forecasters. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 479–496, 2021.

[91] Dan Garber and Elad Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

[92] Dan Garber, Gal Korcia, and Kfir Levy. Online convex optimization in the random order model. In *International Conference on Machine Learning*, pages 3387–3396. PMLR, 2020.

[93] Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-Optimal MAP Inference for Determinantal Point Processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2735–2743. Curran Associates, Inc., 2012.

[94] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[95] Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*, 2014.

[96] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2733–2741. Curran Associates, Inc., 2013.

[97] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1106–1125. SIAM, 2010.

[98] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *International Workshop on Internet and Network Economics*, pages 246–257. Springer, 2010.

[99] Eric Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.

[100] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[101] Nicholas Harvey, Christopher Liaw, and Tasuku Soma. Improved algorithms for online submodular maximization via first-order regret bounds. *Advances in Neural Information Processing Systems*, 33, 2020.

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

[103] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[104] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[105] Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1843–1850, Madison, WI, USA, 2012. Omnipress.

[106] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32:151–178, 1998.

[107] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[108] Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: A primal dual approach. *Games and Economic Behavior*, 118:648 – 667, 2019.

[109] Shinji Ito and Ryohei Fujimaki. Large-scale price optimization via network flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3855–3863. Curran Associates, Inc., 2016.

[110] Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization: Extended version. In *UAI*, 2014.

[111] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online Optimization : Competing with Dynamic Comparators. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 398–406, San Diego, California, USA, 09–12 May 2015. PMLR.

[112] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1. JMLR Workshop and Conference Proceedings, 2012.

[113] Rodolphe Jenatton, Jim Huang, and Cedric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411, 2016.

[114] Rodolphe Jenatton, Jim Huang, Dominik Csiba, and Cedric Archambeau. Online optimization and regret guarantees for non-additive long-term constraints. *arXiv:1602.05394 [cs, math, stat]*, June 2016. ZSCC: 0000010.

[115] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

[116] Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic submodular maximization: The case of coverage functions. In *Advances in Neural Information Processing Systems*, pages 6853–6863, 2017.

[117] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[118] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[119] Thomas Kesselheim and Andreas Tönnis. Submodular secretary problems: Cardinality, matching, and linear constraints. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[120] Katrin Kirchhoff and Jeff Bilmes. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, 2014.

[121] Jyrki Kivinen and Manfred K Warmuth. Averaging expert predictions. In *European Conference on Computational Learning Theory*, pages 153–167. Springer, 1999.

[122] Robert Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 630–631. Society for Industrial and Applied Mathematics, 2005.

[123] Pushmeet Kohli, M Pawan Kumar, and Philip HS Torr. P^3 & beyond: Move making algorithms for solving higher order functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2008.

[124] Wouter M Koolen, Manfred K Warmuth, Jyrki Kivinen, et al. Hedging structured concepts. In *COLT*, pages 93–105. Citeseer, 2010.

[125] Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing*, 47(3):1056–1086, 2018.

[126] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3(71-104):3, 2014.

[127] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.

[128] Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 545–554. Society for Industrial and Applied Mathematics, 2009.

[129] Ariel Kulik, Hadas Shachnai, and Tami Tamir. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Mathematics of Operations Research*, 38(4):729–739, 2013.

[130] Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

[131] Nicolas S Lambert, John Langford, Jennifer Wortman Vaughan, Yiling Chen, Daniel M Reeves, Yoav Shoham, and David M Pennock. An axiomatic characterization of wagering mechanisms. *Journal of Economic Theory*, 156:389–416, 2015.

[132] Nicolas S Lambert, John Langford, Jennifer Wortman, Yiling Chen, Daniel Reeves, Yoav Shoham, and David M Penno k. Self-financed wagering mechanisms for forecasting. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 170–179, 2008.

[133] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[134] Wenxin Li and Ness Shroff. Towards practical constrained monotone submodular maximization. *arXiv preprint arXiv:1804.08178*, 2018.

[135] Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In *International Conference on Machine Learning*, pages 3944–3952. PMLR, 2019.

[136] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520, 2011.

[137] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[138] Lutz Dümbgen, Sara A. van de Geer, Mark C. Veraar, and Jon A. Wellner. Nemirovski's Inequalities Revisited. *The American Mathematical Monthly*, 117(2):138, 2010. ZSCC: 0000049.

[139] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68(4):1–25, 2021.

[140] Takanori Maehara, Atsuhiro Narita, Jun Baba, and Takayuki Kawabata. Optimal bidding strategy for brand advertising. In *IJCAI*, pages 424–432, 2018.

[141] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

[142] Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Efficient Constrained Regret Minimization. *arXiv:1205.2265 [cs]*, October 2012. ZSCC: 0000002.

[143] Shie Mannor, John N Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(Mar):569–590, 2009.

[144] Alberto Marchetti-Spaccamela and Carlo Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68(1-3):73–104, 1995.

[145] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, page 94–103, USA, 2007. IEEE Computer Society.

[146] Aranyak Mehta et al. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

[147] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.

[148] Baharan Mirzasoleiman, Morteza Zadimoghaddam, and Amin Karbasi. Fast distributed submodular cover: Public-private data summarization. In *Advances in Neural Information Processing Systems*, pages 3594–3602, 2016.

[149] Marko Mitrovic, Mark Bun, Andreas Krause, and Amin Karbasi. Differentially private submodular maximization: Data summarization in disguise. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2478–2487, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[150] Eyal Mizrachi, Roy Schwartz, Joachim Spoerhase, and Sumedha Uniyal. A tight approximation for submodular maximization with mixed packing and covering constraints. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[151] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201, 2016.

[152] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895. PMLR, 2018.

[153] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Decentralized submodular maximization: Bridging discrete and continuous settings. In *International Conference on Machine Learning*, pages 3613–3622, 2018.

[154] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *The Journal of Machine Learning Research*, 21(1):4232–4280, 2020.

[155] Theodore S Motzkin and Ernst G Straus. Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[156] Samrat Mukhopadhyay, Sourav Sahoo, and Abhishek Sinha. k-experts-online policies and fundamental limits. In *International Conference on Artificial Intelligence and Statistics*, pages 342–365. PMLR, 2022.

[157] Viswanath Nagarajan and Xiangkun Shen. Online covering with sum of ell_q-norm objectives. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[158] M. J. Neely. Universal scheduling for networks with arbitrary traffic, channels, and mobility. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1822–1829, 2010.

[159] Michael J Neely and Hao Yu. Online convex optimization with time-varying constraints. *arXiv preprint arXiv:1702.04783*, 2017.

[160] George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

[161] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[162] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[163] Rad Niazadeh, Tim Roughgarden, and Joshua Wang. Optimal algorithms for continuous non-monotone submodular and dr-submodular maximization. In *Advances in Neural Information Processing Systems*, pages 9594–9604, 2018.

[164] Sampo Niskanen and Patric RJ Östergård. Cliquer user's guide, version 1.0, communications laboratory, helsinki university of technology, espoo. In *Tech rep, Finland, Tech Rep.* 2003.

[165] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[166] Caspar Oesterheld, Johannes Treutlein, Emery Cooper, and Rubi Hudson. Incentivizing honest performative predictions with proper scoring rules. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 1564–1574. PMLR, 31 Jul–04 Aug 2023.

[167] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[168] Panos M Pardalos and Naina Kovoor. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, 46(1):321–328, 1990.

[169] Stephen Pasteris, Fabio Vitale, Kevin Chan, Shiqiang Wang, and Mark Herbster. Maxhedge: Maximizing a maximum online. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1851–1859. PMLR, 2019.

[170] Santiago Paternain and Alejandro Ribeiro. Online learning of feasible strategies in unknown environments. *IEEE Transactions on Automatic Control*, 62(6):2807–2822, 2017.

[171] Iosif Pinelis. Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, pages 1679–1706, 1994.

[172] Chao Qian, Yibo Zhang, Ke Tang, and Xin Yao. On multiset selection with size constraints. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[173] Akbar Rafiey and Yuichi Yoshida. Fast and private submodular and k-submodular functions maximization with matroid constraints. In *International conference on machine learning*, pages 7887–7897. PMLR, 2020.

[174] Prasanna Raut, Omid Sadeghi, and Maryam Fazel. Online dr-submodular maximization: Minimizing regret and constraint violation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9395–9402, 2021.

[175] Mark D Reid and Robert C Williamson. Surrogate regret bounds for proper losses. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 897–904, 2009.

[176] Adrian Rivera, He Wang, and Huan Xu. The online saddle point problem and online convex optimization with knapsacks. *arXiv preprint arXiv:1806.08301*, 2018.

[177] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.

[178] Tim Roughgarden and Okke Schrijvers. Online prediction with selfish experts. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1300–1310. Curran Associates, Inc., 2017.

[179] Tim Roughgarden and Joshua R Wang. An optimal learning algorithm for online unconstrained submodular maximization. In *Conference On Learning Theory*, pages 1307–1325, 2018.

[180] Omid Sadeghi, Reza Eghbali, and Maryam Fazel. Competitive algorithms for online budget-constrained continuous dr-submodular problems. *arXiv preprint arXiv:1907.00312*, 2019.

[181] Omid Sadeghi and Maryam Fazel. Online continuous dr-submodular maximization with long-term budget constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 4410–4419. PMLR, 2020.

[182] Omid Sadeghi and Maryam Fazel. Differentially private monotone submodular maximization under matroid and knapsack constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 2908–2916. PMLR, 2021.

[183] Omid Sadeghi and Maryam Fazel. Fast first-order methods for monotone strongly dr-submodular maximization. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA23)*, pages 169–179. SIAM, 2023.

[184] Omid Sadeghi and Maryam Fazel. No-regret online prediction with strategic experts. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[185] Omid Sadeghi, Prasanna Raut, and Maryam Fazel. A single recipe for online submodular maximization with adversarial or stochastic constraints. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14712–14723. Curran Associates, Inc., 2020.

[186] Omid Sadeghi, Prasanna Raut, and Maryam Fazel. Improved regret bounds for online submodular maximization. *arXiv preprint arXiv:2106.07836*, 2021.

[187] Sebastian Perez Salazar and Rachel Cummings. Differentially private online submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1279–1287. PMLR, 2021.

[188] Pier Giuseppe Sessa, Maryam Kamgarpour, and Andreas Krause. Bounding inefficiency of equilibria in continuous actions games using submodularity and curvature. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2017–2027. PMLR, 2019.

[189] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[190] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 807–814, New York, NY, USA, 2007. Association for Computing Machinery.

[191] Uri Sherman, Tomer Koren, and Yishay Mansour. Optimal rates for random order online optimization. *Advances in Neural Information Processing Systems*, 34:2097–2108, 2021.

[192] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 1167–1178, New York, NY, USA, 2013. Association for Computing Machinery.

[193] Tasuku Soma. No-regret algorithms for online k-submodular maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1205–1214, 2019.

[194] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low-noise and fast rates. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, page 2199–2207, Red Hook, NY, USA, 2010. Curran Associates Inc.

[195] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems*, 21, 2008.

[196] Wen Sun, Debadeepta Dey, and Ashish Kapoor. Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*, pages 3280–3288. PMLR, 2017.

[197] Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research*, 42(4):1197–1218, 2017.

[198] Philip E Tetlock and Dan Gardner. *Superforecasting: The art and science of prediction.* Random House, 2016.

[199] Nguyen Kim Thang. Online primal-dual algorithms with configuration linear programs. In *31st International Symposium on Algorithms and Computation (ISAAC 2020).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[200] Jan Vondrák. Submodularity in combinatorial optimization. *Charles University, Prague*, 2007.

[201] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.

[202] Jan Vondrák. Submodularity and curvature: The optimal algorithm (combinatorial optimization and discrete algorithms). 23:253–266, 2010.

[203] Volodimir G Vovk. Aggregating strategies. In *Annual Workshop on Computational Learning Theory: Proceedings of the third annual workshop on Computational learning theory, 1990.* Association for Computing Machinery, Inc, 1990.

[204] Yanhao Wang, Tianchen Zhou, Cen Chen, and Yinggui Wang. Federated submodular maximization with differential privacy. *IEEE Internet of Things Journal*, 2023.

[205] Manfred K Warmuth and Dima Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(Oct):2287–2320, 2008.

[206] Xiaohan Wei, Hao Yu, and Michael J Neely. Online primal-dual mirror descent under stochastic constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–36, 2020.

[207] Bryan Wilder. Risk-sensitive submodular optimization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[208] Jiahao Xie, Zebang Shen, Chao Zhang, Boyu Wang, and Hui Qian. Efficient projection-free online methods with stochastic recursive gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6446–6453, 2020.

[209] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd*

International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pages 449–457, New York, New York, USA, 20–22 Jun 2016. PMLR.

[210] Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and Karl Henrik Johansson. Distributed bandit online convex optimization with time-varying coupled inequality constraints. *IEEE Transactions on Automatic Control*, 66(10):4620–4635, 2020.

[211] Xinlei Yi, Xiuxian Li, Tao Yang, Lihua Xie, Tianyou Chai, and H Karl. Regret and cumulative constraint violation analysis for distributed online constrained convex optimization. *IEEE Transactions on Automatic Control*, 2022.

[212] Hao Yu, Michael J. Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1427–1437, Red Hook, NY, USA, 2017. Curran Associates Inc.

[213] Jianjun Yuan and Andrew Lamperski. Online convex optimization for cumulative constraints. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6137–6146. Curran Associates, Inc., 2018.

[214] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1323–1333. Curran Associates, Inc., 2018.

[215] Lijun Zhang, Jinfeng Yi, Rong Jin, Ming Lin, and Xiaofei He. Online kernel learning with a near optimal sparsity bound. In *International Conference on Machine Learning*, pages 621–629, 2013.

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

[217] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *International Workshop on Internet and Network Economics*, pages 566–576. Springer, 2008.

[218] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

Appendix A

APPENDIX OF CHAPTER 3

A.1 Motivating applications

There are several online budgeted discrete submodular problems whose continuous generalization could be cast in our framework. We have listed a number of these applications below:

Online knapsack constrained continuous DR-submodular maximization. In the discrete problem considered in [140], there is a ground set of elements V and a budget constraint $b \in \mathbb{R}_+$. At step $t \in [m]$, an element $v \in V$ with the corresponding cost $c(v) \in \mathbb{R}_+$ arrives online, and we should decide whether to choose v. The overall objective is as follows:

$$\begin{array}{ll} \text{maximize} & F(V') \\ \text{subject to} & \sum_{v \in V'} c(v) \leq b \\ & V' \subseteq V \end{array} ,$$

where $F : 2^V \to \mathbb{R}_+$ is a monotone non-decreasing submodular function and V' is the set of chosen elements. Note that at each step, the value of the function is only known over subsets of items that have already arrived.

Consider the continuous relaxation of this problem where at each step, we are allowed to take a fraction of the arriving element. This problem could be formulated as:

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & \sum_{t=1}^{m} c_t x_t \leq b \\ & 0 \leq x_t \leq 1 \ \forall t \in [m] \end{array} ,$$

where $x = [x_1, \ldots, x_m]^T$, $c_t \in \mathbb{R}_+$ is the cost corresponding to the t-th arriving element and $f : [0,1]^m \to \mathbb{R}_+$ is the multilinear extension of the function F.

Online generalized maximum coverage problem. In this problem, there are m subsets $C_1, \ldots, C_m$ of the ground set V with corresponding costs $c_1, \ldots, c_m$ that are arriving one by one. At step $t \in [m]$, subset C_t could be chosen with confidence level $x_t \in [0,1]$ and the set of covered elements when choosing C_t with confidence x_t is modeled with a monotone normalized covering function $p_t : [0,1] \to 2^{C_t}$ which is not known in advance and is revealed online. The goal is to choose subsets from $C_1, \ldots, C_m$ with confidence level to maximize the overall number of covered elements $|\bigcup_{t=1}^m p_t(x_t)|$ while satisfying the budget constraint $\sum_{t=1}^m c_t x_t \leq b$. The problem could be formulated as follows:

$$\begin{array}{ll} \text{maximize} & |\bigcup_{t=1}^m p_t(x_t)| \\ \text{subject to} & \sum_{t=1}^m c_t x_t \leq b \\ & 0 \leq x_t \leq 1 \ \forall t \in [m] \end{array} .$$

Online continuous DR-submodular welfare maximization. In the submodular welfare problem, there is a set $M = \{1, \ldots, m\}$ of m items and a set $N = \{1, \ldots, n\}$ of n agents. Each agent $i \in N$ has a valuation function $F_i : 2^M \to \mathbb{R}_+$ over subsets of items. Valuation functions are assumed to be submodular and monotone non-decreasing. In this problem, the goal is to partition the items among the agents as $S = (S_1, \ldots, S_n)$, where $S_s \cap S_t = \emptyset \ \forall s, t \in N$ and $\cup_{s=1}^n S_s = M$, in a way that the value of the partition $F(S) = \sum_{i=1}^n F_i(S_i)$ is maximized [201]. Now, consider the continuous relaxation of this problem in the online setting: Each agent has a valuation function $f_i : [0,1]^M \to \mathbb{R}_+$ which is the multilinear extension of the submodular function F_i. At step $t \in M$, item t arrives, and the valuations of agents over subsets of items $\{1, \ldots, t\}$ are accessible. The algorithm should assign item t fractionally among the agents to maximize the aggregate valuation. The problem could be written as:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n f_i(\hat{x}_i) \\ \text{subject to} & \sum_{i=1}^n x_{it} \leq 1 \ \forall t \in M \\ & x_{it} \geq 0 \ \forall i \in [n], t \in [m] \end{array} ,$$

where $\hat{x}_i = [x_{i1}, \ldots, x_{im}]^T$. Note that in this problem, there are no budget constraints.

Online DR-submodular generalized assignment problem. In this problem, there are n bins and m items. Each bin $i \in [n]$ has an associated collection of feasible sets given by the knapsack constraint $\mathcal{F}_i = \{S \subset [m] \ : \ \sum_{j \in S} c_{ij} \leq b_i\}$ and a monotone submodular valuation function $F_i : \{0,1\}^m \to \mathbb{R}_+$ which captures the diversity of the items in each bin. In the online setting, the set of items $t \in [m]$ arrive one by one, and upon arrival of each item t, c_{it} and values of the functions F_i over subsets of $\{1, \dots, t\}$ for all $i \in [n]$ are revealed. The goal is to partition the items among the bins so that the aggregate valuation of the partition is maximized. If the valuation function F_i is modular for all $i \in [n]$, this problem reduces to the Generalized Assignment Problem (GAP) [201]. Now, consider the continuous relaxation of this online problem where fractional assignments of items to bins are possible. The problem could be formulated as:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} f_i(\hat{x}_i) \\ \text{subject to} & \sum_{i=1}^{n} x_{it} \leq 1 \ \forall t \in [m] \\ & \sum_{t=1}^{m} c_{it} x_{it} \leq b_i \ \forall i \in [n] \end{array},$$

where $\hat{x}_i = [x_{i1}, \dots, x_{im}]^T$ and $f_i : [0,1]^m \to \mathbb{R}_+$ is the multilinear extension of the submodular valuation function F_i of the i-th bin.

A.2 Derivation of the dual problem

Let

$$I_{F_t}(x) = \begin{cases} \infty & \text{o.w.} \end{cases},$$

i.e., the convex indicator function of the set F_t.

Remember the offline constrained optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} H_i(\hat{x}_i) \\ \text{subject to} & x_t \in F_t \subseteq \mathbb{R}_+^n \ \forall t \in [m] \\ & \hat{c}_i^T \hat{x}_i \leq 1 \ \forall i \in [n] \end{array}. \tag{A.1}$$

We can equivalently write the optimization problem as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} H_i(\hat{d}_i) \\ \text{subject to} & x_t \in F_t \subseteq \mathbb{R}_+^n \ \forall t \in [m] \\ & \hat{c}_i^T \hat{x}_i \leq 1 \ \forall i \in [n] \\ & \hat{d}_i = \hat{x}_i \ \forall i \in [n] \end{array}. \tag{A.2}$$

We derive the dual of problem (A.2) below:

$$
\begin{aligned}
g(\hat{y}_i, z_i) &= \inf_{\hat{d}_i, \hat{e}_i, X} \sum_{i=1}^{n} -H_i(\hat{d}_i) + \sum_{i=1}^{n} \hat{y}_i^T (\hat{d}_i - \begin{bmatrix} x_{i1} \\ \vdots \\ x_{im} \end{bmatrix}) + \sum_{i=1}^{n} z_i \big(\hat{c}_i^T \begin{bmatrix} x_{i1} \\ \vdots \\ x_{im} \end{bmatrix} - 1\big) + \sum_{t=1}^{m} I_{F_t}(x_t) \\
&= \sum_{i=1}^{n} \inf_{\hat{d}_i} \big(\hat{y}_i^T \hat{d}_i - H_i(\hat{d}_i)\big) - \sum_{i=1}^{n} z_i + \sum_{t=1}^{m} \inf_{x_t \in F_t} \big(I_{F_t}(x_t) - \langle \underbrace{\begin{bmatrix} y_{1t} - z_1 c_{1t} \\ \vdots \\ y_{nt} - z_n c_{nt} \end{bmatrix}}_{v_t}, x_t \rangle\big) \\
&= \sum_{i=1}^{n} H_i^*(\hat{y}_i) - \sum_{i=1}^{n} z_i - \sum_{t=1}^{m} \sup_{x_t \in F_t} \big(\langle v_t, x_t \rangle - I_{F_t}(x_t)\big) \\
&= \sum_{i=1}^{n} H_i^*(\hat{y}_i) - \sum_{i=1}^{n} z_i - \sum_{t=1}^{m} \sigma_{F_t}(v_t),
\end{aligned}
$$

where $\hat{y}_i = [y_{i1}, \ldots, y_{im}]^T$, $\sigma_{F_t}(u) = \sup_{w \in F_t} w^T u$ is the support function of the set F_t and $H_i^*(u) = \inf_w \big(w^T u - H_i(w)\big)$ is the concave conjugate function of H_i. Therefore, the dual problem is:

$$
\begin{array}{ll}
\text{minimize} & \sum_{t=1}^{m} \sigma_{F_t}\big(v_t\big) - \sum_{i=1}^{n} H_i^*(\hat{y}_i) + \sum_{i=1}^{n} z_i \\
\text{subject to} & z_i \geq 0 \ \forall i \in [n]
\end{array} .
$$

A.3 Connection between submodular total curvature and α

First, note that f, i.e., the multilinear extension of the discrete submodular function F, satisfies the DR property and $f(0) = 0$. Since f is linear in each of its arguments, we can write:

$$
\langle \nabla f(x), x \rangle = \sum_{t} \mathbb{E}_{S \sim x}[F(S \cup \{t\}) - F(S \setminus \{t\})] x_t. \tag{A.3}
$$

Depending on whether $t \in S$ or not, one of the terms $\big(F(S \cup \{t\}) - F(S)\big)$ or $\big(F(S) - F(S \setminus \{t\})\big)$ would be zero. So, by definition of total curvature of F, i.e., κ_F, we have:

$$
\begin{aligned}
F(S \cup \{t\}) - F(S \setminus \{t\}) &= \big(F(S \cup \{t\}) - F(S)\big) + \big(F(S) - F(S \setminus \{t\})\big) \\
&\geq (1 - \kappa_F) F(\{t\}).
\end{aligned} \tag{A.4}
$$

Combining (A.3) and (A.4), we have:

$$\langle \nabla f(x), x \rangle \geq (1 - \kappa_F) \sum_t x_t F(\{t\}). \tag{A.5}$$

Defining $\hat{x}_t = [x_1, \dots, x_t, 0, \dots, 0]^T$, we can write:

$$\begin{aligned} f(x) &= \sum_t \big(f(\hat{x}_t) - f(\hat{x}_{t-1}) \big) \\ &= \sum_t x_t \nabla_t f(\hat{x}_{t-1}). \end{aligned}$$

Since $F(\{t\}) = f(1_t) = f(1_t) - f(0) = \nabla_t f(0)$, using the DR property of the function f, $\nabla_t f(0) \geq \nabla_t f(\hat{x}_{t-1})$ and therefore, we have:

$$f(x) \leq \sum_t x_t F(\{t\}). \tag{A.6}$$

Combining (A.5) and (A.6), we conclude:

$$\begin{aligned} \langle \nabla f(x), x \rangle &\geq (1 - \kappa_F) f(x) \\ \frac{\langle \nabla f(x), x \rangle}{f(x)} &\geq (1 - \kappa_F) \\ \alpha_f &\geq -\kappa_F. \end{aligned}$$

As a corollary, since $\alpha_f \in [-1, 0]$ and $\kappa_F \in [0, 1]$, if $\kappa_F = 0$ (i.e., F is modular), we can conclude that $\alpha_F = 0$ as well.

Example A.3.1 *Consider the Ising model with nonpositive pairwise interactions [26]:*

$$F(v) = \sum_{i \in [m]} \theta_i v_i + \sum_{i<j} \theta_{ij} v_i v_j.$$

where $v \in \{0,1\}^{\mathcal{V}}$, $|V| = m$, $\theta_i \geq 0$, $\theta_{ij} \leq 0$ *and* $\theta_i \geq \sum_{j:\ j<i} |\theta_{ji}| + \sum_{j:\ j>i} |\theta_{ij}|$ *for all* $i, j \in [m]$ *such that* $i \neq j$*. Under these assumptions, this class of functions are monotone submodular.*

The multilinear extension of this function could be easily derived as follows:

$$f(x) = \sum_{i \in n} \theta_i x_i + \sum_{i<j} \theta_{ij} x_i x_j.$$

The gradient of this function could be written in the following way:

$$\nabla_i f(x) = \theta_i + \sum_{j:\ j<i} \theta_{ji} x_j + \sum_{j:\ j>i} \theta_{ij} x_j \ \forall i \in [n].$$

As a simple example, consider $F(v) = 2v_1 + v_2 + 2v_3 - v_1 v_3$*, its multilinear extension* $f(x) = 2x_1 + x_2 + 2x_3 - x_1 x_3$ *and assume its corresponding linear packing constraint to be* $\mathcal{P} = \{x \in [0,1]^3 :\ 0.5x_1 + 0.6x_2 + 0.75x_3 \leq 1\}$.

For κ_F*, we have:*

$$\begin{aligned}
\kappa_F &= 1 - \min_{j:F(j)\neq 0} \frac{F(j|(V \setminus j))}{F(j)} \\
&= 1 - \min\{\frac{1}{2}, \frac{1}{1}, \frac{1}{2}\} \\
&= \frac{1}{2}.
\end{aligned}$$

α_f *could be computed as follows:*

$$\begin{aligned}
\alpha_f &= \inf_{x\in\mathcal{P}} \frac{\langle \nabla f(x), x\rangle}{f(x)} - 1 \\
&= \inf_{x\in\mathcal{P}} \frac{2x_1 + x_2 + 2x_3 - 2x_1x_3}{2x_1 + x_2 + 2x_3 - x_1x_3} - 1 \\
&= \inf_{x\in\mathcal{P}} \frac{-x_1x_3}{2x_1 + x_2 + 2x_3 - x_1x_3} \\
&= \frac{-1}{4}.
\end{aligned}$$

A.4 Missing proofs

A.4.1 Proof of Lemma 3.4.1

Considering that $\|x\|_2 \leq \lambda$ holds for all $x \in F_t$ and $t \in [m]$, we can write:

$$\begin{aligned}
P_{\text{gseq}} &= \sum_{i=1}^{n} \big(H_i(\omega_{im}(K)) + G_i((\hat{c}_i^T \omega_{im}(K))\big) \\
&= \sum_{i=1}^{n}\sum_{t=1}^{m} \big((H_i(\omega_{it}(K)) - H_i(\omega_{it}(0))\big) + \sum_{i=1}^{n}\sum_{t=1}^{m} \big(G_i(\hat{c}_i^T \omega_{it}(K)) - G_i \hat{c}_i^T \omega_{it}(0)))\big) \\
&= \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K} \big((H_i(\omega_{it}(k)) - H_i(\omega_{it}(k-1))\big) + \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K} \big(G_i((\hat{c}_i^T \omega_{it}(k)) - G_i(\hat{c}_i^T \omega_{it}(k-1)))\big)
\end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\geq} \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K}\big(\frac{1}{K}[v_t(k)]_i\nabla_t H_i(\omega_{it}(k-1)) - \frac{L}{2K^2}[v_t(k)]_i^2\big) \\
&+\sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K}\big(\frac{1}{K}c_{it}[v_t(k)]_i G_i'(\hat{c}_i^T\omega_{it}(k-1)) - \frac{Lc_{it}^2}{2K^2}[v_t(k)]_i^2\big) \\
&\overset{(b)}{\geq} \sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K}\big(\frac{1}{K}[v_t(k)]_i\nabla_t H_i(\omega_{it}(k-1))\big) - \frac{Lm\lambda^2}{2K} \\
&+\sum_{i=1}^{n}\sum_{t=1}^{m}\sum_{k=1}^{K}\big(\frac{1}{K}c_{it}[v_t(k)]_i G_i'(\hat{c}_i^T\omega_{it}(k-1))\big) - \frac{Lm\lambda^2}{2K} \\
&= \sum_{t=1}^{m}\sum_{k=1}^{K}\frac{1}{K}\langle v_t(k), d_t(k-1)\rangle - \frac{Lm\lambda^2}{K} \\
&\overset{(c)}{=} \sum_{t=1}^{m}\sum_{k=1}^{K}\frac{1}{K}\sigma_{F_t}(d_t(k-1)) - \frac{Lm\lambda^2}{K} \\
&\overset{(d)}{\geq} \sum_{t=1}^{m}\sigma_{F_t}(\frac{1}{K}\sum_{k=1}^{K}d_t(k-1)) - \frac{Lm\lambda^2}{K},
\end{aligned}
$$

where (a) is due to Assumption 1, (b) follows from $\|x\|_2 \leq \lambda \ \forall x \in F_t, t \in [m]$, (c) uses the update rule of the generalized sequential algorithm and (d) is a result of subadditivity of the support function σ_{F_t}.

By definition, for all $t \in [m], i \in [n]$ and $k \in \{0, \ldots, K\}$, we have $[d_t(k)]_i := \nabla_t H_i\big(\omega_{it}(k)\big) + c_{it}G_i'\big(\hat{c}_i^T\omega_{it}(k)\big)$. Since H_i is a continuous DR-submodular function, and that $\omega_{it}(k-1) \preceq \omega_{it}(K)$ holds, $\nabla_t H_i\big(\omega_{it}(K)\big) \leq \nabla_t H_i\big(\omega_{it}(k-1)\big)$ follows. Additionally, $G_i'\big(\hat{c}_i^T\omega_{it}(K)\big) \leq G_i'\big(\hat{c}_i^T\omega_{it}(k-1)\big)$ holds due to concavity of the penalty function G_i (equivalently, G_i' being non-increasing). Combining these results, we have:

$$
\begin{aligned}
d_t(K) &\preceq d_t(k-1) \\
d_t(K) &\preceq \frac{1}{K}\sum_{k=1}^{K}d_t(k-1) \\
x^T d_t(K) &\leq \frac{1}{K}x^T\sum_{k=1}^{K}d_t(k-1), \qquad \text{(A.7)}
\end{aligned}
$$

for $x \in F_t \subset \mathbb{R}^n_+$, i.e., x being element-wise non-negative. Taking supremum of (A.7) over all $x \in F_t$, we obtain:

$$\sigma_{F_t}(d_t(K)) \leq \sigma_{F_t}(\frac{1}{K}\sum_{k=1}^{K} d_t(k-1)).$$

Therefore, we have:

$$\text{ALG} \geq \sum_{t=1}^{m} \sigma_{F_t}(d_t(K)) - \sum_{i=1}^{n} G_i(\hat{c}_i^T \omega_{im}(K)) - \frac{Lm\lambda^2}{K}.$$

A.4.2 Proof of Lemma 3.4.2

First, note that by definition $\omega_{it}(0) = \omega_{i,t-1}(K)$ and $\omega_{i1}(0) = 0$ holds for all $t \in [m], i \in [n]$. Additionally, $H_i(0) = 0$ holds by assumption for all $i \in [n]$. Therefore, using the mean-value theorem, we can write:

$$\begin{aligned} H_i(\omega_{im}(K)) &= \sum_{t=1}^{m} \big(H_i(\omega_{it}(K)) - H_i(\omega_{it}(0))\big) \\ &= \sum_{t=1}^{m} \tilde{x}_{it} \nabla_t H_i(u_t), \end{aligned}$$

where $\tilde{x}_{it} = [\tilde{x}_t]_i = [\tilde{x}_t(K)]_i$, $u_t \in \mathbb{R}^m$ is the intermediate point in the mean-value theorem and $\omega_{it}(0) \preceq u_t \preceq \omega_{it}(K)$. Thus, we can write:

$$L_i \leq \min_{t\in[m]} \frac{\nabla_t H_i(u_t)}{c_{it}} \leq \frac{H_i\big(\omega_{im}(K)\big)}{\hat{c}_i^T \omega_{im}(K)} = \frac{\sum_{t=1}^{m} \tilde{x}_{it} \nabla_t H_i(u_t)}{\sum_{t=1}^{m} c_{it}\tilde{x}_{it}} \leq \max_{t\in[m]} \frac{\nabla_t H_i(u_t)}{c_{it}}.$$

Appendix B

APPENDIX OF CHAPTER 4

B.1 Other related work

Bandits with Knapsacks (BwK). This problem was introduced by [16] and its framework is as follows: there is a fixed set of n arms denoted by $\mathcal{X}$ and there are d resources being consumed. In each round $t \in [T]$, where T is a finite and known time horizon, an algorithm picks an arm $x_t \in \mathcal{X}$, and upon committing to this action, it receives a reward $r_t \in [0, 1]$ and consumes an amount $c_{t,i} \in [0, 1]$ of resource $i \in [d]$. There is a hard budget constraint $B_i \in \mathbb{R}_+$ for each resource $i \in [d]$ and the algorithm stops as soon as one or more budget constraint is violated. The overall reward of the algorithm equals the sum of rewards in all the rounds preceding the stopping time. The goal of the algorithm is to maximize the expected total reward. Depending on the input model for rewards and costs of the arms, the BwK has been classified into stochastic BwK and adversarial BwK categories. Our problem framework considers the adversarial input model for the case with one resource, i.e., $d = 1$. Nonetheless, the BwK problem is different from our setting in the following several important respects:

- The action set in the BwK problem is discrete and finite whereas we consider convex and compact domains $\mathcal{X} \subset \mathbb{R}^n_+$.
- The rewards in the BwK problem are linear in the arms. On the other hand, we consider a more general class of DR-submodular utility functions.
- The BwK problem only observes bandit feedback for the reward and resource consumption while we consider the full feedback setting.
- In the BwK problem, the budget constraints are strict and the algorithm stops as soon as one of the budget constraints is violated. However, in our setting, we allow budget violations

as long as the total budget violation is sub-linear in the time horizon T.

B.2 Motivating applications

In the following, we present several other applications that could be cast into our framework. These applications along with the online ad placement problem introduced in Section 4.2 show that the online continuous DR-submodular maximization problem with long-term budget constraints is indeed well-motivated.

Crowdsourcing markets. In this problem, there exists a requester with a limited budget B_T that submits jobs and benefits from them being completed. There are n types of jobs available to be assigned to workers arriving online. At each step $t \in [T]$, a worker arrives and the requester has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}^n_+ : 0 \preceq x \preceq 1\}$ of the jobs to the worker. The worker has an unknown private cost $[p_t]_i \ \forall i \in [n]$ for performing one unit of the i-th job where $[p_t]_i$ denotes the i-th entry of vector p_t. Therefore, the total cost of the job assignments to this worker equals $\langle p_t, x_t \rangle$. The reward obtained by the requester from this job assignment is a DR-submodular function $f_t(x_t)$. The DR property of the utility function captures the diminishing returns of assigning more jobs to the worker, i.e., as the number of assigned jobs to the worker increases, she has less time to devote to each fixed job $i \in [n]$ and therefore, the reward (quality of the completed task) obtained from the worker performing one unit of job i decreases. In other words, if $x \preceq y$, $\nabla_i f(x) \geq \nabla_i f(y) \ \forall i \in [n]$ holds. The goal is to maximize the overall rewards obtained by the requester while the budget constraint is not violated as well. Note that if the jobs are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [46], we allocate an integral bundle of jobs to the workers at each step.

Welfare maximization with production cost. In this problem, there is a seller who has n types of products for sale that may be produced on demand using a fixed limited budget B_T. At each step $t \in [T]$, an agent (customer) arrives online and the seller has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}^n_+ : 0 \preceq x \preceq 1\}$ of products to the agent. Producing each unit of each product $i \in [n]$ costs an unknown amount $[p_t]_i$ and the production cost of the item may change over time $\{1, \ldots, T\}$ because of the fluctuations of the prices of primitive resources. Therefore, the total production cost at step $t \in [T]$ equals $\langle p_t, x_t \rangle$. The agent has an unknown private DR-submodular valuation function f_t over the items where the DR property characterizes the diversity of the assigned bundle. Therefore, the utility obtained by assigning the bundle x_t equals $f_t(x_t)$. The goal is to maximize the overall valuation of the agents while satisfying the budget constraint of the seller. Note that if the products are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [46], we allocate an integral bundle of products to the agents at each step.

B.3 Missing proofs

B.3.1 Proof of Lemma 4.4.1

Since $\lambda_{t+1} = [\lambda_t - \mu \nabla_\lambda \mathcal{L}_t(x_t, \lambda_t)]_+ = [(1 - \delta\mu^2)\lambda_t + \mu g_t(x_t)]_+$ and $\lambda_1 = 0$, we have:

$$\begin{aligned}
\lambda_{t+1} &\geq (1 - \delta\mu^2)\lambda_t + \mu g_t(x_t) \\
&\geq (1 - \delta\mu^2)^2 \lambda_{t-1} + \mu g_t(x_t) + (1 - \delta\mu^2)\mu g_{t-1}(x_{t-1}) \\
&\geq \mu \sum_{s=1}^{t} (1 - \delta\mu^2)^{t-s} g_s(x_s) + (1 - \delta\mu^2)^t \underbrace{\lambda_1}_{=0} \\
&= \mu \sum_{s=1}^{t} (1 - \delta\mu^2)^{t-s} g_s(x_s).
\end{aligned}$$

Similarly, we can derive the other inequality as follows:

$$\begin{aligned}
\lambda_{t+1} &\leq |(1 - \delta\mu^2)\lambda_t + \mu g_t(x_t)| \\
&\leq (1 - \delta\mu^2)\lambda_t + \mu |g_t(x_t)| \\
&\leq (1 - \delta\mu^2)^2 \lambda_{t-1} + \mu |g_t(x_t)| + (1 - \delta\mu^2)\mu |g_{t-1}(x_{t-1})| \\
&\leq \mu \sum_{s=1}^{t} (1 - \delta\mu^2)^{t-s} |g_s(x_s)| + (1 - \delta\mu^2)^t \underbrace{\lambda_1}_{=0}
\end{aligned}$$

$$= \mu \sum_{s=1}^{t} (1-\delta\mu^2)^{t-s} |g_s(x_s)|.$$

B.3.2 Proof of Lemma 4.4.2

We have:

$$\begin{aligned}
\sum_{\tau=0}^{W-1} \lambda_{t+\tau} g_{t+\tau}(x_W^*) &= \sum_{\tau=0}^{W-1} \lambda_t g_{t+\tau}(x_W^*) + \sum_{\tau=0}^{W-1} \big(\lambda_{t+\tau} - \lambda_t\big) g_{t+\tau}(x_W^*) \\
&\le \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + \sum_{\tau=0}^{W-1} \big(|\lambda_{t+\tau} - \lambda_t|\big) |g_{t+\tau}(x_W^*)| \\
&\overset{\text{(a)}}{\le} \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + \sum_{\tau=0}^{W-1} \big(\sum_{s=1}^{\tau} \underbrace{|\lambda_{t+s} - \lambda_{t+s-1}|}_{(b)}\big) \underbrace{|g_{t+\tau}(x_W^*)|}_{\le G},
\end{aligned}$$

where (a) is due to the triangle inequality.

Using the result of Lemma 4.1, for all non-negative integers $r \ge 1$, we can write:

$$\begin{aligned}
\lambda_r &\le \mu \sum_{u=1}^{r-1} (1-\delta\mu^2)^{r-1-u} \underbrace{|g_u(x_u)|}_{\le G} \\
&\le \mu G \sum_{u=1}^{r-1} (1-\delta\mu^2)^{r-1-u} \\
&\le \mu G \sum_{u=0}^{\infty} (1-\delta\mu^2)^{u} \\
&= \frac{\mu G}{1-(1-\delta\mu^2)} \\
&= \frac{G}{\delta\mu}.
\end{aligned}$$

Consider the term (b). If $(1-\delta\mu^2)\lambda_{t+s-1} + \mu g_{t+s-1}(x_{t+s-1}) < 0$ (equivalently, $-\lambda_{t+s-1} \ge \frac{\mu}{1-\delta\mu^2} g_{t+s-1}(x_{t+s-1})$), we have:

$$\begin{aligned}
\text{(b)} &= |-\lambda_{t+s-1}| \\
&\le \frac{\mu}{1-\delta\mu^2} |g_{t+s-1}(x_{t+s-1})| \\
&\le \frac{\mu}{1-\delta\mu^2} G
\end{aligned}$$

$$\leq 2\mu G.$$

We will choose parameters δ and μ in our algorithm such that $\delta\mu^2 \ll 1$ holds. Otherwise, if $(1-\delta\mu^2)\lambda_{t+s-1} + \mu g_{t+s-1}(x_{t+s-1}) \geq 0$, we have:

$$\begin{aligned}
\text{(b)} &= |(1-\delta\mu^2)\lambda_{t+s-1} + \mu g_{t+s-1}(x_{t+s-1}) - \lambda_{t+s-1}| \\
&= |-\delta\mu^2\lambda_{t+s-1} + \mu g_{t+s-1}(x_{t+s-1})| \\
&\leq \delta\mu^2\lambda_{t+s-1} + \mu|g_{t+s-1}(x_{t+s-1})| \\
&\leq \delta\mu^2\frac{G}{\delta\mu} + \mu G \\
&= 2\mu G.
\end{aligned}$$

Therefore, we can write:

$$\begin{aligned}
\sum_{\tau=0}^{W-1} \lambda_{t+\tau} g_{t+\tau}(x_W^*) &\leq \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + \sum_{\tau=0}^{W-1} \big(\sum_{s=1}^{\tau} 2\mu G\big) G \\
&= \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + \sum_{\tau=0}^{W-1} 2\mu G^2 \tau \\
&= \lambda_t \sum_{\tau=0}^{W-1} g_{t+\tau}(x_W^*) + \mu G^2 W(W-1).
\end{aligned}$$

B.3.3 Proof of Lemma 4.4.3

Fix $k \in [K]$. Using L-smoothness of the function $\mathcal{L}_t$, we have:

$$\begin{aligned}
\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) &\geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_2^2 \\
&\overset{\text{(a)}}{\geq} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x_W^*\rangle + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x_W^*\rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x_W^*\rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x_W^*\rangle \\
&- \frac{1}{K}\lambda_t\langle \nabla g_t(x_t^{(k)}), x_W^*\rangle - \frac{LR^2}{2K^2} \\
&\overset{\text{(b)}}{=} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x_W^*\rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x_W^*\rangle
\end{aligned}$$

$$-\frac{1}{K}\lambda_t g_t(x_W^*) - \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2},$$

where (a) is due to the assumption that $\mathrm{diam}(\mathcal{X}) \leq R$. Note that in order to obtain (b), we have used linearity of the budget functions for all $t \in [T]$ to write $\langle \nabla g_t(x_t^{(k)}), x_W^* \rangle = \langle p_t, x_W^* \rangle = g_t(x_W^*) + \frac{B_T}{T}$. More general assumptions such as convexity would not be enough for the proof to go through.

Considering that $f_t(x)$ is monotone DR-submodular for all $t \in [T]$, we can write:

$$\begin{aligned} f_t(x_W^*) - f_t(x_t^{(k)}) &\overset{\text{(c)}}{\leq} f_t(x_W^* \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\ &\overset{\text{(d)}}{\leq} \langle \nabla f_t(x_t^{(k)}), (x_W^* \vee x_t^{(k)}) - x_t^{(k)} \rangle \\ &= \langle \nabla f_t(x_t^{(k)}), (x_W^* - x_t^{(k)}) \vee 0 \rangle \\ &\overset{\text{(e)}}{\leq} \langle \nabla f_t(x_t^{(k)}), x_W^* \rangle, \end{aligned}$$

where for $a, b \in \mathbb{R}^n$, $a \vee b$ denotes the entry-wise maximum of vectors a and b, (c) and (e) are due to the monotonicity of f_t and (d) uses concavity of f_t along non-negative directions. Therefore, we conclude:

$$\begin{aligned} \mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq{}& \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x_W^* \rangle + \frac{1}{K}\big(f_t(x_W^*) - f_t(x_t^{(k)})\big) \\ &- \frac{1}{K}\lambda_t g_t(x_W^*) - \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2}. \end{aligned}$$

Equivalently, we can write:

$$\begin{aligned} f_t(x_W^*) - f_t(x_t^{(k+1)}) \leq{}& (1 - \frac{1}{K})\big(f_t(x_W^*) - f_t(x_t^{(k)})\big) - \lambda_t\big(g_t(x_t^{(k+1)}) - g_t(x_t^{(k)})\big) + \frac{1}{K}\lambda_t g_t(x_W^*) \\ &+ \frac{1}{K}\lambda_t \frac{B_T}{T} + \frac{LR^2}{2K^2} + \frac{1}{K}\langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x_W^* - v_t^{(k)} \rangle \\ ={}& (1 - \frac{1}{K})\big(f_t(x_W^*) - f_t(x_t^{(k)})\big) + \frac{1}{K}\big[\lambda_t \frac{B_T}{T} - \lambda_t \langle p_t, v_t^{(k)} \rangle + \lambda_t g_t(x_W^*) \\ &+ \frac{LR^2}{2K} + \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x_W^* - v_t^{(k)} \rangle\big]. \end{aligned} \tag{B.1}$$

Replacing t by $t + \tau$ in inequality $(B.1)$ and taking the sum over $\tau \in \{0, \ldots, W-1\}$ and $t \in \{1, \ldots, T - W + 1\}$, we obtain:

$$\sum_{t=1}^{T-W+1} \sum_{\tau=0}^{W-1} \big(f_{t+\tau}(x_W^*) - f_{t+\tau}(x_{t+\tau}^{(k+1)})\big) \leq (1 - \frac{1}{K}) \sum_{t=1}^{T-W+1} \sum_{\tau=0}^{W-1} \big(f_{t+\tau}(x_W^*) - f_{t+\tau}(x_{t+\tau}^{(k)})\big)$$

$$+\frac{1}{K}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big[-\lambda_{t+\tau}\langle p_{t+\tau}, v_{t+\tau}^{(k)}\rangle+\lambda_{t+\tau}g_{t+\tau}(x_W^*)+\lambda_{t+\tau}\frac{B_T}{T}+\frac{LR^2}{2K}$$
$$+\langle\nabla\mathcal{L}_{t+\tau}(x_{t+\tau}^{(k)},\lambda_{t+\tau}), x_W^*-v_{t+\tau}^{(k)}\rangle\big]. \tag{B.2}$$

Applying inequality $(B.2)$ recursively for all $k\in\{1,\dots,K\}$, we obtain:

$$\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau}(x_W^*)-f_{t+\tau}(\underbrace{x_{t+\tau}^{(K+1)}}_{=x_{t+\tau}})\big)\le$$
$$\Pi_{k=0}^{K-1}(1-\frac{1}{K})\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau}(x_W^*)-f_{t+\tau}(x_{t+\tau}^{(0)})\big)$$
$$+\sum_{k=0}^{K-1}\frac{1}{K}\Pi_{j=k+1}^{K-1}(1-\frac{1}{K})\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big[-\lambda_{t+\tau}\langle p_{t+\tau}, v_{t+\tau}^{(k)}\rangle+\lambda_{t+\tau}g_{t+\tau}(x_W^*)+\lambda_{t+\tau}\frac{B_T}{T}+\frac{LR^2}{2K}$$
$$+\langle\nabla\mathcal{L}_{t+\tau}(x_{t+\tau}^{(k)},\lambda_{t+\tau}), x_W^*-v_{t+\tau}^{(k)}\rangle\big]. \tag{B.3}$$

Using the regret bound of Online Gradient Ascent instance $\mathcal{E}_k$ $\forall k\in[K]$, the following holds (Theorem 3.1. of [104]):

$$\begin{aligned}
\sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), x_W^*-v_t^{(k)}\rangle &= \sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), x_W^*\rangle-\sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), v_t^{(k)}\rangle\\
&\le \max_x\sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), x\rangle-\sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), v_t^{(k)}\rangle\\
&\le \frac{R^2}{\mu}+\frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)\|^2\\
&= \frac{R^2}{\mu}+\frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x f_t(x_t^{(k)})-\lambda_t p_t\|^2\\
&\overset{(a)}{\le} \frac{R^2}{\mu}+\frac{\mu}{2}\sum_{t=1}^{T}\big(2\|\nabla_x f_t(x_t^{(k)})\|^2+2\lambda_t^2\|p_t\|^2\big)\\
&\overset{(b)}{\le} \frac{R^2}{\mu}+\beta^2\mu T+\beta^2\mu\sum_{t=1}^{T}\lambda_t^2,
\end{aligned}$$

where (a) uses the inequality $\|a+b\|^2\le 2\|a\|^2+2\|b\|^2$ $\forall a,b\in\mathbb{R}^n$ and (b) is due to β-Lipschitzness of functions f_t, g_t for all $t\in[T]$.

Using the inequality $(1-\frac{1}{K})^K \le \frac{1}{e}$ in $(B.3)$, we have:

$$\begin{aligned}
\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau}(x_W^*) - f_{t+\tau}(x_{t+\tau})\big) &\le \frac{1}{e}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau}(x_W^*) - f_{t+\tau}(x_{t+\tau}^{(0)})\big)\\
&+ \sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\sum_{k=0}^{K-1}\frac{1}{K}\big[-\lambda_{t+\tau}\langle p_{t+\tau}, v_{t+\tau}^{(k)}\rangle + \lambda_{t+\tau}g_{t+\tau}(x_W^*)\\
&+ \lambda_{t+\tau}\frac{B_T}{T} + \frac{LR^2}{2K} + \langle \nabla\mathcal{L}_{t+\tau}(x_{t+\tau}^{(k)}, \lambda_{t+\tau}), x_W^* - v_{t+\tau}^{(k)}\rangle\big]\\
&= \frac{1}{e}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau}(x_W^*) - \underbrace{f_{t+\tau}(0)}_{=0}\big)\\
&+ \sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big[-\lambda_{t+\tau}g_{t+\tau}(x_{t+\tau}) - \lambda_{t+\tau}\frac{B_T}{T}\\
&+ \lambda_{t+\tau}g_{t+\tau}(x_W^*) + \lambda_{t+\tau}\frac{B_T}{T} + \frac{LR^2}{2K}\\
&+ \sum_{k=0}^{K-1}\frac{1}{K}\langle \nabla\mathcal{L}_{t+\tau}(x_{t+\tau}^{(k)}, \lambda_{t+\tau}), x_W^* - v_{t+\tau}^{(k)}\rangle\big]. \qquad \text{(B.4)}
\end{aligned}$$

Rearranging the terms in $(B.4)$, we obtain:

$$\begin{aligned}
&\underbrace{\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big((1-\frac{1}{e})f_{t+\tau}(x_W^*) - f_{t+\tau}(x_{t+\tau})\big)}_{\text{(a)}} + \underbrace{\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\lambda_{t+\tau}g_{t+\tau}(x_{t+\tau})}_{\text{(b)}} \le\\
&\frac{LR^2}{2K}W(T-W+1) + \underbrace{\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\lambda_{t+\tau}g_{t+\tau}(x_W^*)}_{\text{(c)}} + \sum_{k=0}^{K-1}\frac{1}{K}\underbrace{\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\langle \nabla L_{t+\tau}(x_{t+\tau}^{(k)}, \lambda_{t+\tau}), x_W^* - v_{t+\tau}^{(k)}\rangle}_{\text{(d)}}.
\end{aligned} \qquad \text{(B.5)}$$

(a) could be lower bounded as follows:

$$\begin{aligned}
\text{(a)} &= WR_T - \sum_{i=1}^{W-1}(W-i)\big([(1-\frac{1}{e})f_i(x_W^*) - f_i(x_i)] + [(1-\frac{1}{e})f_{T-i+1}(x_W^*) - f_{T-i+1}(x_{T-i+1})]\big)\\
&\ge WR_T - 2F\sum_{i=1}^{W-1}(W-i)\\
&= WR_T - FW(W-1). \qquad \text{(B.6)}
\end{aligned}$$

Using Lemma 4.1 with $(1-\delta\mu^2) \leq 1$, we have:

$$\begin{aligned}
\text{(b)} &= W\sum_{t=1}^{T}\lambda_t g_t(x_t) - \sum_{i=1}^{W-1}(W-i)\big(\lambda_i g_i(x_i) + \lambda_{T-i+1}g_{T-i+1}(x_{T-i+1})\big) \\
&\geq W\sum_{t=1}^{T}\lambda_t g_t(x_t) - \sum_{i=1}^{W-1}(W-i)\big(\mu(i-1)G^2 + \mu(T-i)G^2\big) \\
&\geq W\sum_{t=1}^{T}\lambda_t g_t(x_t) - \frac{G^2}{2}\mu W(W-1)(T-1). \qquad \text{(B.7)}
\end{aligned}$$

In order to bound (c), we use Lemma 4.2 and write:

$$\begin{aligned}
\text{(c)} &\leq \sum_{t=1}^{T-W+1}\big(\underbrace{\lambda_t\sum_{\tau=0}^{W-1}g_{t+\tau}(x^*_W)}_{\leq 0} + G^2\mu W(W-1)\big) \\
&\leq \mu G^2 W(W-1)(T-W+1). \qquad \text{(B.8)}
\end{aligned}$$

Finally, for a fixed $k \in [K]$, we can bound (d) as follows:

$$\begin{aligned}
\text{(d)} &= W\sum_{t=1}^{T}\langle\nabla\mathcal{L}_t(x_t^{(k)}), x^*_W - v_t^{(k)}\rangle - \sum_{i=1}^{W-1}(W-i)\big(\underbrace{[\langle\nabla\mathcal{L}_i(x_i^{(k)}), x^*_W - v_i^{(k)}\rangle]}_{\geq -\beta R(1+\lambda_i)} \\
&\quad + \underbrace{[\langle\nabla\mathcal{L}_{T-i+1}(x_{T-i+1}^{(k)}), x^*_W - v_{T-i+1}^{(k)}\rangle]}_{\geq -\beta R(1+\lambda_{T-i+1})}\big) \\
&\leq \frac{R^2W}{\mu} + \beta^2\mu TW + \beta^2\mu W\sum_{t=1}^{T}\lambda_t^2 + \sum_{i=1}^{W-1}(W-i)\big(2\beta R + \beta R\underbrace{\lambda_i}_{\leq(i-1)\mu G} + \beta R\underbrace{\lambda_{T-i+1}}_{\leq(T-i)\mu G}\big) \\
&= \frac{R^2W}{\mu} + \beta^2\mu TW + \beta^2\mu W\sum_{t=1}^{T}\lambda_t^2 + \beta RW(W-1) + \frac{\beta RG}{2}\mu W(W-1)(T-1). \qquad \text{(B.9)}
\end{aligned}$$

Using the regret bound for Online Gradient Descent (Theorem 3.1. of [104]), we have:

$$\begin{aligned}
\sum_{t=1}^{T}\big(\mathcal{L}_t(x_t,\lambda_t) - \mathcal{L}_t(x_t,\lambda)\big) &= \sum_{t=1}^{T}\big(-\lambda_t g_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2 + \lambda g_t(x_t) - \frac{\delta\mu}{2}\lambda^2\big) \\
&\leq \frac{\lambda^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_\lambda\mathcal{L}_t(x_t,\lambda_t)\|^2 \\
&\leq \frac{\lambda^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\big(-g_t(x_t) + \delta\mu\lambda_t\big)^2
\end{aligned}$$

$$\overset{\text{(a)}}{\leq} \frac{\lambda^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}(2g_t^2(x_t) + 2\delta^2\mu^2\lambda_t^2)$$
$$\leq \frac{\lambda^2}{\mu} + G^2\mu T + \delta^2\mu^3\sum_{t=1}^{T}\lambda_t^2, \tag{B.10}$$

where we use $(a+b)^2 \leq 2a^2 + 2b^2 \ \forall a, b \in \mathbb{R}$ to derive inequality (a).

Combining $(B.5)$, $(B.6)$, $(B.7)$, $(B.8)$, $(B.9)$ and $(B.10)$, dividing both sides by W and rearranging the terms, we conclude:

$$\begin{aligned} R_T + C_T\lambda + \frac{\delta\mu}{2}\sum_{t=1}^{T}\lambda_t^2 - \frac{\delta\mu}{2}T\lambda^2 - \frac{\lambda^2}{\mu} &\leq (F+\beta R)(W-1) + \frac{G}{2}(G+\beta R)\mu(W-1)(T-1) \\ &+ \frac{R^2}{\mu} + (G^2+\beta^2)\mu T + G^2\mu(W-1)(T-W+1) \\ &+ \frac{LR^2}{2K}(T-W+1) + (\delta^2\mu^3 + \beta^2\mu)\sum_{t=1}^{T}\lambda_t^2. \end{aligned}$$

Note that if T is large enough such that $WT \geq 16R^2$ holds, we can write:

$$\begin{aligned} \delta^2\mu^2 + \beta^2 &= 16\beta^4.\frac{R^2}{\beta^2 WT} + \beta^2 \\ &= \frac{16R^2}{WT}\beta^2 + \beta^2 \\ &\leq 2\beta^2 \\ &= \frac{\delta}{2}. \end{aligned}$$

Therefore, we can remove the terms $\sum_{t=1}^{T}\lambda_t^2$ from both sides of the inequality. Ignoring these terms, we obtain the desired result.

Appendix C

APPENDIX OF CHAPTER 5

C.1 Motivating applications

In the following, to illustrate the generality of our framework, we have listed several interesting applications that could be cast into our setting.

Online ad allocation. Consider the following online ad placement problem: At round $t \in [T]$, an advertiser should choose an investment vector $x_t \in R^n_+$ over n different websites where i-th entry of x_t denotes the amount that the advertiser is willing to pay per each click on the ad on the i-th website (i.e., cost per click). In other words, each website has different tiers of ads, and choosing x_t corresponds to ordering a certain type of ad. The aggregate cost of investment is determined when the number of clicks the ad receives is revealed. Namely, the cost of such an investment is characterized by p_t where the i-th entry of the vector p_t is the number of clicks the ad on the i-th website receives. The stochastic nature of the number of visitors to these n websites validates our choice of stochastic linear constraint functions. The advertiser needs to balance her total investment against an allotted long-term budget B_T. At round $t \in [T]$, the advertiser's utility function $f_t(x_t)$ is a monotone DR-submodular function with respect to the vector of investments and this function quantifies the overall impressions of the ads. DR-submodularity of the utility function characterizes the diminishing returns property of the impressions. In other words, making an ad more visible will attract proportionally fewer extra viewers because each website shares a portion of its visitors with other websites.

Online task assignment in crowdsourcing markets. In this problem, there exists a requester with a limited budget B_T that submits jobs and benefits from them being completed. There are n types of jobs available to be assigned to workers arriving online. At each step $t \in [T]$, a worker arrives and the requester has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}_+^n : 0 \preceq x \preceq 1\}$ of the jobs to the worker. The worker has a cost $[p_t]_i$ $\forall i \in [n]$ for performing one unit of the i-th job where $[p_t]_i$ denotes the i-th entry of vector p_t. The workers' evaluation of the cost of performing each of these n jobs is governed by the fluctuations of the wages in the job market and is stochastic. The reward obtained by the requester from this job assignment is a DR-submodular function $f_t(x_t)$. The DR property of the utility function captures the diminishing returns of assigning more jobs to the worker, i.e., as the number of assigned jobs to the worker increases, she has less time to devote to each fixed job $i \in [n]$ and therefore, the reward (quality of the completed task) obtained from the worker performing one unit of job i decreases. In other words, if $x \preceq y$, $\nabla_i f(x) \geq \nabla_i f(y)$ $\forall i \in [n]$ holds. The goal is to maximize the overall rewards obtained by the requester while the budget constraint is not violated as well. Note that if the jobs are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [46], we allocate an integral bundle of jobs to the workers at each step.

Online welfare maximization with production cost [108]. In this problem, there are n types of products for sale that may be produced on demand using a fixed limited budget B_T. At each step $t \in [T]$, an agent (customer) arrives online and the algorithm has to assign a bundle $x_t \in \mathcal{X} = \{x \in \mathbb{R}_+^n : 0 \preceq x \preceq 1\}$ of products to the agent. Producing each unit of product $i \in [n]$ costs an unknown amount $[p_t]_i$ and the production cost of the item may change over time $\{1, \ldots, T\}$ because of the stochastic fluctuations of the prices of ingredients. The agent has an unknown private DR-submodular valuation function f_t over the items where the DR property characterizes the diversity of the assigned bundle. Therefore, the utility obtained by assigning the bundle x_t equals $f_t(x_t)$. The goal is to maximize the overall valuation of the agents while satisfying the budget constraint. Note that if the products are indivisible, for all $t \in [T]$, the utility function f_t corresponds to the multilinear extension of the monotone submodular set function $F_t : 2^n \to \mathbb{R}$ and using the lossless pipage rounding technique of [46], we allocate an integral bundle of products to the agents at each step.

C.2 Lemma C.2.1

Lemma C.2.1 *[115] For all $t \in [T]$, the following holds:*

$$\mathbb{P}\{\|p_t - p\|_2 \geq \zeta\} \leq 2e^{-\frac{\zeta^2}{2\sigma^2}} \quad \forall \zeta \in \mathbb{R}.$$

Proof From assumption A3, we have that $\|p_t\| \leq \beta$. Thus, Lemma 1 of [115] holds with norm sub-gaussian parameter $\sigma = c\beta$ for some universal constant c. The result follows immediately. ■

C.3 Lemma C.3.1

Lemma C.3.1 *For $t = 2, 3, \ldots, T$, the following holds with probability at least $1 - \frac{\epsilon}{T}$:*

$$\|\widehat{p}_t - p\| \leq c'\sigma\sqrt{\frac{\log(\frac{2nT}{\epsilon})}{t-1}}.$$

Proof Note that $\|\widehat{p}_t - p\| = \frac{1}{t-1}\|\sum_{s=1}^{t-1}(p_s - p)\|$ and since vector $p_s - p$ satisfies the result of Lemma C.2.1, we can apply Corollary 7 of [115] to the random vectors $\{p_s - p\}_{s=1}^{t-1}$ and obtain:

$$\begin{aligned}\|\sum_{s=1}^{t-1}(p_s - p)\| &\leq c'\sqrt{\sum_{s=1}^{t-1}\sigma^2\log(\frac{2nT}{\epsilon})}\\ &= \sqrt{t-1}c'\sigma\sqrt{\log(\frac{2nT}{\epsilon})}.\end{aligned}$$

Combining the above equations, we get the desired result. ■

C.4 Missing proofs

C.4.1 Proof of Lemma 5.4.1

Fix $k \in [K]$. Using L-smoothness of the function $\mathcal{L}_t$, we have:

$$\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_2^2$$

$$
\begin{aligned}
&\overset{\text{(a)}}{\geq} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x \rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x \rangle \\
&- \frac{1}{K}\lambda_t \langle \nabla \widehat{g}_t(x_t^{(k)}), x \rangle - \frac{LR^2}{2K^2} \\
&\overset{\text{(b)}}{=} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x \rangle - \frac{1}{K}\lambda_t \widehat{g}_t(x) \\
&- \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2},
\end{aligned}
$$

where (a) is due to the assumption that $\operatorname{diam}(\mathcal{X}) \leq R$. Note that in order to obtain (b), we have used linearity of the budget functions for all $t \in [T]$ to write $\langle \nabla \widehat{g}_t(x_t^{(k)}), x \rangle = \langle \widehat{p}_t, x \rangle = \widehat{g}_t(x) + \frac{B_T}{T}$. More general assumptions such as convexity would not be enough for the proof to go through.

Considering that $f_t(x)$ is monotone DR-submodular for all $t \in [T]$, we can write:

$$
\begin{aligned}
f_t(x) - f_t(x_t^{(k)}) &\overset{\text{(c)}}{\leq} f_t(x \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\
&\overset{\text{(d)}}{\leq} \langle \nabla f_t(x_t^{(k)}), (x \vee x_t^{(k)}) - x_t^{(k)} \rangle \\
&= \langle \nabla f_t(x_t^{(k)}), (x - x_t^{(k)}) \vee 0 \rangle \\
&\overset{\text{(e)}}{\leq} \langle \nabla f_t(x_t^{(k)}), x \rangle,
\end{aligned}
$$

where for $a, b \in \mathbb{R}^n$, $a \vee b$ denotes the entry-wise maximum of vectors a and b, (c) and (e) are due to the monotonicity of f_t and (d) uses concavity of f_t along non-negative directions. Therefore, we conclude:

$$
\begin{aligned}
\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) &+ \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)} - x \rangle + \frac{1}{K}\big(f_t(x) - f_t(x_t^{(k)})\big) - \frac{1}{K}\lambda_t \widehat{g}_t(x) \\
&- \frac{1}{K}\lambda_t \frac{B_T}{T} - \frac{LR^2}{2K^2}.
\end{aligned}
$$

Equivalently, we can write:

$$
f_t(x) - f_t(x_t^{(k+1)}) \leq (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) - \lambda_t\big(\widehat{g}_t(x_t^{(k+1)}) - \widehat{g}_t(x_t^{(k)})\big)
$$

$$
\begin{aligned}
&+ \frac{1}{K}\lambda_t\widehat{g}_t(x) + \frac{1}{K}\lambda_t\frac{B_T}{T} + \frac{LR^2}{2K^2} + \frac{1}{K}\langle\nabla\mathcal{L}_t(x_t^{(k)},\lambda_t), x - v_t^{(k)}\rangle \\
&= (1-\frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) + \frac{1}{K}\big[\lambda_t\frac{B_T}{T} - \lambda_t\langle\widehat{p}_t, v_t^{(k)}\rangle + \lambda_t\widehat{g}_t(x) \\
&+ \frac{LR^2}{2K} + \langle\nabla\mathcal{L}_t(x_t^{(k)},\lambda_t), x - v_t^{(k)}\rangle\big].
\end{aligned}
$$

Taking the sum over $t \in \{1, \dots, T\}$, we obtain:

$$
\begin{aligned}
\sum_{t=1}^{T}\big(f_t(x) - f_t(x_t^{(k+1)})\big) \leq (1-\frac{1}{K})&\sum_{t=1}^{T}\big(f_t(x) - f_t(x_t^{(k)})\big) \\
&+ \frac{1}{K}\sum_{t=1}^{T}\big[-\lambda_t\langle\widehat{p}_t, v_t^{(k)}\rangle + \lambda_t\widehat{g}_t(x) + \lambda_t\frac{B_T}{T} + \frac{LR^2}{2K} \\
&+ \langle\nabla\mathcal{L}_t(x_t^{(k)},\lambda_t), x - v_t^{(k)}\rangle\big]. \qquad (C.1)
\end{aligned}
$$

Applying inequality $(C.1)$ recursively for all $k \in \{1, \dots, K\}$, we obtain:

$$
\begin{aligned}
\sum_{t=1}^{T}\big(f_t(x) - f_t(\underbrace{x_t^{(K+1)}}_{=x_t})\big) \leq \Pi_{k=0}^{K-1}(1-\frac{1}{K})&\sum_{t=1}^{T}\big(f_t(x) - f_t(x_t^{(0)})\big) \\
+ \sum_{k=0}^{K-1}\frac{1}{K}\Pi_{j=k+1}^{K-1}(1-\frac{1}{K})&\sum_{t=1}^{T}\big[-\lambda_t\langle\widehat{p}_t, v_t^{(k)}\rangle + \lambda_t\widehat{g}_t(x) + \lambda_t\frac{B_T}{T} \\
+ \frac{LR^2}{2K} + \langle\nabla\mathcal{L}_t(x_t^{(k)},\lambda_t), x - v_t^{(k)}\rangle\big]. \qquad (C.2)
\end{aligned}
$$

Using the regret bound of Online Gradient Ascent instance $\mathcal{E}_k$ $\forall k \in [K]$, the following holds:

$$
\begin{aligned}
\sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), x - v_t^{(k)}\rangle &= \sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), x\rangle - \sum_{t=1}^{T}\langle\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t), v_t^{(k)}\rangle \\
&\leq \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)\|^2 \\
&= \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\|\nabla_x f_t(x_t^{(k)}) - \lambda_t\widehat{p}_t\|^2 \\
&\overset{(a)}{\leq} \frac{R^2}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\big(2\|\nabla_x f_t(x_t^{(k)})\|^2 + 2\lambda_t^2\|\widehat{p}_t\|^2\big) \\
&\overset{(b)}{\leq} \frac{R^2}{\mu} + \beta^2\mu T + \beta^2\mu\sum_{t=1}^{T}\lambda_t^2,
\end{aligned}
$$

where (a) uses the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2 \ \forall a, b \in \mathbb{R}^n$ and (b) is due to β-Lipschitzness of functions f_t, g_t for all $t \in [T]$.

Using the inequality $(1 - \frac{1}{K})^K \leq \frac{1}{e}$ in $(C.2)$, we have:

$$\begin{aligned}
\sum_{t=1}^{T} \big(f_t(x) - f_t(x_t)\big) &\leq \frac{1}{e} \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(0)})\big) + \sum_{t=1}^{T} \sum_{k=0}^{K-1} \frac{1}{K} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) \\
&\quad + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} + \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)} \rangle \big] \\
&= \frac{1}{e} \sum_{t=1}^{T} \big(f_t(x_t^*) - \underbrace{f_t(0)}_{=0}\big) + \sum_{t=1}^{T} \big[- \lambda_t \widehat{g}_t(x_t) - \lambda_t \frac{B_T}{T} + \lambda_t \widehat{g}_t(x) \\
&\quad + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} + \sum_{k=0}^{K-1} \frac{1}{K} \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)} \rangle \big]. \qquad \text{(C.3)}
\end{aligned}$$

Rearranging the terms in $(C.3)$, we obtain:

$$\begin{aligned}
\sum_{t=1}^{T} \big((1 - \frac{1}{e}) f_t(x) - f_t(x_t)\big) + \sum_{t=1}^{T} \lambda_t \widehat{g}_t(x_t) &\leq \frac{LR^2 T}{2K} + \sum_{t=1}^{T} \lambda_t \widehat{g}_t(x) + \sum_{k=0}^{K-1} \frac{1}{K} \sum_{t=1}^{T} \langle \nabla \mathcal{L}_t(x_t^{(k)}, \lambda_t), x - v_t^{(k)} \rangle \\
&\leq \frac{LR^2 T}{2K} + \sum_{t=1}^{T} \lambda_t \widehat{g}_t(x) + \frac{R^2}{\mu} + \beta^2 \mu T + \beta^2 \mu \sum_{t=1}^{T} \lambda_t^2. \qquad \text{(C.4)}
\end{aligned}$$

Now, subtract $\sum_{t=1}^{T} \lambda_t \gamma_t$ from each side of the inequality $(C.4)$. Thus, obtain that:

$$\sum_{t=1}^{T} \big((1 - \frac{1}{e}) f_t(x) - f_t(x_t)\big) \leq \frac{LR^2 T}{2K} + \frac{R^2}{\mu} + \beta^2 \mu T + \sum_{t=1}^{T} \lambda_t \widehat{g}_t(x) + \big[\beta^2 \mu \sum_{t=1}^{T} \lambda_t^2 - \sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x_t)\big].$$

Setting $\delta = \beta^2$ and using the update rule of the algorithm for $\lambda_t \ \forall t \in [T]$, we have:

$$\begin{aligned}
\beta^2 \mu \sum_{t=1}^{T} \lambda_t^2 - \sum_{t=1}^{T} \lambda_t \widetilde{g}_t(x_t) &= \frac{\beta^2 \mu}{\delta^2 \mu^2} \sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+^2 - \frac{1}{\delta \mu} \sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ \widetilde{g}_t(x_t) \\
&\overset{\text{(a)}}{\leq} \frac{1}{\delta \mu} \sum_{t=1}^{T} \big(\widetilde{g}_t^2(x_t) - \widetilde{g}_t(x_t) \widetilde{g}_t(x_t)\big) \\
&\leq 0,
\end{aligned}$$

where we have used $[\widetilde{g}_t(x_t)]_+ \geq \widetilde{g}_t(x_t)$ and $|[\widetilde{g}_t(x_t)]_+| \leq \widetilde{g}_t(x_t)$ to obtain (a).

C.4.2 Proof of Lemma 5.4.2

Consider the events $\mathcal{E}_t := \{\|\widehat{p}_t - p\| \leq c'\sigma\sqrt{\log(\frac{2nT}{\epsilon})/(t-1)}\}$ for $t \in [T]/\{1\}$ and $\mathcal{E} := \{\sum_{t=2}^{T} \|\widehat{p}_t - p\| \leq C\sigma\sqrt{T\log(2nT/\epsilon)}\}$. Then, apply union bound to $\bigcup_{t=2}^{T} \mathcal{E}_t^c$ with the observation that $\bigcap_{t=2}^{T} \mathcal{E}_t \subset \mathcal{E}$. Explicitly,

$$\begin{aligned}
\mathbb{P}(\bigcup_{t=2}^{T} \mathcal{E}_t) &\leq \sum_{t=2}^{T} \mathbb{P}(\mathcal{E}_t^c), \\
1 - \mathbb{P}(\bigcap_{t=2}^{T} \mathcal{E}_t) &\leq \sum_{t=2}^{T} (1 - \mathbb{P}(\mathcal{E}_t)), \\
1 - \sum_{t=2}^{T} (1 - \mathbb{P}(\mathcal{E}_t)) \leq \mathbb{P}(\bigcap_{t=2}^{T} \mathcal{E}_t) &\leq \mathbb{P}(\mathcal{E}).
\end{aligned}$$

Using the result of Lemma C.3.1, we obtain the result.

C.4.3 Proof of Lemma 5.4.4

Bounding $\sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+$ from below, we obtain:

$$\begin{aligned}
\sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ &\geq \sum_{t=1}^{T} \widetilde{g}_t(x_t) \\
&= \sum_{t=1}^{T} g(x_t) + \sum_{t=1}^{T} (\widehat{g}_t(x_t) - g(x_t)) - \sum_{t=1}^{T} \gamma_t \\
&\geq \sum_{t=1}^{T} g(x_t) - \sum_{t=1}^{T} |\widehat{g}_t(x_t) - g(x_t)| - \sum_{t=1}^{T} \gamma_t \\
&= C_T - \sum_{t=1}^{T} |\langle \widehat{p}_t - p, x_t \rangle| - \sum_{t=1}^{T} \gamma_t \\
&\geq C_T - \sum_{t=1}^{T} \|\widehat{p}_t - p\| \|x_t\| - \sum_{t=1}^{T} \gamma_t \\
&\geq C_T - R \sum_{t=1}^{T} \|\widehat{p}_t - p\| - \sum_{t=1}^{T} \gamma_t.
\end{aligned}$$

Rearranging the above inequality, we obtain the desired result.

C.4.4 Proof of Lemma 5.4.5

We can write:

$$\begin{aligned}
\mathbb{E}\|\widehat{p}_t - p\|^2 &= \mathbb{E}(\widehat{p}_t - p)^T(\widehat{p}_t - p) \\
&= \mathbb{E}[Tr\big((\widehat{p}_t - p)^T(\widehat{p}_t - p)\big)] \\
&= \mathbb{E}[Tr\big((\widehat{p}_t - p)(\widehat{p}_t - p)^T\big)] \\
&= Tr\big(\mathbb{E}[(\widehat{p}_t - p)(\widehat{p}_t - p)^T]\big) \\
&= Tr(Cov(\widehat{p}_t)) \\
&= Tr(\frac{\Sigma}{t-1}) \\
&= \frac{Tr(\Sigma)}{t-1}.
\end{aligned}$$

C.4.5 Proof of Theorem 5.4.4

Similar to the proof of Theorem 2, we begin by setting $x = 0$ to be the fixed vector in (2). We obtain

$$\sum_{t=1}^{T} [\widetilde{g}_t(x_t)]_+ \le \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta).$$

Now, we lower bound the left-hand side following the idea of the proof for Lemma 4. Thus, we obtain

$$C_T \le \frac{T}{B_T} R\beta F\sqrt{T} + \frac{TR\beta}{B_T}(\frac{LR^2}{2} + 2R\beta) + \underbrace{R\sum_{t=1}^{T} \|\widehat{p}_t - p\|}_{\text{(C)}}.$$

In order to bound (C), we take expectation on both sides to obtain:

$$\begin{aligned}
\mathbb{E}[\text{(C)}] &= R\mathbb{E}[\sum_{t=2}^{T} \|\widehat{p}_t - p\|] \\
&= R\sum_{t=2}^{T} \mathbb{E}\|\widehat{p}_t - p\| \\
&= R\sum_{t=2}^{T} \mathbb{E}\sqrt{\|\widehat{p}_t - p\|^2}
\end{aligned}$$

$$\leq R \sum_{t=2}^{T} \sqrt{\mathbb{E}\|\widehat{p}_t - p\|^2},$$

where the last inequality is due to Jensen's inequality.

Therefore, we have:

$$\mathbb{E}[(\mathrm{C})] \leq R \sum_{t=2}^{T} \frac{\sqrt{Tr(\Sigma)}}{\sqrt{t-1}}$$
$$\leq R\sqrt{Tr(\Sigma)}\sqrt{T}.$$

Combining the inequalities, we obtain the result.

C.5 Relation with previous results

- [213] studied a similar problem in the convex setting where all the constraint functions are deterministic and given offline and they obtained $\mathcal{O}(\sqrt{T})$ regret bound and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bound. On the other hand, applying our OLFW algorithm with update rule (I) to the online DR-submodular maximization problem subject to deterministic linear constraints, we obtain $\mathcal{O}(\sqrt{T})$ regret and constraint violation bounds simultaneously. Note that in this setting, $\Sigma = 0$, and thus, the dominating $\mathcal{O}(T^{\frac{3}{4}})$ term in the expected regret bound of the OLFW algorithm vanishes.
- [114] considered a related problem with concave utility functions and linear constraint functions where at each round, the reward functions arrive before committing to an action (i.e., the 1-lookahead setting) and long-term constraints are penalized through a penalty function in the objective. They obtained sub-linear bounds for the dynamic regret in their setting. On the contrary, in our framework, we deal with the extra complication that the utility function at each step is DR-submodular (and generally non-concave) and it is revealed after committing to an action.
- [142] considered the same framework as ours in the special case where the utility functions are linear and they obtained $\mathcal{O}(\sqrt{T})$ regret bound and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bound in both expectation and high probability settings. Note that our OLFW algorithm obtains improved $\mathcal{O}(\sqrt{T})$ constraint violation bounds in expectation and with high probability for

the more general setting of DR-submodular utility functions.

- [181] considered the online DR-submodular maximization problem subject to linear constraints in the adversarial setting where the constraint functions are chosen arbitrarily. Their proposed OSPHG algorithm in the setting with window length $W = 1$ could be adapted to obtain $\mathcal{O}(\sqrt{T})$ regret and $\mathcal{O}(T^{\frac{3}{4}})$ constraint violation bounds in expectation. However, the OSPHG algorithm fails to provide *any* bounds for the high probability setting. On the other hand, our algorithm uses the current estimate of the cost vector to exploit the stochastic nature of the constraints. Furthermore, through using the update rule (II) in the OLFW algorithm, we can guarantee sub-linear bounds in the high probability setting as well.

C.6 Modified Online Lagrangian Frank-Wolfe (MOLFW) algorithm

For the setting where only unbiased stochastic gradient estimates of the utility functions $\{f_t\}_{t=1}^T$ with bounded variance σ^2 are available, inspired by the variance reduction technique introduced by [60], we propose the Modified Online Lagrangian Frank-Wolfe (MOLFW) algorithm in Algorithm 2. Compared to the OLFW algorithm, the MOLFW algorithm uses $\langle v_t^{(k)}, d_t^{(k)}\rangle$ (instead of $\langle v_t^{(k)}, \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\rangle$) as the payoff to be received by $\mathcal{E}_k$ $\forall k \in [K]$ at round $t \in [T]$. We analyze the performance of the MOLFW algorithm below.

Fix $k \in [K]$. Using L-smoothness of the function $\mathcal{L}_t$, we have:

$$\begin{aligned}
\mathcal{L}_t(x_t^{(k+1)}, \lambda_t) &\geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_2^2 \\
&\overset{\text{(a)}}{\geq} \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), v_t^{(k)}\rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x\rangle \\
&+ \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t), x\rangle + \frac{1}{K}\langle d_t^{(k)}, v_t^{(k)} - x\rangle - \frac{LR^2}{2K^2} \\
&= \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x\rangle \\
&+ \frac{1}{K}\langle \nabla f_t(x_t^{(k)}), x\rangle - \frac{1}{K}\lambda_t\langle \widehat{p}_t, x\rangle + \frac{1}{K}\langle d_t^{(k)}, v_t^{(k)} - x\rangle - \frac{LR^2}{2K^2},
\end{aligned}$$

where (a) is due to the assumption that $\operatorname{diam}(\mathcal{X}) \leq R$. Considering that $f_t(x)$ is monotone DR-submodular for all $t \in [T]$, we can write:

$$\begin{aligned} f_t(x) - f_t(x_t^{(k)}) &\overset{\text{(b)}}{\leq} f_t(x \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\ &\overset{\text{(c)}}{\leq} \langle \nabla f_t(x_t^{(k)}), (x \vee x_t^{(k)}) - x_t^{(k)} \rangle \\ &= \langle \nabla f_t(x_t^{(k)}), (x - x_t^{(k)}) \vee 0 \rangle \\ &\overset{\text{(d)}}{\leq} \langle \nabla f_t(x_t^{(k)}), x \rangle, \end{aligned}$$

where for $a, b \in \mathbb{R}^n$, $a \vee b$ denotes the entry-wise maximum of vectors a and b, (b) and (d) are due to the monotonicity of f_t and (c) uses concavity of f_t along non-negative directions. Therefore, we conclude:

$$\begin{aligned} \mathcal{L}_t(x_t^{(k+1)}, \lambda_t) \geq \mathcal{L}_t(x_t^{(k)}, \lambda_t) + \frac{1}{K}\big(\langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle + f_t(x) - f_t(x_t^{(k)}) - \lambda_t \widehat{g}_t(x) \\ - \lambda_t \frac{B_T}{T} + \langle d_t^{(k)}, v_t^{(k)} - x \rangle\big) - \frac{LR^2}{2K^2}. \end{aligned}$$

Using the Young's inequality, we have:

$$\begin{aligned} \langle \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}, v_t^{(k)} - x \rangle &\geq -\frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 - \frac{\beta_k}{2} \|v_t^{(k)} - x\|^2 \\ &\geq -\frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 - \frac{R^2 \beta_k}{2}. \end{aligned}$$

Therefore, we can equivalently write:

$$\begin{aligned} f_t(x) - f_t(x_t^{(k+1)}) &\leq (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) - \lambda_t\big(\widehat{g}_t(x_t^{(k+1)}) - \widehat{g}_t(x_t^{(k)})\big) \\ &\quad + \frac{1}{K} \lambda_t \widehat{g}_t(x) + \frac{1}{K} \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K^2} + \frac{1}{K} \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{1}{K} \frac{R^2 \beta_k}{2} \\ &\quad + \frac{1}{K} \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \\ &= (1 - \frac{1}{K})\big(f_t(x) - f_t(x_t^{(k)})\big) + \frac{1}{K}\big[\lambda_t \frac{B_T}{T} - \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) \\ &\quad + \frac{LR^2}{2K} + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2\big]. \end{aligned}$$

Taking the sum over $t \in \{1, \ldots, T\}$, we obtain:

$$\sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k+1)})\big) \leq (1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(k)})\big) + \frac{1}{K} \sum_{t=1}^{T} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x)$$

$$+ \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \big]. \tag{C.5}$$

Applying inequality $(C.5)$ recursively for all $k \in \{1, \ldots, K\}$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} \big(f_t(x) - f_t(\underbrace{x_t^{(K+1)}}_{=x_t})\big) &\leq \Pi_{k=0}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(0)})\big) \\ &+ \sum_{k=0}^{K-1} \frac{1}{K} \Pi_{j=k+1}^{K-1}(1 - \frac{1}{K}) \sum_{t=1}^{T} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} \\ &+ \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \big]. \end{aligned} \tag{C.6}$$

Using the regret bound of Online Gradient Ascent instance $\mathcal{E}_k$ $\forall k \in [K]$, the following holds:

$$\begin{aligned}\sum_{t=1}^{T} \langle d_t^{(k)}, x - v_t^{(k)} \rangle &= \sum_{t=1}^{T} \langle d_t^{(k)}, x \rangle - \sum_{t=1}^{T} \langle d_t^{(k)}, v_t^{(k)} \rangle \\ &\leq \frac{R^2}{\mu} + \frac{\mu}{2} \sum_{t=1}^{T} \|d_t^{(k)}\|^2 \\ &\leq \frac{R^2}{\mu} + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)\|^2 \\ &\overset{(a)}{\leq} \frac{R^2}{\mu} + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 + \mu \sum_{t=1}^{T} \big(2\|\nabla f_t(x_t^{(k)})\|^2 + 2\lambda_t^2 \|\widehat{p}_t\|^2\big) \\ &\overset{(b)}{\leq} \frac{R^2}{\mu} + 2\beta^2 \mu T + 2\beta^2 \mu \sum_{t=1}^{T} \lambda_t^2 + \mu \sum_{t=1}^{T} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2, \end{aligned}$$

where (a) uses the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ $\forall a, b \in \mathbb{R}^n$ and (b) is due to β-Lipschitzness of functions f_t, g_t for all $t \in [T]$.

Using the inequality $(1 - \frac{1}{K})^K \leq \frac{1}{e}$ in $(C.6)$, we have:

$$\begin{aligned}\sum_{t=1}^{T} \big(f_t(x) - f_t(x_t)\big) &\leq \frac{1}{e} \sum_{t=1}^{T} \big(f_t(x) - f_t(x_t^{(0)})\big) + \sum_{t=1}^{T} \sum_{k=0}^{K-1} \frac{1}{K} \big[- \lambda_t \langle \widehat{p}_t, v_t^{(k)} \rangle + \lambda_t \widehat{g}_t(x) \\ &+ \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} + \langle d_t^{(k)}, x - v_t^{(k)} \rangle + \frac{R^2 \beta_k}{2} + \frac{1}{2\beta_k} \|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2 \big] \\ &= \frac{1}{e} \sum_{t=1}^{T} \big(f_t(x) - \underbrace{f_t(0)}_{=0}\big) + \sum_{t=1}^{T} \big[- \lambda_t \widehat{g}_t(x_t) - \lambda_t \frac{B_T}{T} + \lambda_t \widehat{g}_t(x) + \lambda_t \frac{B_T}{T} + \frac{LR^2}{2K} \rangle \end{aligned}$$

$$+\sum_{k=0}^{K-1}\frac{1}{K}\big(\langle d_t^{(k)}, x-v_t^{(k)}+\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)\big]. \tag{C.7}$$

Rearranging the terms in $(C.7)$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big)+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x_t) &\le \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)+\sum_{k=0}^{K-1}\frac{1}{K}\big(\langle d_t^{(k)}, x-v_t^{(k)}\rangle\\ &\quad+\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big)\\ &\le \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widehat{g}_t(x)+\frac{R^2}{\mu}+2\beta^2\mu T+2\beta^2\mu\sum_{t=1}^{T}\lambda_t^2\\ &\quad+\sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big(\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\\ &\quad+\mu\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big).\end{aligned} \tag{C.8}$$

Now, subtract $\sum_{t=1}^{T}\lambda_t\gamma_t$ from each side of the inequality $(C.8)$. Thus, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big((1-\frac{1}{e})f_t(x)-f_t(x_t)\big) &\le \frac{LR^2T}{2K}+\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x)+\frac{R^2}{\mu}+2\beta^2\mu T+\big[2\beta^2\mu\sum_{t=1}^{T}\lambda_t^2-\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x_t)\big]\\ &\quad+\sum_{t=1}^{T}\sum_{k=0}^{K-1}\frac{1}{K}\big(\frac{R^2\beta_k}{2}+\frac{1}{2\beta_k}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\\ &\quad+\mu\sum_{t=1}^{T}\|\nabla_x\mathcal{L}_t(x_t^{(k)},\lambda_t)-d_t^{(k)}\|^2\big).\end{aligned}$$

Setting $\delta=2\beta^2$ and using the update rule of the algorithm for $\lambda_t\ \forall t\in[T]$, we have:

$$\begin{aligned}2\beta^2\mu\sum_{t=1}^{T}\lambda_t^2-\sum_{t=1}^{T}\lambda_t\widetilde{g}_t(x_t) &= \frac{2\beta^2\mu}{\delta^2\mu^2}\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+^2-\frac{1}{\delta\mu}\sum_{t=1}^{T}[\widetilde{g}_t(x_t)]_+\widetilde{g}_t(x_t)\\ &\overset{(a)}{\le} \frac{1}{\delta\mu}\sum_{t=1}^{T}\big(\widetilde{g}_t^2(x_t)-\widetilde{g}_t(x_t)\widetilde{g}_t(x_t)\big)\\ &\le 0,\end{aligned}$$

where we have used $[\widetilde{g}_t(x_t)]_+ \geq \widetilde{g}_t(x_t)$ and $|[\widetilde{g}_t(x_t)]_+| \leq \widetilde{g}_t(x_t)$ to obtain (a).

Using Theorem 3 of [60], we have:

$$\mathbb{E}\big[\|\nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t) - d_t^{(k)}\|^2\big] \leq \frac{Q}{(k+4)^{2/3}} \ \forall t \in [T], k \in [K],$$

where $Q = \max\{\max_{t\in[T]} \|\nabla f_t(0)\|^2 4^{2/3}, 4\sigma^2 + 6L^2R^2\}$. Therefore, taking expectation of both sides, setting $\beta_k = \frac{Q^{1/2}}{R(k+4)^{1/3}}$ and simplifying the result, we obtain:

$$\mathbb{E}[R_T] \leq \frac{R^2}{\mu} + 2\beta^2\mu T + \frac{LR^2T}{2K} + \sum_{t=1}^{T} \mathbb{E}[\lambda_t \widetilde{g}_t(x)] + \frac{3Q\mu T}{K^{2/3}} + \frac{3RQ^{1/2}T}{2K^{1/3}}.$$

Therefore, if we set $K = \mathcal{O}(T^{3/2})$, we can analyze the expected regret and constraint violation of the MOLFW algorithm similarly to Theorem 5.4.3 and Theorem 5.4.4, and we obtain similar performance bounds.

Algorithm 2 Modified Online Lagrangian Frank-Wolfe (MOLFW)

```
**Input:** $\mathcal{X}$ is the constraint set, $T$ is the horizon, $\mu > 0$, $\delta > 0$, $\{\gamma_t\}_{t=1}^{T}$, $\{\rho_k\}_{k=1}^{K}$ and $K$.
**Output:** $\{x_t : 1 \leq t \leq T\}$.
Initialize $K$ instances $\{\mathcal{E}_k\}_{k\in[K]}$ of Online Gradient Ascent with step size $\mu$ for online maximization of linear functions over $\mathcal{X}$.
for $t = 1$ to $T$ do
  $x_t^{(1)} = 0$.
  for $k = 1$ to $K$ do
    Let $v_t^{(k)}$ be the output of oracle $\mathcal{E}_k$ from round $t-1$.
    $x_t^{(k+1)} = x_t^{(k)} + \frac{1}{K} v_t^{(k)}$.
  end for
  Set $x_t = x_t^{(K+1)}$.
  Let $\widehat{p}_t := \frac{1}{t-1}\sum_{s=1}^{t-1} p_s$ for $t > 1$.
  Let
    $$\widetilde{g}_t(\cdot) = \begin{cases} \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} & \text{expectation analysis (I)} \\ \langle \widehat{p}_t, \cdot \rangle - \frac{B_T}{T} - \gamma_t & \text{high probability analysis (II)} \end{cases}$$
  Set $\lambda_t = \frac{[\widetilde{g}_t(x_t)]_+}{\delta\mu}$ for $t > 1$ and 0 otherwise.
  Play $x_t$ and observe the Lagrangian function $\mathcal{L}_t(x_t, \lambda_t) = f_t(x_t) - \lambda_t \widetilde{g}_t(x_t) + \frac{\delta\mu}{2}\lambda_t^2$
  $d_t^{(0)} = -\lambda_t \widehat{p}_t$.
  for $k = 1$ to $K$ do
    $d_t^{(k)} = (1-\rho_k) d_t^{(k-1)} + \rho_k \nabla_x \mathcal{L}_t(x_t^{(k)}, \lambda_t)$.
    Feedback $\langle v_t^{(k)}, d_t^{(k)} \rangle$ as the payoff to be received by $\mathcal{E}_k$.
  end for
end for
```

Appendix D

APPENDIX OF CHAPTER 6

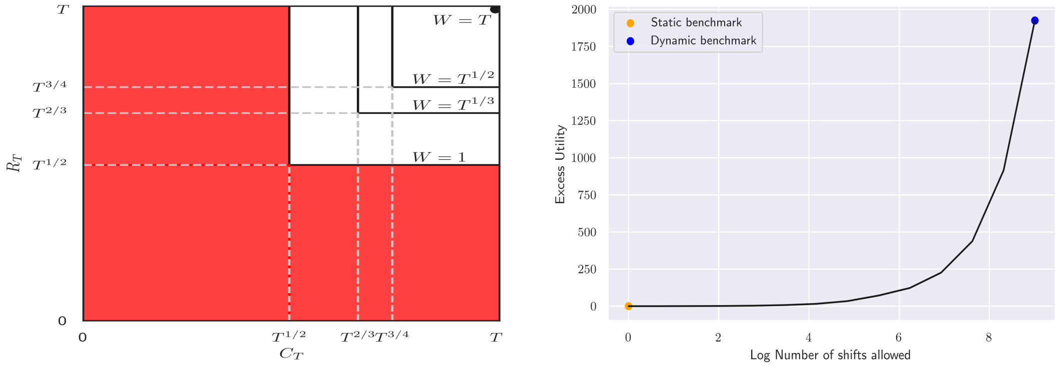
D.1 Additional plots

Figure D.1(a) demonstrates the achievable adversarial static regret and total constraint violation by Algorithm 1 for different choices of the window length W. The black curves indicate the Pareto frontier for different values of W and all the points north-east of the frontier are achievable. In particular, we can observe that for $W = T$, both the regret and total constraint violation of Algorithm 1 are linear.

In Figure D.1(b), we have plotted the excess overall utility of the shifting benchmark sequence [106] with respect to the static benchmark action for different number of allowed shifts for experiment 2. In other words, we consider the dynamic benchmark sequence $\{x_t^*\}_{t=1}^T$ which changes at most m times (i.e., $x_t^* \neq x_{t+1}^*$ for at most m values of $t \in \{1, \ldots, T-1\}$) and we plot the excess overall utility for different choices of m. Note that the adversarial static and dynamic regret defined in the paper correspond to $m = 0$ and $m = T-1$ respectively. Figure D.1(b) verifies that the dynamic benchmark sequence achieves much higher utility compared to the static benchmark action and therefore, for the settings where the environment is changing, dynamic regret is a more suitable performance metric.

D.2 Algorithm 2

We present our second algorithm for online DR-submodular maximization with adversarial or stochastic constraints in Algorithm 2. Note that in the algorithm, we denote $\bar{g}_q(x) = \frac{\sum_{k=1}^K g_{t_{q,k}}(x)}{K}$ $\forall q \in [Q], k \in [K]$.



(a) (b)

Figure D.1: (a) Achievable bounds of Algorithm 1 in the adversarial setting for different choices of window length W, and (b) Excess overall utility of the shifting benchmark sequence with respect to the static benchmark for different numbers of allowed shifts.

D.3 *Missing proofs*

D.3.1 Proof of Lemma 6.4.1

Denote $r_t^{(k)} = V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)} \nabla g_{t-1}(v_{t-1}^{(k)})$. We have:

$$\begin{aligned}
v_t^{(k)} &= \arg\max_{x\in\mathcal{X}} \big(\langle r_t^{(k)}, x\rangle - \alpha\|x - v_{t-1}^{(k)}\|^2\big) \\
&= \arg\min_{x\in\mathcal{X}} \big(- \langle r_t^{(k)}, x\rangle + \alpha\|x - v_{t-1}^{(k)}\|^2\big) \\
&= \arg\min_{x\in\mathcal{X}} \big(- \langle \frac{r_t^{(k)}}{\alpha}, x\rangle + \|x - v_{t-1}^{(k)}\|^2\big) \\
&\overset{(a)}{=} \arg\min_{x\in\mathcal{X}} \big(- \langle \frac{r_t^{(k)}}{\alpha}, x - v_{t-1}^{(k)}\rangle + \|x - v_{t-1}^{(k)}\|^2 + \left\|\frac{r_t^{(k)}}{2\alpha}\right\|^2 \big) \\
&= \arg\min_{x\in\mathcal{X}} \left\| x - v_{t-1}^{(k)} - \frac{r_t^{(k)}}{2\alpha}\right\|^2 \\
&= \mathbb{P}_{\mathcal{X}}\big(v_{t-1}^{(k)} + \frac{r_t^{(k)}}{2\alpha}\big),
\end{aligned}$$

Algorithm 2

```
Input: $\mathcal{X}$ is the constraint set, $T$ is the horizon, $V > 0$, $\alpha > 0$ and $K$.
Output: $\{x_t : 1 \leq t \leq T\}$.
Initialize $\lambda_1^{(k)} = v_0^{(k)} = x_{t_{0,k}}^{(k)} = 0\ \forall k \in [K]$.
for $q = 1$ to $Q$ do
  $x_q^{(1)} = 0$.
  for $k = 1$ to $K$ do
    $v_q^{(k)} = \arg\max_{x \in \mathcal{X}} \left( \langle V \nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}) - \lambda_q^{(k)} \nabla \bar{g}_{q-1}(v_{q-1}^{(k)}), x \rangle - \alpha \|x - v_{q-1}^{(k)}\|^2 \right)$,
    $x_q^{(k+1)} = x_q^{(k)} + \frac{1}{K} v_q^{(k)}$.
  end for
  Let $(t_{q,1}, \ldots, t_{q,K})$ be a random permutation of $\{(q-1)K + 1, \ldots, qK\}$.
  for $t = (q-1)K + 1$ to $qK$ do
    Set $x_t = x_q^{(K+1)}$ and play $x_t$.
  end for
  for $k = 1$ to $K$ do
    $\lambda_{q+1}^{(k)} = [\lambda_q^{(k)} + \bar{g}_{q-1}(v_{q-1}^{(k)}) + \langle \nabla \bar{g}_{q-1}(v_{q-1}^{(k)}), v_q^{(k)} - v_{q-1}^{(k)} \rangle]_+$.
  end for
end for
```

where in (a), we have added the constant term $\langle \frac{r_t^{(k)}}{\alpha}, v_{t-1}^{(k)} \rangle + \left|\left| \frac{r_t^{(k)}}{2\alpha} \right|\right|^2$ to complete the square norm which does not affect the minimizer.

D.3.2 Proof of Lemma 6.4.2

Using the update rule of the algorithm for $\lambda_{t+1}^{(k)}$, we can write:

$$\begin{aligned}\lambda_{t+1}^{(k)} - \lambda_t^{(k)} &\geq g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle \\ &\geq g_{t-1}(v_{t-1}^{(k)}) - \beta \|v_t^{(k)} - v_{t-1}^{(k)}\|\end{aligned}$$

$$= g_{t-1}(v_{t-1}^{(k)}) - \frac{\beta^2}{4V} - V\|v_t^{(k)} - v_{t-1}^{(k)}\|^2 + \underbrace{(\frac{\beta}{2\sqrt{V}} - \sqrt{V}\|v_t^{(k)} - v_{t-1}^{(k)}\|)^2}_{\geq 0}$$

$$\geq g_{t-1}(v_{t-1}^{(k)}) - \frac{\beta^2}{4V} - V\left\|v_t^{(k)} - v_{t-1}^{(k)}\right\|^2,$$

where we have used β-Lipschitzness of the constraint function g_{t-1} and the Cauchy-Schwarz inequality to obtain the second inequality.

Taking the sum over $t \in [T]$, we obtain:

$$\lambda_{T+1}^{(k)} - \underbrace{\lambda_1^{(k)}}_{=0} \geq \sum_{t=1}^{T} g_{t-1}(v_{t-1}^{(k)}) - \frac{\beta^2 T}{4V} - V\sum_{t=1}^{T}\left\|v_t^{(k)} - v_{t-1}^{(k)}\right\|^2,$$

$$\sum_{t=1}^{T} g_{t-1}(v_{t-1}^{(k)}) \leq \lambda_{T+1}^{(k)} + \frac{\beta^2 T}{4V} + V\sum_{t=1}^{T}\left\|v_t^{(k)} - v_{t-1}^{(k)}\right\|^2.$$

Therefore, we have:

$$\begin{aligned} C_T &= \sum_{t=1}^{T} g_{t-1}(x_{t-1}) \\ &= \sum_{t=1}^{T} g_{t-1}(\frac{1}{K}\sum_{k=1}^{K} v_{t-1}^{(k)}) \\ &\leq \frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T} g_{t-1}(v_{t-1}^{(k)}) \\ &\leq \frac{1}{K}\sum_{k=1}^{K}\lambda_{T+1}^{(k)} + \frac{\beta^2 T}{4V} + \frac{V}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\left\|v_t^{(k)} - v_{t-1}^{(k)}\right\|^2, \end{aligned}$$

where the first inequality is due to Jensen's inequality.

D.3.3 Proof of Lemma 6.4.3

Using the update rule of the algorithm for $\lambda_{t+1}^{(k)}$, we have:

$$\begin{aligned} (\lambda_{t+1}^{(k)})^2 &= [\lambda_t^{(k)} + g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle]_+^2 \\ &\leq \big(\lambda_t^{(k)} + g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle\big)^2 \\ &= (\lambda_t^{(k)})^2 + (g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle)^2 + 2\lambda_t^{(k)}(g_{t-1}(v_{t-1}^{(k)}) \end{aligned}$$

$$+ \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle)$$
$$\leq (\lambda_t^{(k)})^2 + (G + \beta R)^2 + 2\lambda_t^{(k)}(g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle).$$

So, we can conclude:

$$\Delta_t^{(k)} = \frac{(\lambda_{t+1}^{(k)})^2}{2} - \frac{(\lambda_t^{(k)})^2}{2} \leq \frac{(G + \beta R)^2}{2} + \lambda_t^{(k)}(g_{t-1}(v_{t-1}^{(k)}) + \langle \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle).$$

D.3.4 Lemma D.3.1

Lemma D.3.1 *The following holds for all $t \in [T]$ and $k \in [K]$:*

$$\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k+1)})\big) \leq (1 - \frac{1}{K})\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)})\big) - \frac{1}{KV}\Delta_t^{(k)} + \frac{V\beta^2}{4\alpha K} + \frac{(G + \beta R)^2}{2KV} + \frac{1}{KV}\lambda_t^{(k)} g_{t-1}(x) + \frac{\alpha}{KV}\left|\left|x - v_{t-1}^{(k)}\right|\right|^2 - \frac{\alpha}{KV}\left|\left|x - v_t^{(k)}\right|\right|^2 + \frac{LR^2}{2K^2}. \tag{D.1}$$

Proof Combining the result of Lemma 6.4.3 with the update rule for $v_t^{(k)}$, we have:

$$\begin{aligned}
&\Delta_t^{(k)} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle + \alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \\
&\leq \frac{(G + \beta R)^2}{2} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)} \nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle + \lambda_t^{(k)} g_{t-1}(v_{t-1}^{(k)}) + \alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \\
&\leq \frac{(G + \beta R)^2}{2} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)} \nabla g_{t-1}(v_{t-1}^{(k)}), x - v_{t-1}^{(k)} \rangle + \lambda_t^{(k)} g_{t-1}(v_{t-1}^{(k)}) + \alpha \left|\left|x - v_{t-1}^{(k)}\right|\right|^2 \\
&- \alpha \left|\left|x - v_t^{(k)}\right|\right|^2 \\
&\leq \frac{(G + \beta R)^2}{2} - \langle V\nabla f_{t-1}(x_{t-1}^{(k)}), x - v_{t-1}^{(k)} \rangle + \lambda_t^{(k)} g_{t-1}(x) + \alpha \left|\left|x - v_{t-1}^{(k)}\right|\right|^2 - \alpha \left|\left|x - v_t^{(k)}\right|\right|^2,
\end{aligned} \tag{D.2}$$

where the second inequality is due to the optimality condition of the optimization problem for updating $v_t^{(k)}$ and the last inequality follows from the convexity of the constraint function g_{t-1}.

We have:

$$- \langle V\nabla f_{t-1}(x_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)} \rangle + \alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2$$

$$= \left|\left|\sqrt{\alpha}(v_t^{(k)} - v_{t-1}^{(k)}) - \frac{V}{2\sqrt{\alpha}} \nabla f_{t-1}(x_{t-1}^{(k)})\right|\right|^2 - \frac{V^2}{4\alpha} \left|\left|\nabla f_{t-1}(x_{t-1}^{(k)})\right|\right|^2$$

$$\geq -\frac{V^2\beta^2}{4\alpha}.$$

Using L-smoothness of the utility functions, we can write:

$$f_{t-1}(x_{t-1}^{(k+1)}) \geq f_{t-1}(x_{t-1}^{(k)}) + \frac{1}{K}\langle v_{t-1}^{(k)}, \nabla f_{t-1}(x_{t-1}^{(k)})\rangle - \frac{L}{2K^2}\left|\left|v_{t-1}^{(k)}\right|\right|^2,$$

$$V\langle v_{t-1}^{(k)}, \nabla f_{t-1}(x_{t-1}^{(k)})\rangle \leq KV\big(f_{t-1}(x_{t-1}^{(k+1)}) - f_{t-1}(x_{t-1}^{(k)})\big) + \frac{LR^2V}{2K}.$$

Using the DR-submodularity and monotonicity of the reward functions, we have:

$$\begin{aligned} f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)}) &\leq f_{t-1}(x \vee x_{t-1}^{(k)}) - f_{t-1}(x_{t-1}^{(k)}) \\ &\leq \langle \nabla f_{t-1}(x_{t-1}^{(k)}), (x \vee x_{t-1}^{(k)}) - x_{t-1}^{(k)}\rangle \\ &= \langle \nabla f_{t-1}(x_{t-1}^{(k)}), (x - x_{t-1}^{(k)}) \vee 0\rangle \\ &\leq \langle \nabla f_{t-1}(x_{t-1}^{(k)}), x\rangle. \end{aligned}$$

Putting the above inequalities together, we have:

$$\begin{aligned} \Delta_t^{(k)} - \frac{V^2\beta^2}{4\alpha} \leq \frac{(G+\beta R)^2}{2} &- V\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)})\big) + KV\big(f_{t-1}(x_{t-1}^{(k+1)}) - f_{t-1}(x_{t-1}^{(k)})\big) \\ &+ \lambda_t^{(k)} g_{t-1}(x) + \alpha\left|\left|x - v_{t-1}^{(k)}\right|\right|^2 - \alpha\left|\left|x - v_t^{(k)}\right|\right|^2 + \frac{LR^2V}{2K}. \end{aligned}$$

Rearranging the terms, we can write:

$$\begin{aligned} KV\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k+1)})\big) \leq (K-1)V\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)})\big) - \Delta_t^{(k)} + \frac{V^2\beta^2}{4\alpha} + \frac{(G+\beta R)^2}{2} \\ + \lambda_t^{(k)} g_{t-1}(x) + \alpha\left|\left|x - v_{t-1}^{(k)}\right|\right|^2 - \alpha\left|\left|x - v_t^{(k)}\right|\right|^2 + \frac{LR^2V}{2K}. \end{aligned}$$

Dividing both sides by KV, we obtain the desired result. ■

D.3.5 Lemma D.3.2

Lemma D.3.2 *The static regret of Algorithm 1 against $x \in \mathcal{X}$ is bounded as follows:*

$$\sum_{t=1}^{T}\big((1-\frac{1}{e})f_{t-1}(x) - f_{t-1}(x_{t-1})\big) \leq \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x) + \frac{\alpha R^2}{V} + \frac{LR^2T}{2K}. \tag{D.3}$$

Proof Using inequality D.1 and taking the sum over $t \in [T]$, we obtain:

$$\begin{aligned}
&\sum_{t=1}^{T}\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k+1)})\big) \\
&\le (1-\frac{1}{K})\sum_{t=1}^{T}\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)})\big) - \frac{1}{2KV}(\lambda_{T+1}^{(k)})^2 + \frac{1}{2KV}(\underbrace{\lambda_1^{(k)}}_{=0})^2 + \frac{V\beta^2 T}{4\alpha K} + \frac{(G+\beta R)^2 T}{2KV} \\
&+ \frac{1}{KV}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x) + \frac{\alpha}{KV}\left\|x - v_0^{(k)}\right\|^2 - \frac{\alpha}{KV}\left\|x - v_T^{(k)}\right\|^2 + \frac{LR^2T}{2K^2} \\
&\le (1-\frac{1}{K})\sum_{t=1}^{T}\big(f_{t-1}(x) - f_{t-1}(x_{t-1}^{(k)})\big) + \frac{V\beta^2 T}{4\alpha K} + \frac{(G+\beta R)^2 T}{2KV} + \frac{1}{KV}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x) + \frac{\alpha R^2}{KV} + \frac{LR^2 T}{2K^2}.
\end{aligned}$$

Applying the above inequality recursively for all $k \in [K]$, we have:

$$\begin{aligned}
\sum_{t=1}^{T}\big(f_{t-1}(x) - f_{t-1}(\underbrace{x_{t-1}^{(K+1)}}_{x_{t-1}})\big) \le \underbrace{(1-\frac{1}{K})^K}_{\le \frac{1}{e}}\sum_{t=1}^{T}\big(f_{t-1}(x) - f_{t-1}(\underbrace{x_{t-1}^{(1)}}_{=0})\big) + \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} \\
+ \frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x) + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K}.
\end{aligned}$$

Rearranging the terms, we obtain the desired result. ■

D.3.6 Lemma D.3.3

Lemma D.3.3 *For all $t \in [T], k \in [K]$, we have $\lambda_t^{(k)} \le \theta V$ where*

$$\theta = \max\{G+\beta R, \frac{\frac{(G+\beta R)^2}{2} + (\beta R + \frac{V\beta^2}{4\alpha})V}{VB_T/T} + \frac{\alpha R^2}{V(V+1)B_T/T} + \frac{(G+\beta R)(V+2)}{2V}\}.$$

In particular, if we choose $\alpha \le \mathcal{O}(V^2)$, we have $\lambda_t^{(k)} \le \mathcal{O}(V)$.

Proof Plugging in $x = 0$ in inequality D.2, we obtain:

$$\Delta_t^{(k)} - \frac{V^2\beta^2}{4\alpha} \le \frac{(G+\beta R)^2}{2} + \underbrace{\langle V\nabla f_{t-1}(x_{t-1}^{(k)}), v_{t-1}^{(k)}\rangle}_{\le \beta RV} - \frac{B_T}{T}\lambda_t^{(k)} + \alpha\left\|v_{t-1}^{(k)}\right\|^2 - \alpha\left\|v_t^{(k)}\right\|^2,$$

$$\Delta_t^{(k)} \le \frac{(G+\beta R)^2}{2} + (\beta R + \frac{V\beta^2}{4\alpha})V - \frac{B_T}{T}\lambda_t^{(k)} + \alpha\left\|v_{t-1}^{(k)}\right\|^2 - \alpha\left\|v_t^{(k)}\right\|^2.$$

Therefore, setting $\eta = -\frac{B_T}{T}$, $B = \frac{(G+\beta R)^2}{2}$ and $R = \beta R + \frac{V\beta^2}{4\alpha}$ in Thoerem 2 of [159], we obtain the desired result. ■

This $\mathcal{O}(V)$ bound on the dual variables is crucial in obtaining improved total constraint violation bounds compared to [181]. Note that [159] requires the extra assumption that there exists an action $z \in \mathcal{X}$ such that $g_t(z) < 0\ \forall t \in [T]$ (Slater condition) to obtain Theorem 2 in their paper. However, in our framework, since $g_t(\cdot) = h_t(\cdot) - \frac{B_T}{T}$ and $h_t(0) = 0$ for all $t \in [T]$, the Slater condition holds naturally with $z = 0$.

D.3.7 Proof of Theorem 6.4.1

First, using the result of Lemma 6.4.2, we have:

$$C_T \leq \frac{1}{K}\sum_{k=1}^{K} \lambda_{T+1}^{(k)} + \frac{\beta^2 T}{4V} + \frac{V}{K}\sum_{k=1}^{K}\sum_{t=1}^{T} \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2.$$

Therefore, in order to bound the total constraint violation, we need to bound $\lambda_{T+1}^{(k)}$ and $\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2$ for all $k \in [K]$ and $t \in [T]$. Lemma D.3.3 provides the bound $\lambda_{T+1}^{(k)} \leq \theta V$ for the dual variables. Thus, it suffices to obtain upper bounds for the terms $\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2$ which is done in the following.

Using the update rule of the algorithm for $v_t^{(k)}$, we have:

$$\begin{aligned}\langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)}\rangle - \alpha\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 &\geq \langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)}), v_{t-1}^{(k)}\rangle \\ &\quad + \alpha\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2.\end{aligned}$$

Equivalently, we can write:

$$\begin{aligned}2\alpha\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 &\leq \langle V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)}), v_t^{(k)} - v_{t-1}^{(k)}\rangle \\ &\leq \left|\left|V\nabla f_{t-1}(x_{t-1}^{(k)}) - \lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)})\right|\right| \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|,\end{aligned}$$

where we have used the Cauchy-Schwarz inequality to obtain the last inequality.
Dividing both sides by $2\alpha\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|$ and using the triangle inequality, we obtain:

$$\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right| \leq \frac{1}{2\alpha}\big(\left|\left|V\nabla f_{t-1}(x_{t-1}^{(k)})\right|\right| + \left|\left|\lambda_t^{(k)}\nabla g_{t-1}(v_{t-1}^{(k)})\right|\right|\big)$$

$$\begin{aligned} &\leq \frac{\beta}{2\alpha}(V + \lambda_t^{(k)}) \\ &\leq \frac{\beta(1+\theta)}{2\alpha}V. \end{aligned}$$

Therefore, the following holds:

$$\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \leq \frac{\beta^2(1+\theta)^2}{4\alpha^2}V^2.$$

Plugging the above inequality in the result of Lemma 6.4.2, we obtain the total constraint violation bound as follows:

$$C_T \leq \frac{1}{K}\sum_{k=1}^{K}\theta V + \frac{\beta^2 T}{4V} + \frac{V}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \leq \theta V + \frac{\beta^2 T}{4V} + \frac{\beta^2(1+\theta)^2 V^3 T}{4\alpha^2}.$$

D.3.8 Proof of Theorem 6.4.2

First, note that in the adversarial setting with window size $W = 1$, since for all $t \in [T]$ and $k \in [K]$, $g_t(x^*) \leq 0$ holds and $\lambda_t^{(k)}$ is non-negative, we have $\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x^*) \leq 0$. Thus, in this setting, the result follows immediately from inequality D.3.

For the case where $W > 1$, plugging in $t \leftarrow t + \tau$ and $x = x_W^*$ in inequality D.1, we have:

$$\begin{aligned} &\big(f_{t+\tau-1}(x_W^*) - f_{t+\tau-1}(x_{t+\tau-1}^{(k+1)})\big) \leq (1 - \frac{1}{K})\big(f_{t+\tau-1}(x_W^*) - f_{t+\tau-1}(x_{t+\tau-1}^{(k)})\big) - \frac{1}{KV}\Delta_{t+\tau}^{(k)} + \frac{V\beta^2}{4\alpha K} \\ &+ \frac{(G+\beta R)^2}{2KV} + \frac{LR^2}{2K^2} + \frac{1}{KV}\lambda_{t+\tau}^{(k)} g_{t+\tau-1}(x_W^*) + \frac{\alpha}{KV}\left|\left|x_W^* - v_{t+\tau-1}^{(k)}\right|\right|^2 - \frac{\alpha}{KV}\left|\left|x_W^* - v_{t+\tau}^{(k)}\right|\right|^2. \end{aligned}$$

Taking the sum over all $t \in [T - W + 1], \tau \in \{0, \ldots, W - 1\}$ and applying the inequality recursively for all $k \in [K]$, we obtain:

$$\begin{aligned} &\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau-1}(x_W^*) - f_{t+\tau-1}(\underbrace{x_{t+\tau-1}^{(K+1)}}_{=x_{t+\tau-1}})\big) \leq \frac{1}{e}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big(f_{t+\tau-1}(x_W^*) - f_{t+\tau-1}(\underbrace{x_{t+\tau-1}^{(1)}}_{=0})\big) \\ &- \frac{1}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\sum_{t=1}^{T-W+1}\Delta_{t+\tau}^{(k)} + \frac{V\beta^2 W(T-W+1)}{4\alpha} + \frac{(G+\beta R)^2 W(T-W+1)}{2V} + \frac{LR^2 W(T-W+1)}{2K} \\ &+ \frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\lambda_{t+\tau}^{(k)} g_{t+\tau-1}(x_W^*) + \frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\sum_{t=1}^{T-W+1}\big(\left|\left|x_W^* - v_{t+\tau-1}^{(k)}\right|\right|^2 - \left|\left|x_W^* - v_{t+\tau}^{(k)}\right|\right|^2\big). \end{aligned}$$

Equivalently, we can write:

$$\begin{aligned}
&\underbrace{\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\big((1-\frac{1}{e})f_{t+\tau-1}(x_W^*) - f_{t+\tau-1}(x_{t+\tau-1})\big)}_{\text{(a)}} \le \underbrace{-\frac{1}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\sum_{t=1}^{T-W+1}\Delta_{t+\tau}^{(k)}}_{\text{(b)}} \\
&+ \frac{V\beta^2 W(T-W+1)}{4\alpha} + \frac{(G+\beta R)^2 W(T-W+1)}{2V} + \underbrace{\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T-W+1}\sum_{\tau=0}^{W-1}\lambda_{t+\tau}^{(k)} g_{t+\tau-1}(x_W^*)}_{\text{(c)}} \\
&+ \underbrace{\frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\sum_{t=1}^{T-W+1}\big(\left\|x_W^* - v_{t+\tau-1}^{(k)}\right\|^2 - \left\|x_W^* - v_{t+\tau}^{(k)}\right\|^2\big)}_{\text{(d)}} + \frac{LR^2 W(T-W+1)}{2K}. \qquad \text{(D.4)}
\end{aligned}$$

The main challenge in obtaining regret bounds for the $W > 1$ case is to bound terms (a), (b), (c), and (d) in the above inequality. We exploit ideas from the analysis in [181, 135] to obtain these bounds.

We bound term (a) in the following:

$$\begin{aligned}
\text{(a)} &= WR_T - \sum_{i=1}^{W-1}(W-i)\big([(1-\frac{1}{e})f_{i-1}(x_W^*) - f_{i-1}(x_i)] \\
&\quad + [(1-\frac{1}{e})f_{T-i}(x_W^*) - f_{T-i}(x_{T-i})]\big) \\
&\ge WR_T - 2F\sum_{i=1}^{W-1}(W-i) \\
&= WR_T - FW(W-1). \qquad \text{(D.5)}
\end{aligned}$$

For the term (b), we have:

$$\begin{aligned}
\text{(b)} &= -\frac{1}{2KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\sum_{t=1}^{T-W+1}\big((\lambda_{t+\tau+1}^{(k)})^2 - (\lambda_{t+\tau}^{(k)})^2\big) \\
&= -\frac{1}{2KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\big((\lambda_{T-W+\tau+2}^{(k)})^2 - (\lambda_{\tau+1}^{(k)})^2\big) \\
&\le \frac{1}{2KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}(\lambda_{\tau+1}^{(k)})^2 \\
&\le \frac{1}{2KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\min\{\theta^2V^2, \tau^2(G+\beta R)^2\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2KV}\min\{\theta^2V^2KW, (G+\beta R)^2K\sum_{\tau=0}^{W-1}\tau^2\} \\
&= \frac{1}{2KV}\min\{\theta^2V^2KW, (G+\beta R)^2K\frac{W(W-1)(2W-1)}{6}\} \\
&= \frac{1}{2V}\min\{\theta^2V^2W, (G+\beta R)^2\frac{W(W-1)(2W-1)}{6}\}, \qquad \text{(D.6)}
\end{aligned}
$$

where we have used the dual variable bounds in Lemma D.3.3 and the fact that $\lambda_t^{(k)}$ changes by at most $G+\beta R$ over one slot to obtain the second inequality.

In order to bound (c), we use Lemma 8 of [135] to obtain:

$$
\begin{aligned}
\text{(c)} &\leq \frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T-W+1}\big(\underbrace{\lambda_t^{(k)}\sum_{\tau=0}^{W-1}g_{t+\tau-1}(x_W^*)}_{\leq 0} + \frac{(G+\beta R)^2}{2}W(W-1)\big)\big) \\
&\leq \frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T-W+1}\frac{(G+\beta R)^2}{2}W(W-1) \\
&= \frac{(G+\beta R)^2W(W-1)(T-W+1)}{2V}. \qquad \text{(D.7)}
\end{aligned}
$$

Finally, for the term (d), we can write:

$$
\begin{aligned}
\text{(d)} &= \frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\big(\left|\left|x_W^* - v_\tau^{(k)}\right|\right|^2 - \left|\left|x_W^* - v_{T-W+\tau+1}^{(k)}\right|\right|^2\big) \\
&\leq \frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}\left|\left|x_W^* - v_\tau^{(k)}\right|\right|^2 \\
&\leq \frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{\tau=0}^{W-1}R^2 \\
&= \frac{\alpha R^2W}{V}. \qquad \text{(D.8)}
\end{aligned}
$$

Combining inequalities D.4, D.5, D.6, D.7 and D.8, dividing both sides by W and rearranging the terms, we obtain the regret bound as stated.

D.3.9 Proof of Theorem 6.4.3

Set $x_0^* = 0$. Plugging in $x = x_{t-1}^*$ in inequality D.1, we have:

$$
\big(f_{t-1}(x_{t-1}^*) - f_{t-1}(x_{t-1}^{(k+1)})\big) \leq (1-\frac{1}{K})\big(f_{t-1}(x_{t-1}^*) - f_{t-1}(x_{t-1}^{(k)})\big) - \frac{1}{KV}\Delta_t^{(k)} + \frac{V\beta^2}{4\alpha K}
$$

$$+\frac{(G+\beta R)^2}{2KV}+\frac{1}{KV}\lambda_t^{(k)}g_{t-1}(x_{t-1}^*)+\frac{\alpha}{KV}\left|\left|x_{t-1}^*-v_{t-1}^{(k)}\right|\right|^2-\frac{\alpha}{KV}\left|\left|x_{t-1}^*-v_t^{(k)}\right|\right|^2+\frac{LR^2}{2K^2}.$$

Taking the sum over all $t \in [T]$ and applying the inequality recursively for all $k \in [K]$, we obtain:

$$\sum_{t=1}^{T}\big(f_{t-1}(x_{t-1}^*)-f_{t-1}(\underbrace{x_{t-1}^{(K+1)}}_{=x_{t-1}})\big)\leq\frac{1}{e}\sum_{t=1}^{T}\big(f_{t-1}(x_{t-1}^*)-f_{t-1}(\underbrace{x_{t-1}^{(1)}}_{=0})\big)+\frac{V\beta^2T}{4\alpha}+\frac{(G+\beta R)^2T}{2V}$$
$$+\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)}g_{t-1}(x_{t-1}^*)+\frac{\alpha}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T-1}\big(\left|\left|x_t^*-v_t^{(k)}\right|\right|^2-\left|\left|x_{t-1}^*-v_t^{(k)}\right|\right|^2\big)+\frac{LR^2T}{2K}$$
$$+\frac{\alpha}{KV}\sum_{k=1}^{K}\underbrace{\left|\left|x_0^*-v_0^{(k)}\right|\right|^2}_{=0}-\frac{\alpha}{KV}\sum_{k=1}^{K}\left|\left|x_{T-1}^*-v_t^{(k)}\right|\right|^2.$$

Considering that $\left|\left|x_t^*-v_t^{(k)}\right|\right|^2-\left|\left|x_{t-1}^*-v_t^{(k)}\right|\right|^2=||x_t^*||^2-\left|\left|x_{t-1}^*\right|\right|^2+2\langle v_t^{(k)},x_{t-1}^*-x_t^*\rangle\leq$ $||x_t^*||^2-\left|\left|x_{t-1}^*\right|\right|^2+2R\left|\left|x_{t-1}^*-x_t^*\right|\right|$ holds, we can write:

$$\sum_{t=1}^{T}\big(f_{t-1}(x_{t-1}^*)-f_{t-1}(x_{t-1})\big)\leq\frac{1}{e}\sum_{t=1}^{T}f_{t-1}(x_{t-1}^*)+\frac{V\beta^2T}{4\alpha}+\frac{(G+\beta R)^2T}{2V}+\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)}g_{t-1}(x_{t-1}^*)$$
$$+\frac{\alpha}{KV}\sum_{k=1}^{K}\big(\underbrace{\left|\left|x_{T-1}^*\right|\right|^2}_{\leq R^2}-\underbrace{||x_0^*||^2}_{=0}+2R\sum_{t=1}^{T-1}\left|\left|x_{t-1}^*-x_t^*\right|\right|\big)+\frac{LR^2T}{2K}.$$

Denoting the drift of the benchmark sequence $P_T^*=\sum_{t=1}^{T-1}\left|\left|x_{t-1}^*-x_t^*\right|\right|$, we get the dynamic regret bound as desired.

D.3.10 Proof of Theorem 6.4.4

Taking expectation of both sides of inequality D.1, we have:

$$\mathbb{E}\big[f_{t-1}(x^*)-f_{t-1}(x_{t-1}^{(k+1)})\big]\leq(1-\frac{1}{K})\mathbb{E}\big[f_{t-1}(x^*)-f_{t-1}(x_{t-1}^{(k)})\big]-\frac{1}{KV}\mathbb{E}[\Delta_t^{(k)}]+\frac{V\beta^2}{4\alpha K}+\frac{(G+\beta R)^2}{2KV}$$
$$+\frac{1}{KV}\mathbb{E}[\lambda_t^{(k)}g_{t-1}(x^*)]+\frac{\alpha}{KV}\mathbb{E}\left|\left|x^*-v_{t-1}^{(k)}\right|\right|^2-\frac{\alpha}{KV}\mathbb{E}\left|\left|x^*-v_t^{(k)}\right|\right|^2+\frac{LR^2}{2K^2}.$$

Let $\mathcal{F}_t=\{g_\tau\}_{\tau=0}^{t-1}$. Considering that $\lambda_t^{(k)}$ is $\mathcal{F}_{t-1}$-measurable and $g_{t-1}(x^*)$ is independent of $\mathcal{F}_{t-1}$, we can write:

$$\mathbb{E}[\lambda_t^{(k)}g_{t-1}(x^*)]=\mathbb{E}\big[\mathbb{E}[\lambda_t^{(k)}g_{t-1}(x^*)|\mathcal{F}_{t-1}]\big]=\mathbb{E}\big[\lambda_t^{(k)}\underbrace{\mathbb{E}[g_{t-1}(x^*)]}_{\leq 0}\big]\leq 0.$$

Combining the above inequalities, taking the sum over $t \in [T]$ and applying the inequality recursively for all $k \in [K]$, we obtain:

$$\sum_{t=1}^{T} \mathbb{E}\big[f_{t-1}(x^*) - f_{t-1}(\underbrace{x_{t-1}^{(K+1)}}_{=x_{t-1}})\big] \leq \underbrace{(1-\frac{1}{K})^K}_{\leq \frac{1}{e}} \mathbb{E}\big[\sum_{t=1}^{T}\big(f_{t-1}(x^*) - f_{t-1}(\underbrace{x_{t-1}^{(1)}}_{=0})\big] \underbrace{-\frac{1}{2KV}\sum_{k=1}^{K}\mathbb{E}[\lambda_{T+1}^{(k)}]^2}_{\leq 0}$$
$$+ \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K}.$$

Therefore, the expected regret bound is derived as stated.

D.3.11 Proof of Theorem 6.4.5

Considering the regret bound in inequality D.3, in order to obtain a high probability regret bound, we have to bound $\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)} g_{t-1}(x^*)$. Denote $Y_t = \frac{1}{KV}\sum_{k=1}^{K}\sum_{s=1}^{t}\lambda_s^{(k)} g_{s-1}(x^*)$ and let $\mathcal{F}_t = \{g_\tau\}_{\tau=0}^{t-1}$. Considering that $g_{t-1}(x^*)$ is independent of $\mathcal{F}_{t-1}$,We have:

$$\begin{aligned}
\mathbb{E}[Y_t|\mathcal{F}_{t-1}] &= \mathbb{E}[Y_{t-1} + \frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)} g_{t-1}(x^*)|\mathcal{F}_{t-1}] \\
&= Y_{t-1} + \mathbb{E}[\frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)} g_{t-1}(x^*)|\mathcal{F}_{t-1}] \\
&= Y_{t-1} + \frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)} \mathbb{E}[g_{t-1}(x^*)|\mathcal{F}_{t-1}] \\
&= Y_{t-1} + \frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)} \underbrace{\mathbb{E}[g_{t-1}(x^*)]}_{\leq 0} \\
&\leq Y_{t-1}.
\end{aligned}$$

Therefore, $\{Y_t, \mathcal{F}_t\}_{t\geq 0}$ is a supermartingale. Also, note that for all $t \in [T]$, we have:

$$|Y_t - Y_{t-1}| = |\frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)} g_{t-1}(x^*)| \leq \frac{1}{KV}\sum_{k=1}^{K}\lambda_t^{(k)}|g_{t-1}(x^*)| \leq \theta G.$$

Thus, using the Azuma-Hoeffding inequality, we can conclude that with probability $1-\delta$, the following holds:

$$\frac{1}{KV}\sum_{k=1}^{K}\sum_{t=1}^{T}\lambda_t^{(k)}g_{t-1}(x^*) \leq \theta G\sqrt{2T\log(\frac{1}{\delta})}.$$

Combining the above inequality with the regret bound in inequality D.3, with probability $1-\delta$, we have:

$$R_T^{(S,S)} \leq \theta G\sqrt{2T\log(\frac{1}{\delta})} + \frac{V\beta^2 T}{4\alpha} + \frac{(G+\beta R)^2 T}{2V} + \frac{\alpha R^2}{V} + \frac{LR^2 T}{2K}.$$

D.3.12 Proof of Theorem 6.5.1

Using an analysis similar to Lemma 6.4.1, we have:

$$\sum_{q=1}^{Q}\bar{g}_{q-1}(v_{q-1}^{(k)}) \leq \lambda_{Q+1}^{(k)} + \frac{\beta^2 Q}{4V} + V\sum_{q=1}^{Q}\left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2.$$

The constraint violation bound follows immediately from the result of Lemma 6.4.2 and it is provided below:

$$\begin{aligned}
\mathbb{E}[C_T] &= \mathbb{E}\big[K\sum_{q=1}^{Q}\bar{g}_{q-1}(x_{q-1}^{(K+1)})\big] \\
&= \mathbb{E}\big[K\sum_{q=1}^{Q}\bar{g}_{q-1}(\frac{1}{K}\sum_{k=1}^{K}v_{q-1}^{(k)})\big] \\
&\leq \sum_{k=1}^{K}\sum_{q=1}^{Q}\mathbb{E}[\bar{g}_{q-1}(v_{q-1}^{(k)})] \\
&\leq \sum_{k=1}^{K}\mathbb{E}[\lambda_{Q+1}^{(k)}] + \frac{\beta^2 QK}{4V} + V\sum_{k=1}^{K}\sum_{q=1}^{Q}\mathbb{E}\left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2, \qquad \text{(D.9)}
\end{aligned}$$

where the first inequality is due to Jensen's inequality.

Plugging in $t \leftarrow q$ and using the dual update of Algorithm 2 instead of Algorithm 1 in Lemma 6.4.3, we have:

$$\Delta_q^{(k)} := \frac{(\lambda_{q+1}^{(k)})^2}{2} - \frac{(\lambda_q^{(k)})^2}{2} \leq \frac{(G+\beta R)^2}{2} + \lambda_q^{(k)}(\bar{g}_{q-1}(v_{q-1}^{(k)}) + \langle\nabla\bar{g}_{q-1}(v_{q-1}^{(k)}), v_q^{(k)} - v_{q-1}^{(k)}\rangle).$$

Combining the above inequality with the update rule for $v_q^{(k)}$, we have:

$$
\begin{aligned}
&\Delta_q^{(k)} \underbrace{-\langle V\nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}), v_q^{(k)} - v_{q-1}^{(k)}\rangle + \alpha \left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2}_{\text{(a)}} \\
&\leq \frac{(G+\beta R)^2}{2} - \langle V\nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}) - \lambda_q^{(k)}\nabla \bar{g}_{q-1}(v_{q-1}^{(k)}), v_q^{(k)} - v_{q-1}^{(k)}\rangle + \lambda_t^{(k)}\bar{g}_{q-1}(v_{q-1}^{(k)}) \\
&+ \alpha \left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2 \\
&\leq \frac{(G+\beta R)^2}{2} - \langle V\nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}) - \lambda_q^{(k)}\nabla \bar{g}_{q-1}(v_{q-1}^{(k)}), x - v_{q-1}^{(k)}\rangle + \lambda_q^{(k)}\bar{g}_{q-1}(v_{q-1}^{(k)}) \\
&+ \alpha \left|\left|x - v_{q-1}^{(k)}\right|\right|^2 - \alpha \left|\left|x - v_q^{(k)}\right|\right|^2 \\
&\leq \frac{(G+\beta R)^2}{2} - \langle V\nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}), x - v_{q-1}^{(k)}\rangle + \lambda_q^{(k)}\bar{g}_{q-1}(x) + \alpha \left|\left|x - v_{q-1}^{(k)}\right|\right|^2 - \alpha \left|\left|x - v_q^{(k)}\right|\right|^2,
\end{aligned}
$$

where we have used convexity of $\bar{g}_{q-1}$ to derive the last inequality.

For the term (a), we have:

$$
\begin{aligned}
\text{(a)} &= \left|\left|\sqrt{\alpha}(v_q^{(k)} - v_{q-1}^{(k)}) - \frac{V}{2\sqrt{\alpha}}\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})\right|\right|^2 - \frac{V^2}{4\alpha}\left|\left|\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})\right|\right|^2 \\
&\geq -\frac{V^2\beta^2}{4\alpha}.
\end{aligned}
$$

Using L-smoothness of the utility functions, we can write:

$$
\begin{aligned}
f_{t_{q-1,k}}(x_{q-1}^{(k+1)}) &\geq f_{t_{q-1,k}}(x_{q-1}^{(k)}) + \frac{1}{K}\langle v_{q-1}^{(k)}, \nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})\rangle - \frac{L}{2K^2}\left|\left|v_{q-1}^{(k)}\right|\right|^2 \\
V\langle v_{q-1}^{(k)}, \nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})\rangle &\leq KV\big(f_{t_{q-1,k}}(x_{q-1}^{(k+1)}) - f_{t_{q-1,k}}(x_{q-1}^{(k)})\big) + \frac{LR^2V}{2K}.
\end{aligned}
$$

Using the DR-submodularity and monotonicity of the reward functions, we have:

$$
\begin{aligned}
f_{t_{q-1,k}}(x) - f_{t_{q-1,k}}(x_{q-1}^{(k)}) &\leq f_{t_{q-1,k}}(x \vee x_{q-1}^{(k)}) - f_{t_{q-1,k}}(x_{q-1}^{(k)}) \\
&\leq \langle \nabla f_{t_{q-1,k}}(x_{q-1}^{(k)}), (x \vee x_{q-1}^{(k)}) - x_{q-1}^{(k)}\rangle \\
&= \langle \nabla f_{t_{q-1,k}}(x_{q-1}^{(k)}), (x - x_{q-1}^{(k)}) \vee 0\rangle \\
&\leq \langle \nabla f_{t_{q-1,k}}(x_{q-1}^{(k)}), x\rangle.
\end{aligned}
$$

Putting the above inequalities together, we have:

$$
\Delta_q^{(k)} - \frac{V^2\beta^2}{4\alpha} \leq \frac{(G+\beta R)^2}{2} - V\big(f_{t_{q-1,k}}(x) - f_{t_{q-1,k}}(x_{q-1}^{(k)})\big) + KV\big(f_{t_{q-1,k}}(x_{q-1}^{(k+1)}) - f_{t_{q-1,k}}(x_{q-1}^{(k)})\big)
$$

$$+\lambda_q^{(k)}\bar{g}_{q-1}(x)+\alpha\left|\left|x-v_{q-1}^{(k)}\right|\right|^2-\alpha\left|\left|x-v_q^{(k)}\right|\right|^2+\frac{LR^2V}{2K}.$$

Equivalently, we can write:

$$KV\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(x_{q-1}^{(k+1)})\big)\leq(K-1)V\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(x_{q-1}^{(k)})\big)-\Delta_q^{(k)}+\frac{V^2\beta^2}{4\alpha}$$
$$+\frac{(G+\beta R)^2}{2}+\lambda_q^{(k)}\bar{g}_{q-1}(x)+\alpha\left|\left|x-v_{q-1}^{(k)}\right|\right|^2-\alpha\left|\left|x-v_q^{(k)}\right|\right|^2+\frac{LR^2V}{2K}.$$

Dividing both sides by KV and taking the sum over $q\in[Q]$, we obtain:

$$\sum_{q=1}^{Q}\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(x_{q-1}^{(k+1)})\big)\leq(1-\frac{1}{K})\sum_{q=1}^{Q}\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(x_{q-1}^{(k)})\big)+\frac{V\beta^2Q}{4\alpha K}$$
$$+\frac{(G+\beta R)^2Q}{2KV}+\frac{1}{KV}\sum_{q=1}^{Q}\lambda_q^{(k)}\bar{g}_{q-1}(x)+\frac{\alpha}{KV}\left|\left|x-v_0^{(k)}\right|\right|^2-\frac{\alpha}{KV}\left|\left|x-v_Q^{(k)}\right|\right|^2+\frac{LR^2Q}{2K^2}.$$

Applying the above inequality recursively for all $k\in[K]$, we have:

$$\sum_{q=1}^{Q}\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(x_{q-1}^{(K+1)})\big)\leq\underbrace{(1-\frac{1}{K})^K}_{\leq\frac{1}{e}}\sum_{q=1}^{Q}\big(f_{t_{q-1,k}}(x)-f_{t_{q-1,k}}(\underbrace{x_{q-1}^{(1)}}_{=0})\big)+\frac{V\beta^2Q}{4\alpha}$$
$$+\frac{(G+\beta R)^2Q}{2V}+\frac{1}{KV}\sum_{k=1}^{K}\sum_{q=1}^{Q}\lambda_q^{(k)}\bar{g}_{q-1}(x)+\frac{\alpha R^2}{V}+\frac{LR^2Q}{2K}.$$

Therefore, the regret against the benchmark with window length $W\in[1,T^{\frac{1}{3}}]$ is derived as stated.

D.3.13 Proof of Theorem 6.5.2

For all $q\in[Q],k\in[K]$, using a similar analysis to Lemma D.3.3, we have $\lambda_q^{(k)}\leq\theta V$ where

$$\theta=\max\{G+\beta R,\frac{\frac{(G+\beta R)^2T}{2}+(\beta R+\frac{V\beta^2}{4\alpha})VT}{VB_T}+\frac{\alpha R^2T}{V(V+1)B_T}+\frac{(G+\beta R)(V+2)}{2V}\}.$$

Using the update rule of the algorithm, we have:

$$\langle V\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})-\lambda_q^{(k)}\nabla\bar{g}_{q-1}(v_{q-1}^{(k)}),v_q^{(k)}\rangle-\alpha\left|\left|v_q^{(k)}-v_{q-1}^{(k)}\right|\right|^2\geq$$
$$\langle V\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})-\lambda_q^{(k)}\nabla\bar{g}_{q-1}(v_{q-1}^{(k)}),v_{q-1}^{(k)}\rangle+\alpha\left|\left|v_q^{(k)}-v_{q-1}^{(k)}\right|\right|^2.$$

Equivalently, we can write:

$$\begin{aligned}
2\alpha \left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2 &\leq \langle V\nabla f_{t_{q-1,k}}(x_{t_{q-1,k}}^{(k)}) - \lambda_q^{(k)}\nabla \bar{g}_{q-1}(v_{q-1}^{(k)}), v_q^{(k)} - v_{q-1}^{(k)}\rangle \\
&\leq \left|\left|V\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)}) - \lambda_q^{(k)}\nabla \bar{g}_{q-1}(v_{q-1}^{(k)})\right|\right| \left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|.
\end{aligned}$$

Dividing both sides by $2\alpha \left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|$ and using the triangle inequality, we obtain:

$$\begin{aligned}
\left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right| &\leq \frac{1}{2\alpha}\big(\left|\left|V\nabla f_{t_{q-1,k}}(x_{q-1}^{(k)})\right|\right| + \left|\left|\lambda_q^{(k)}\nabla \bar{g}_{q-1}(v_{q-1}^{(k)})\right|\right| \big) \\
&\leq \frac{\beta}{2\alpha}(V + \lambda_q^{(k)}) \\
&\leq \frac{\beta(1+\theta)}{2\alpha}V.
\end{aligned}$$

Therefore, the following holds:

$$\left|\left|v_q^{(k)} - v_{q-1}^{(k)}\right|\right|^2 \leq \frac{\beta^2(1+\theta)^2}{4\alpha^2}V^2.$$

Plugging the above inequality in inequality D.9, we obtain the constraint violation bound as follows:

$$\mathbb{E}[C_T] \leq \sum_{k=1}^{K} \theta V + \frac{\beta^2 QK}{4V} + V\sum_{k=1}^{K}\sum_{q=1}^{Q} \mathbb{E}\left|\left|v_t^{(k)} - v_{t-1}^{(k)}\right|\right|^2 \leq \theta KV + \frac{\beta^2 T}{4V} + \frac{\beta^2(1+\theta)^2 V^3 T}{4\alpha^2}.$$

Appendix E

APPENDIX OF CHAPTER 7

E.1 Missing proofs

E.1.1 Proof of Theorem 7.5.1

We first provide a lemma that will be used in the proof.

Lemma E.1.1 *If we apply PGA with the update rule* $x_{k+1} = Proj_{\mathcal{K}}\big(x_k + \frac{1}{L}\nabla f(x_k)\big)$ $\forall k \in [K]$ *to an* L*-smooth function* f*, the following holds:*

$$f(x_{k+1}) \geq f(x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \geq f(x_k).$$

Proof We can use the L-smoothness of f and write:

$$\begin{aligned} f(x_{k+1}) &\geq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle - \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\geq f(x_k) + L\|x_{k+1} - x_k\|^2 - \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= f(x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\geq f(x_k), \end{aligned}$$

where the second inequality follows from the optimality condition for x_{k+1}, i.e., $x_{k+1} = \arg\min_{z \in \mathcal{K}} \|z - x_k - \frac{1}{L}\nabla f(x_k)\|^2$. ■

First, consider the DR-submodular setting ($\mu = 0$). Note that for all $k \in [K]$, we can rewrite the update rule of PGA in the following equivalent way:

$$\begin{aligned} x_{k+1} &= \arg\max_{x \in \mathcal{K}} \big(f(x_k) + \langle \nabla f(x_k), x - x_k \rangle - \frac{L}{2}\|x - x_k\|^2\big) \\ &= \text{Proj}_{\mathcal{K}}\big(x_k + \frac{1}{L}\nabla f(x_k)\big). \end{aligned}$$

If we denote $h(x) := f(x_k) + \langle \nabla f(x_k), x - x_k \rangle - \frac{L}{2}\|x - x_k\|^2$, h is L-strongly concave. Therefore, we can write:

$$\begin{aligned} h(x^*) &\leq h(x_{k+1}) + \langle \nabla h(x_{k+1}), x^* - x_{k+1} \rangle - \frac{L}{2}\|x_{k+1} - x^*\|^2 \\ &\overset{\text{(a)}}{\leq} h(x_{k+1}) - \frac{L}{2}\|x_{k+1} - x^*\|^2, \end{aligned}$$

where (a) follows from the optimality condition for x_{k+1}, i.e., $x_{k+1} = \arg\max_{x \in \mathcal{K}} h(x)$. We can write:

$$\begin{aligned} f(x_{k+1}) &\overset{\text{(b)}}{\geq} h(x_{k+1}) \\ &\geq h(x^*) + \frac{L}{2}\|x_{k+1} - x^*\|^2 \\ &= f(x_k) + \langle \nabla f(x_k), x^* - x_k \rangle - \frac{L}{2}\|x_k - x^*\|^2 + \frac{L}{2}\|x_{k+1} - x^*\|^2 \\ &\overset{\text{(c)}}{\geq} f(x^*) - c_f f(x_k) + \frac{L}{2}\big(\|x_{k+1} - x^*\|^2 - \|x_k - x^*\|^2\big), \end{aligned}$$

where (b) follows from the L-smoothness of f and (c) uses the result of Lemma 7.5.1 with $z = x^*$, $x = x_k$ and $\mu = 0$. Rearranging the terms and taking the sum over $k \in [K]$, we obtain:

$$\begin{aligned} K(1 + c_f) f(x_{K+1}) &\overset{\text{(d)}}{\geq} \sum_{k=1}^{K} \big(f(x_{k+1}) + c_f f(x_k)\big) \\ &\geq K f(x^*) + \frac{L}{2}\|x_{K+1} - x^*\|^2 - \frac{L}{2}\|x_1 - x^*\|^2 \\ &\geq K f(x^*) - \frac{L}{2}\|x_1 - x^*\|^2, \end{aligned}$$

where (d) is due to Lemma E.1.1. Dividing both sides by $K(1 + c_f)$, we derive the result as stated.

Now, we move on to the general strongly DR-submodular setting with $\mu > 0$. Note that for all $k \in [K]$, we can equivalently write the update rule of PGA as follows:

$$\begin{aligned} x_{k+1} &= \arg\max_{x \in \mathcal{K}} \big(\langle \nabla f(x_k), x - x_k \rangle - \frac{L}{2}\|x - x_k\|^2\big) \\ &= \mathrm{Proj}_{\mathcal{K}}\big(x_k + \frac{1}{L}\nabla f(x_k)\big). \end{aligned}$$

Using the L-smoothness property of f, we can write:

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\geq \max_{x \in \mathcal{K}} \big(\langle \nabla f(x_k), x - x_k \rangle - \frac{L}{2} \|x - x_k\|^2 \big) \\ &= \frac{1}{2L} \|\nabla f(x_k)\|^2 - \frac{L}{2} \min_{x \in \mathcal{K}} \|x - x_k - \frac{1}{L} \nabla f(x_k)\|^2 \\ &= \frac{1}{2L} \|\nabla f(x_k)\|^2 - \frac{L}{2} \|x_{k+1} - x_k - \frac{1}{L} \nabla f(x_k)\|^2. \end{aligned} \tag{E.1}$$

Setting $z = x^*$ and $x = x_k$ in Lemma 7.5.1, we have:

$$\begin{aligned} \frac{\mu}{L}\big(f(x^*) - (1 + c_f) f(x_k)\big) &\leq \frac{\mu}{L}\big(\langle \nabla f(x_k), x^* - x_k \rangle - \frac{\mu}{2} \|x^* - x_k\|^2 \big) \\ &= -\frac{\mu^2}{2L} \|x^* - x_k - \frac{1}{\mu} \nabla f(x_k)\|^2 + \frac{1}{2L} \|\nabla f(x_k)\|^2. \end{aligned} \tag{E.2}$$

Since $x_{k+1} = \arg\min_{z \in \mathcal{K}} \|z - x_k - \frac{1}{L} \nabla f(x_k)\|^2$, $0 \leq \frac{\mu}{L} \leq 1$ and $\mathcal{K}$ is convex, we can write:

$$\begin{aligned} \|x_{k+1} - x_k - \frac{1}{L} \nabla f(x_k)\|^2 &\leq \|\frac{\mu}{L} x^* + (1 - \frac{\mu}{L}) x_k - x_k - \frac{1}{L} \nabla f(x_k)\|^2 \\ &= \|\frac{\mu}{L}(x^* - x_k) - \frac{1}{L} \nabla f(x_k)\|^2 \\ &= \frac{\mu^2}{L^2} \|x^* - x_k - \frac{1}{\mu} \nabla f(x_k)\|^2. \end{aligned}$$

Multiplying both sides of the above inequality by $-\frac{L}{2}$, we obtain:

$$-\frac{\mu^2}{2L} \|x^* - x_k - \frac{1}{\mu} \nabla f(x_k)\|^2 \leq -\frac{L}{2} \|x_{k+1} - x_k - \frac{1}{L} \nabla f(x_k)\|^2. \tag{E.3}$$

Combining the inequalities E.1, E.2 and E.3, we have:

$$\frac{\mu}{L}\big(f(x^*) - (1 + c_f) f(x_k)\big) \leq f(x_{k+1}) - f(x_k).$$

Therefore, we can write:

$$\begin{aligned} f(x^*) - (1 + c_f) f(x_{k+1}) &\leq (1 - \frac{\mu}{L})\big(f(x^*) - (1 + c_f) f(x_k)\big) + c_f\big(f(x_k) - f(x_{k+1})\big) \\ &\leq (1 - \frac{\mu}{L})\big(f(x^*) - (1 + c_f) f(x_k)\big), \end{aligned}$$

where the last inequality uses the result of Lemma E.1.1. Applying the above inequality recursively for all $k \in [K]$, we obtain:

$$f(x^*) - (1 + c_f) f(x_{K+1}) \leq (1 - \frac{\mu}{L})^K \big(f(x^*) - (1 + c_f) f(x_1)\big)$$

$$\leq e^{-\mu K/L}\big(f(x^*) - (1+c_f)f(x_1)\big).$$

Rearranging the terms and dividing both sides by $1 + c_f$, we obtain the result as stated.

E.1.2 Proof of Theorem 7.5.2

We can write:

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &\leq \|x_t + \eta_t \nabla f_t(x_t) - x^*\|^2 \\ &= \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f_t(x_t)\|^2 - 2\eta_t \langle \nabla f_t(x_t), x^* - x_t\rangle.\end{aligned}$$

Rearranging the terms and dividing both sides by $2\eta_t$, we can equivalently write:

$$\begin{aligned}\langle \nabla f_t(x_t), x^* - x_t\rangle &\leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 + \eta_t^2 \|\nabla f_t(x_t)\|^2}{2\eta_t} \\ &\leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta_t} + \frac{\eta_t \beta^2}{2}. \end{aligned} \tag{E.4}$$

Using the result of Lemma 7.5.1 with $x = x_t$, $z = x^*$, $c_f = \max_{t\in[T]} c_{f_t}$, and $\mu = 0$ and combining it with the above inequality, we have:

$$f_t(x^*) - (1+c_f)f_t(x_t) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta_t} + \frac{\eta_t\beta^2}{2}.$$

Setting $\eta_t = \eta = \frac{R}{\beta\sqrt{T}}$ $\forall t \in [T]$ and taking the sum over $t \in [T]$, we obtain:

$$\begin{aligned}\sum_{t=1}^{T}\big(f_t(x^*) - (1+c_f)f_t(x_t)\big) &\leq \sum_{t=1}^{T} \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta} + \frac{\eta\beta^2 T}{2} \\ &= \frac{\|x_1 - x^*\|^2 - \|x_{T+1} - x^*\|^2}{2\eta} + \frac{\eta\beta^2 T}{2} \\ &\leq \frac{R^2}{2\eta} + \frac{\eta\beta^2 T}{2}.\end{aligned}$$

Plugging in the value of η and dividing both sides by $1 + c_f$, we conclude:

$$\sum_{t=1}^{T}\big(\frac{1}{1+c_f} f_t(x^*) - f_t(x_t)\big) \leq \frac{R\beta\sqrt{T}}{2(1+c_f)} + \frac{R\beta\sqrt{T}}{2(1+c_f)}$$

$$= \frac{R\beta\sqrt{T}}{1+c_f}.$$

For the setting where all $\{f_t\}_{t=1}^T$ are μ-strongly DR-submodular, we can combine the result of Lemma 7.5.1 and inequality E.4 to obtain the following:

$$f_t(x^*) - (1+c_f)f_t(x_t) \leq \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{2\eta_t} - \frac{\mu}{2}\|x_t - x^*\|^2 + \frac{\eta_t\beta^2}{2}.$$

If we set $\eta_t = \frac{1}{\mu t}$ $\forall t \in [T]$, we have $\frac{1}{2\eta_1} - \frac{\mu}{2} = 0$ and $\frac{1}{2\eta_t} - \frac{\mu}{2} = \frac{\mu(t-1)}{2} = \frac{1}{2\eta_{t-1}}$ $\forall t > 1$. Therefore, we can rewrite the above inequality in the following equivalent way:

$$f_t(x^*) - (1+c_f)f_t(x_t) \leq \frac{\|x_t - x^*\|^2}{2\eta_{t-1}} - \frac{\|x_{t+1} - x^*\|^2}{2\eta_t} + \frac{\eta_t\beta^2}{2}.$$

Taking the sum over $t \in [T]$, we have:

$$\begin{aligned}
\sum_{t=1}^T \big(f_t(x^*) - (1+c_f)f_t(x_t)\big) &\leq -\frac{1}{2\eta_1}\|x_2 - x^*\|^2 + \sum_{t=2}^T \big(\frac{\|x_t - x^*\|^2}{2\eta_{t-1}} - \frac{\|x_{t+1} - x^*\|^2}{2\eta_t}\big) + \frac{\beta^2}{2}\sum_{t=1}^T \eta_t \\
&= -\frac{1}{2\eta_1}\|x_2 - x^*\|^2 + \frac{1}{2\eta_1}\|x_2 - x^*\|^2 - \frac{1}{2\eta_T}\|x_{T+1} - x^*\|^2 + \frac{\beta^2}{2}\sum_{t=1}^T \eta_t \\
&\leq \frac{\beta^2}{2}\sum_{t=1}^T \eta_t \\
&\leq \frac{\beta^2}{2\mu}(1 + \ln T).
\end{aligned}$$

Dividing both sides by $1 + c_f$, we obtain the regret bound as follows:

$$\sum_{t=1}^T \big(\frac{1}{1+c_f} f_t(x^*) - f_t(x_t)\big) \leq \frac{\beta^2}{2\mu(1+c_f)}(1 + \ln T).$$

Appendix F

APPENDIX OF CHAPTER 8

F.1 Related work

Online optimization in the i.i.d. model. For the setting where the objective functions $\{f_t\}_{t=1}^T$ are drawn i.i.d. from an unknown distribution $\mathcal{D}$ with a convex mean $f(\cdot) = \mathbb{E}_{\mathcal{D}}[f_t(\cdot)]$, several algorithms have been proposed that could be divided into two categories of projection-based and projection-free algorithms where the latter is most relevant to our work. In particular, [105] proposed the Online Frank-Wolfe (OFW) algorithm which achieves a nearly optimal $\tilde{\mathcal{O}}(\sqrt{T})$ stochastic regret bound with high probability. The OFW algorithm requires to access the exact gradient of $\{f_t\}_{t=1}^T$ and has an average $\mathcal{O}(T)$ per-iteration computational cost. To remedy this issue, more recently, [60] proposed the One-Shot Frank-Wolfe (OSFW) algorithm with $\mathcal{O}(T^{2/3})$ stochastic regret bound in expectation. Note that the OSFW algorithm obtains similar bounds for the setting where f is monotone continuous DR-submodular. The OSFW algorithm only uses unbiased stochastic gradient estimates of loss functions and has an $\mathcal{O}(1)$ cost per iteration. However, the derived $\mathcal{O}(T^{2/3})$ stochastic regret bound is sub-optimal. To bridge this gap, [208] introduced the Online stochastic Recursive Gradient-based Frank-Wolfe (ORGFW) algorithm which not only has a nearly optimal $\tilde{O}(\sqrt{T})$ stochastic regret bound with high probability, but it also maintains a low $\mathcal{O}(1)$ computational cost per round.

Stochastic optimization. Online optimization in the i.i.d. model is closely related but different from the stochastic optimization problem [29]. In online optimization in the i.i.d. model, the goal is to choose a *sequence* $\{x_t\}_{t=1}^T$ of decision variables that has low stochastic regret, and the algorithm requires updating the decision variables as soon as new data arrives online [60]. In contrast, stochastic optimization focuses on the quality of the *final* output of the algorithm [152] and aims to find an approximate optimal point of the underlying objective function, where the performance is measured by the convergence rate.

Online convex optimization in the random order model. [92] studied OCO in a weaker model where the loss functions are still chosen adversarially but their order of arrival is random. This model is termed Random Order Online Convex Optimization (ROOCO). ROOCO is a natural middle ground between the standard adversarial OCO setting and the model where the loss functions are i.i.d. drawn from some unknown underlying distribution. In [92], the authors assumed that the average of the sequence of loss functions is α-strongly convex while each loss function may not even be convex. They showed that if all the loss functions are quadratic, with probability at least $1-\delta$, Online Gradient Descent (OGD) with step size $\eta_t = \frac{1}{\alpha t}$ $\forall t \in [T]$ obtains a regret bound of $\mathcal{O}(\frac{\beta^2}{\alpha^3}\log^2 T)$, where β is the Lipschitz constant of the loss functions. Moreover, for general loss functions, OGD with the same choice of step sizes obtains a regret bound of $\mathcal{O}(\frac{n\beta^2}{\alpha^3}\log^2 T)$ (n is the dimension of the domain space) with probability at least $1-\delta$. More recently, [191] provided a new algorithm with an improved $\mathcal{O}(\frac{\beta^2}{\alpha}\log T)$ for the problem.

Offline DR-submodular maximization. [25] proposed a variant of the Frank-Wolfe algorithm for maximizing a monotone DR-submodular function f subject to a convex domain $\mathcal{K}$. Starting from $x^{(1)} = 0$, the algorithm performs K Frank-Wolfe updates where at each iteration $k \in [K]$, it finds v_k such that $v_k = \arg\max_{x\in\mathcal{K}}\langle x, \nabla f(x^{(k)})\rangle$, and performs the update $x^{(k+1)} = x^{(k)} + \frac{1}{K}v_k$. [25] showed that the output of this algorithm ($x^{(K+1)}$) obtains the provably optimal approximation ratio of $1-\frac{1}{e}$.

F.2 Missing proofs

F.2.1 Proof of Theorem 8.4.1

For all $t \in [T]$, using the L-smoothness of f_t, we have:

$$f_t(x_t^{(k+1)}) \geq f_t(x_t^{(k)}) + \frac{1}{K}\langle v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle - \frac{L}{2K^2}\|v_t^{(k)}\|^2.$$

Taking the sum over $t \in [T]$, we obtain:

$$\sum_{t=1}^{T} \big(f_t(x_t^{(k+1)}) - f_t(x_t^{(k)})\big) \geq \frac{1}{K}\sum_{t=1}^{T}\langle v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle - \frac{L}{2K^2}\sum_{t=1}^{T}\|v_t^{(k)}\|^2.$$

For all $k \in [K]$, let $v_k^{(*)} = \arg\max_{v\in\mathcal{K}}\big(\sum_{t=1}^{T}\langle v, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu T}{2}\|v\|^2\big)$. If we use Follow the Leader (FTL) as $\mathcal{E}_k$ $\forall k \in [K]$, using Corollary 7.18 of [167], we can write:

$$\begin{aligned}\sum_{t=1}^{T}\langle x^*, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu T}{2}\|x^*\|^2 &\overset{\text{(a)}}{\leq} \sum_{t=1}^{T}\langle v_k^{(*)}, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu T}{2}\|v_k^{(*)}\|^2\\ &\overset{\text{(b)}}{\leq} \sum_{t=1}^{T}\big(\langle v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu}{2}\|v_t^{(k)}\|^2\big) + \frac{(\beta+\mu R)^2}{2\mu}(1+\ln T),\end{aligned}$$

where (a) uses the definition of $v_k^{(*)}$ and (b) is due to the regret bound of the FTL algorithm. We have:

$$\begin{aligned}\sum_{t=1}^{T}\big(f_t(x^*) - f_t(x_t^{(k)})\big) &\overset{\text{(c)}}{\leq} \sum_{t=1}^{T}\big(f_t(x^* + x_t^{(k)}) - f_t(x_t^{(k)})\big)\\ &\overset{\text{(d)}}{\leq} \sum_{t=1}^{T}\langle x^*, \nabla f_t(x_t^{(k)})\rangle - \frac{\mu T}{2}\|x^*\|^2,\end{aligned}$$

where (c) and (d) use the monotonicity and μ-strong DR-submodularity of $\{f_t\}_{t=1}^{T}$. Combining the above inequalities, we have:

$$\begin{aligned}\sum_{t=1}^{T}\big(f_t(x_t^{(k+1)}) - f_t(x_t^{(k)})\big) &\geq \frac{1}{K}\sum_{t=1}^{T}\big(f_t(x^*) - f_t(x_t^{(k)})\big) + \frac{\mu}{2K}\sum_{t=1}^{T}\|v_t^{(k)}\|^2 - \frac{L}{2K^2}\sum_{t=1}^{T}\|v_t^{(k)}\|^2\\ &\quad - \frac{(\beta+\mu R)^2}{2\mu K}(1+\ln T)\\ &= \frac{1}{K}\sum_{t=1}^{T}\big(f_t(x^*) - f_t(x_t^{(k)})\big) + \frac{1}{2K}\big(\mu - \frac{L}{K}\big)\sum_{t=1}^{T}\|v_t^{(k)}\|^2\\ &\quad - \frac{(\beta+\mu R)^2}{2\mu K}(1+\ln T).\end{aligned}$$

Therefore, if we set $K = \lceil \frac{L}{\mu} \rceil$, we can write:

$$\sum_{t=1}^{T}\big(f_t(x_t^{(k+1)}) - f_t(x^*)\big) \geq (1 - \frac{1}{K})\sum_{t=1}^{T}\big(f_t(x_t^{(k)}) - f_t(x^*)\big) - \frac{(\beta+\mu R)^2}{2\mu K}(1+\ln T).$$

Applying the inequality recursively for all $k \in [K]$, we obtain:

$$\sum_{t=1}^{T} \big(f_t(\underbrace{x_t^{(K+1)}}_{=x_t}) - f_t(x^*)\big) \geq (1 - \frac{1}{K})^K \sum_{t=1}^{T} \big(\underbrace{f_t(x_t^{(1)})}_{=0} - f_t(x^*)\big) - \frac{(\beta + \mu R)^2}{2\mu}(1 + \ln T).$$

Rearranging the terms and using the inequality $(1 - \frac{1}{K})^K \leq \frac{1}{e}$, we can write:

$$\sum_{t=1}^{T} \big((1 - \frac{1}{e}) f_t(x^*) - f_t(x_t)\big) \leq \frac{(\beta + \mu R)^2}{2\mu}(1 + \ln T).$$

F.2.2 Proof of Theorem 8.5.2

For non-quadratic DR-submodular functions, the Hessian of the utility functions is not fixed and therefore, we have to ensure the strong DR-submodularity of every block of size W holds at every point $x \in \mathcal{K}$. We provide the analysis for the class of concave functions with negative dependence below (this analysis could be easily extended to more general DR-submodular functions). For all $t \in [T]$, let f_t be of the form of concave functions with negative dependence. In this case, $\nabla^2_{ii} f_t(x) = \frac{\partial^2 h_i^{(t)}}{\partial x_i^2}(x)$ for all $i \in [n]$. Note that for all $i \in [n]$, we are assuming $\frac{1}{T}\sum_{t=1}^{T} h_i^{(t)}$ is μ-strongly concave while each individual $h_i^{(t)}$ may be neither convex nor concave. Assume that for all $i \in [n]$ and $\tau \in \{0, \ldots, \lceil \frac{T}{W} \rceil - 1\}$, the second derivative of the scalar function $\frac{1}{W}\sum_{t=\tau W+1}^{(\tau+1)W} h_i^{(t)}$ is H-Lipschitz. Also, assume that for all $x \in \mathcal{K}$, $0 \leq x_i \leq R_i$ holds. For all $i \in [n]$, consider the discretized set of points $\mathcal{Z}_i = \{2k\gamma \ \forall k \in \{0, \ldots, \frac{R_i}{2\gamma}\}\}$. Note that for all $x \in \mathcal{K}$ and $i \in [n]$, there exists $z_i \in \mathcal{Z}_i$ such that $|x_i - z_i| \leq \gamma$. If the desired property holds for this set of points in the domain $\mathcal{K}$, we can use the H-Lipschitzness assumption to conclude

$$\frac{1}{W} \sum_{t=\tau W+1}^{(\tau+1)W} (h_i^{(t)})''(x_i) \leq -(\mu - \epsilon - \gamma H).$$

Therefore, in order to take the union bound in this setting, we set $\frac{\delta}{T\sum_{i=1}^{n} \lceil \frac{R_i}{2\gamma} \rceil} = 4\exp\big(-\frac{W\epsilon^2}{128\theta L^2}\big)$ which is equivalent to $W_0 = \frac{128\theta L^2}{\epsilon^2} \ln(\frac{4T\sum_{i=1}^{n} R_i}{2\gamma\delta})$.

F.2.3 Proof of Theorem 8.6.1

For Algorithm 2, we can write:

$$\begin{aligned}
f(x_{t+1}^{(k+1)}) &\overset{\text{(a)}}{\geq} f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle v_t^{(k)}, \nabla f(x_{t+1}^{(k)})\rangle - \frac{L}{2K_t^2}\|v_t^{(k)}\|^2 \\
&\geq f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle v_t^{(k)}, d_t^{(k)}\rangle + \frac{1}{K_t}\langle v_t^{(k)}, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2} \\
&\overset{\text{(b)}}{\geq} f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle x^*, d_t^{(k)}\rangle + \frac{1}{K_t}\langle v_t^{(k)}, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2} \\
&= f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle x^*, \nabla f(x_{t+1}^{(k)})\rangle + \frac{1}{K_t}\langle v_t^{(k)} - x^*, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2} \\
&\overset{\text{(c)}}{\geq} f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle (x^* - x_{t+1}^{(k)}) \vee 0, \nabla f(x_{t+1}^{(k)})\rangle + \frac{1}{K_t}\langle v_t^{(k)} - x^*, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2} \\
&\overset{\text{(d)}}{\geq} f(x_{t+1}^{(k)}) + \frac{1}{K_t}f(x^* \vee x_{t+1}^{(k)}) - \frac{1}{K_t}f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle v_t^{(k)} - x^*, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2} \\
&\overset{\text{(e)}}{\geq} f(x_{t+1}^{(k)}) + \frac{1}{K_t}f(x^*) - \frac{1}{K_t}f(x_{t+1}^{(k)}) + \frac{1}{K_t}\langle v_t^{(k)} - x^*, \nabla f(x_{t+1}^{(k)}) - d_t^{(k)}\rangle - \frac{LR^2}{2K_t^2},
\end{aligned}$$

where (a) uses L-smoothness of f, (b) is due to the update rule of the algorithm, (c) and (e) follow from monotonicity of f, and (d) exploits concavity of f along non-negative directions. Defining $\epsilon_t^{(k)} := d_t^{(k)} - \nabla f(x_{t+1}^{(k)})$ and rearranging the terms in the above inequality, we have:

$$f(x^*) - f(x_{t+1}^{(k+1)}) \leq (1 - \frac{1}{K_t})\big(f(x^*) - f(x_{t+1}^{(k)})\big) + \frac{R}{K_t}\|\epsilon_t^{(k)}\| + \frac{LR^2}{2K_t^2}.$$

Applying the inequality recursively for all $k \in [K_t]$ and taking expectation of both sides, we have:

$$f(x^*) - \mathbb{E}[f(\underbrace{x_{t+1}^{(K_t+1)}}_{=x_{t+1}})] \leq \underbrace{(1 - \frac{1}{K_t})^{K_t}}_{\leq \frac{1}{e}}\big(f(x^*) - \mathbb{E}[f(\underbrace{x_{t+1}^{(1)}}_{=0})]\big) + \frac{R}{K_t}\sum_{k=1}^{K_t}\mathbb{E}\|\epsilon_t^{(k)}\| + \frac{LR^2}{2K_t}.$$

Rearranging the terms, taking the expectation of both sides and taking the sum over $t \in \{1, \ldots, T-1\}$, we obtain:

$$\mathbb{E}[(1 - \frac{1}{e})\text{-SR}_T] \leq \sum_{t=1}^{T-1}\frac{LR^2}{2K_t} + \sum_{t=1}^{T-1}\frac{R}{K_t}\sum_{k=1}^{K_t}\mathbb{E}\|\epsilon_t^{(k)}\| + \mathcal{O}(1). \tag{F.1}$$

We can bound $\mathbb{E}\|\epsilon_t^{(k)}\|$ as follows:

$$\begin{aligned}
\mathbb{E}\|\epsilon_t^{(k)}\| &= \mathbb{E}\sqrt{\|\epsilon_t^{(k)}\|^2} \\
&\leq \sqrt{\mathbb{E}\|\epsilon_t^{(k)}\|^2} \\
&= \sqrt{\mathbb{E}\big[\big(\frac{1}{t}\sum_{\tau=1}^{t}\tilde{\nabla} f_\tau(x_{t+1}^{(k)}) - \nabla f(x_{t+1}^{(k)})\big)^T\big(\frac{1}{t}\sum_{\tau=1}^{t}\tilde{\nabla} f_\tau(x_{t+1}^{(k)}) - \nabla f(x_{t+1}^{(k)})\big)\big]} \\
&= \sqrt{\frac{1}{t^2}\sum_{\tau=1}^{t}\mathbb{E}\|\tilde{\nabla} f_\tau(x_{t+1}^{(k)}) - \nabla f(x_{t+1}^{(k)})\|^2} \\
&\leq \sqrt{\frac{1}{t^2}\sum_{\tau=1}^{t}\sigma^2} \\
&= \frac{\sigma}{\sqrt{t}},
\end{aligned}$$

where the first inequality is due to Jensen's inequality. Therefore, if we set $K_t = \sqrt{t}\ \forall t \in [T]$, the expected regret bound of the algorithm is $\mathcal{O}((LR^2 + R\sigma)\sqrt{T})$.

We can also obtain a high probability regret bound for the algorithm. Considering that f is β-Lipschitz, we have $\|\tilde{\nabla} f_t(\cdot)\| \leq \|\nabla f(\cdot)\| + \|\tilde{\nabla} f_t(\cdot) - \nabla f(\cdot)\| \leq \beta + \sigma$. Therefore, we can use Corollary 7 of [115] to conclude that with probability at least $1 - \frac{\delta}{T^{3/2}}$, the following holds:

$$\|\epsilon_t^{(k)}\| \leq \mathcal{O}\big((\sigma + \beta)\sqrt{\frac{\ln(2T^{3/2}/\delta)}{t}}\big).$$

Taking the union bound, we can conclude that $(1-\frac{1}{e})\text{-SR}_T \leq \sum_{t=1}^{T-1}\frac{LR^2}{2K_t} + \mathcal{O}(\sigma\sqrt{T\ln(T^{3/2}/\delta)})$ holds with probability at least $1-\delta$. Thus, if we set $K_t = \sqrt{t}\ \forall t \in [T]$, we obtain an $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound, both in expectation and with high probability.

F.2.4 Proof of Theorem 8.6.2

Using the L-smoothness of f and the update rule of Algorithm 3, we have:

$$\begin{aligned}
f(x_{t+1}) &\overset{\text{(a)}}{\geq} f(x_t) + \frac{1}{T}\langle v_t, \nabla f(x_t)\rangle - \frac{L}{2T^2}\|v_t\|^2 \\
&\geq f(x_t) + \frac{1}{T}\langle v_t, d_t\rangle + \frac{1}{T}\langle v_t, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2}
\end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(b)}}{\geq} f(x_t) + \frac{1}{T}\langle x^*, d_t\rangle + \frac{1}{T}\langle v_t, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2} \\
&= f(x_t) + \frac{1}{T}\langle x^*, \nabla f(x_t)\rangle + \frac{1}{T}\langle v_t - x^*, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2} \\
&\overset{\text{(c)}}{\geq} f(x_t) + \frac{1}{T}\langle (x^* - x_t) \vee 0, \nabla f(x_t)\rangle + \frac{1}{T}\langle v_t - x^*, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2} \\
&\overset{\text{(d)}}{\geq} f(x_t) + \frac{1}{T} f(x^* \vee x_t) - \frac{1}{T} f(x_t) + \frac{1}{T}\langle v_t - x^*, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2} \\
&\overset{\text{(e)}}{\geq} f(x_t) + \frac{1}{T} f(x^*) - \frac{1}{T} f(x_t) + \frac{1}{T}\langle v_t - x^*, \nabla f(x_t) - d_t\rangle - \frac{LR^2}{2T^2},
\end{aligned}
$$

where (a) uses L-smoothness of f, (b) is due to the update rule of the algorithm, (c) and (e) follow from monotonicity of f, and (d) exploits concavity of f along non-negative directions. Defining $\epsilon_t := d_t - \nabla f(x_t)$ and rearranging the terms in the above inequality, we have:

$$
f(x^*) - f(x_{t+1}) \leq (1 - \frac{1}{T})\big(f(x^*) - f(x_t)\big) + \frac{R}{T}\|\epsilon_t^{(k)}\| + \frac{LR^2}{2T^2}.
$$

Applying the above inequality recursively, we have:

$$
f(x^*) - f(x_{t+1}) \leq \underbrace{(1 - \frac{1}{T})^t}_{\leq e^{-t/T}} \big(f(x^*) - \underbrace{f(x_1)}_{=0}\big) + \frac{R}{T}\sum_{s=1}^{t} \|\epsilon_s\| + \frac{LR^2}{2T}. \tag{F.2}
$$

Using inequality F.2 and $\sum_{t=1}^{T-1} e^{-t/T} \leq T(e^{-1/T} - \frac{1}{e}) \leq T(1 - \frac{1}{e})$, we have $(\frac{1}{e})\text{-SR}_T \leq \frac{LR^2}{2} + \frac{R}{T}\sum_{t=1}^{T-1}\sum_{s=1}^{t}\|\epsilon_s\|$. We can write:

$$
\begin{aligned}
\epsilon_t &= d_t - \nabla f(x_t) \\
&= (1 - \rho_t)\epsilon_{t-1} + \rho_t\big(\tilde{\nabla} f_t(x_t) - \nabla f(x_t)\big) \\
&\quad + (1 - \rho_t)\big(\tilde{\nabla} f_t(x_t) - \tilde{\nabla} f_t(x_{t-1}) - (\nabla f(x_t) - \nabla f(x_{t-1}))\big).
\end{aligned}
$$

Applying the above equality recursively, we obtain:

$$
\begin{aligned}
\epsilon_t = &\prod_{s=2}^{t}(1 - \rho_s)\epsilon_1 + \sum_{\tau=2}^{t}\prod_{s=\tau}^{t}(1 - \rho_s)\big(\tilde{\nabla} f_\tau(x_\tau) - \tilde{\nabla} f_\tau(x_{\tau-1}) - (\nabla f(x_\tau) - \nabla f(x_{\tau-1}))\big) \\
&+ \sum_{\tau=2}^{t}\rho_\tau \prod_{s=\tau+1}^{t}(1 - \rho_s)\big(\tilde{\nabla} f_\tau(x_\tau) - \nabla f(x_\tau)\big).
\end{aligned}
$$

Let $\epsilon_t = \sum_{\tau=1}^{t} \zeta_{t,\tau}$, where $\zeta_{t,1} = \prod_{s=2}^{t}(1-\rho_s)\epsilon_1$ and $\zeta_{t,\tau} = \prod_{s=\tau}^{t}(1-\rho_s)\big(\tilde{\nabla} f_\tau(x_\tau) - \tilde{\nabla} f_\tau(x_{\tau-1}) - (\nabla f(x_\tau) - \nabla f(x_{\tau-1}))\big) + \rho_\tau \prod_{s=\tau+1}^{t}(1-\rho_s)\big(\tilde{\nabla} f_\tau(x_\tau) - \nabla f(x_\tau)\big)$ for $\tau > 1$. Let $\mathcal{F}_{\tau-1}$ be the σ-field generated by $\{f_s\}_{s=1}^{\tau-1}$. Clearly, $\mathbb{E}[\zeta_{t,1}] = 0$. Also, for $\tau > 1$, we have:

$$\begin{aligned}
\mathbb{E}[\zeta_{t,\tau}|\mathcal{F}_{\tau-1}] &= \rho_\tau \prod_{s=\tau+1}^{t}(1-\rho_s)\big(\nabla f(x_\tau) - \nabla f(x_\tau)\big) \\
&+ \prod_{s=\tau}^{t}(1-\rho_s)\big(\nabla f(x_\tau) - \nabla f(x_{\tau-1}) - (\nabla f(x_\tau) - \nabla f(x_{\tau-1}))\big) \\
&= 0.
\end{aligned}$$

Therefore, for all $t \in [T]$, $\{\zeta_{t,\tau}\}_{\tau=1}^{t}$ is a martingale difference sequence.

For any $\tau \in [t]$, we can write:

$$\prod_{s=\tau}^{t}(1-\rho_s) = \prod_{s=\tau}^{t}(1-\frac{1}{s+1}) = \prod_{s=\tau}^{t}\frac{s}{s+1} = \frac{\tau}{t+1}.$$

Thus, we have $\|\zeta_{t,1}\| = \frac{2}{t+1}\|\tilde{\nabla} f_1(x_1) - \nabla f(x_1)\| \le \frac{2\sigma}{t+1}$. For $\tau > 1$, $\|\zeta_{t,\tau}\|$ could be bounded as follows:

$$\begin{aligned}
\|\zeta_{t,\tau}\| \le& \prod_{s=\tau}^{t}(1-\rho_s)\big(\|\tilde{\nabla} f_\tau(x_\tau) - \tilde{\nabla} f_\tau(x_{\tau-1})\| + \|\nabla f(x_\tau) - \nabla f(x_{\tau-1})\|\big) \\
&+ \rho_\tau \prod_{s=\tau+1}^{t}(1-\rho_s)\|\tilde{\nabla} f_\tau(x_\tau) - \nabla f(x_\tau)\| \\
&\le \frac{2L\tau}{t+1}\|\underbrace{x_\tau - x_{\tau-1}}_{\frac{1}{T}v_t}\| + \frac{\sigma}{t+1} \\
&\le \frac{2LR\tau/T + \sigma}{t+1} \\
&\le \frac{2LR + \sigma}{t+1}.
\end{aligned}$$

Using the concentration inequality for vector-valued martingales (Theorem 3.5 of [171]), we have:

$$\mathbb{P}\big(\|\epsilon_t\| \ge \lambda_t\big) \le 4\exp\big(-\frac{\lambda_t^2}{(\frac{2\sigma}{t+1})^2 + (t-1)(\frac{2LR+\sigma}{t+1})^2}\big)$$

$$\leq 4\mathrm{exp}\big(-\frac{\lambda_t^2(t+1)}{(2LR+2\sigma)^2}\big).$$

Therefore, if we set $\lambda_t = \frac{(2LR+2\sigma)\sqrt{\ln(4T/\delta)}}{\sqrt{t+1}}$, for all $t \in [T]$, with probability at least $1-\delta$, the following holds:

$$\|\epsilon_t\| \leq \frac{(2LR+2\sigma)\sqrt{\ln(4T/\delta)}}{\sqrt{t+1}}.$$

Thus, with probability at least $1-\delta$, the regret bound is $\mathcal{O}(\sigma\sqrt{T\ln(T/\delta)})$.

We can also obtain the expected regret bound of the algorithm. We have:

$$\begin{aligned}\mathbb{E}\|\epsilon_t\| &= \int_{\lambda=0}^{\infty} \mathbb{P}(\|\epsilon_t\| \geq \lambda)d\lambda \\ &\leq \int_{\lambda=0}^{\infty} 4\mathrm{exp}\big(-\frac{\lambda^2(t+1)}{(2LR+2\sigma)^2}\big)d\lambda \\ &= \int_{x=0}^{\infty} 4\mathrm{exp}(-x^2)\frac{2LR+2\sigma}{\sqrt{t+1}}dx \\ &= \frac{4\sqrt{\pi}(LR+\sigma)}{\sqrt{t+1}}.\end{aligned}$$

Therefore, using inequality F.2, the expected regret bound is $\mathcal{O}(\sigma\sqrt{T})$.

Appendix G

APPENDIX OF CHAPTER 9

G.1 DPCG algorithm with Gaussian noise

For all $k \in [K]$, let $Y^{(k)} \sim \mathcal{N}(0, \sigma^2 I)$. In order to compute the ℓ_2-sensitivity of the gradient, We can write:

$$\begin{aligned}
\|\nabla f_D(x) - \nabla f_{D'}(x)\|_2 &= \sqrt{\sum_{i=1}^{|V|} |\nabla_i f_D(x) - \nabla_i f_{D'}(x)|^2} \\
&\leq \sqrt{\sum_{i=1}^{|V|} (2\Delta)^2} \\
&= 2\sqrt{|V|}\Delta.
\end{aligned}$$

The Gaussian mechanism combined with the basic composition theorem provides the following privacy guarantee for the DPCG algorithm with Gaussian noise.

Theorem G.1.1 *[78] Let $\epsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25K/\delta)$, the DPCG algorithm under Gaussian noise with parameter $\sigma \geq \frac{2cK\sqrt{|V|}\Delta}{\epsilon}$ is (ϵ, δ)-differentially private.*

We now analyze the approximation guarantee in this setting. First, we remind the reader that the following holds using Lemma 9.4.1 of the paper:

$$\mathbb{E}[f(x)] \geq (1 - \frac{1}{e})f(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K}.$$

If $\mathcal{D} = \mathcal{N}(0, \sigma^2 I)$, we have $G_{\mathcal{D}} \leq 2\text{rank}(\mathcal{M})\mathbb{E}_{Y\sim\mathcal{N}(0,\sigma^2 I)}\|Y\|_\infty$ and $G_{\mathcal{D}} \leq \frac{2}{c_{\min}}\mathbb{E}_{Y\sim\mathcal{N}(0,\sigma^2 I)}\|Y\|_\infty$ for matroid and knapsack constraints respectively. For a $|V|$-dimensional Gaussian random vector $Y \sim \mathcal{N}(0, \sigma^2 I)$, we can write:

$$\mathbb{E}\|Y\|_\infty \leq \mathcal{O}(\sigma\sqrt{\ln(|V|)}),$$

$$\mathbb{P}\big(\|Y\|_\infty - \sigma\sqrt{2\ln(|V|)} \leq 2\sigma\sqrt{\ln(K)}\big) \geq 1 - \frac{1}{K^2}.$$

Combining the above results and setting $\sigma = \frac{2cK\sqrt{|V|}\Delta}{\epsilon}$ for $c^2 > 2\ln(1.25K/\delta)$, the following holds for matroid and knapsack constraints respectively:

$$\mathbb{E}[f(x)] \geq (1-\frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{\text{rank}(\mathcal{M})K\sqrt{|V|\ln(|V|)\ln(K/\delta)}\Delta}{\epsilon}),$$
$$\mathbb{E}[f(x)] \geq (1-\frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{K\sqrt{|V|\ln(|V|)\ln(K/\delta)}\Delta}{c_{\min}\epsilon}).$$

Also, with probability at least $1 - \frac{1}{K}$, we have:

$$f(x) \geq (1-\frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{\text{rank}(\mathcal{M})K\sqrt{|V|\ln(\max\{|V|,K\})\ln(K/\delta)}\Delta}{\epsilon}),$$
$$f(x) \geq (1-\frac{1}{e})f(x^*) - \frac{LR^2}{2K} - \mathcal{O}(\frac{K\sqrt{|V|\ln(\max\{|V|,K\})\ln(K/\delta)}\Delta}{c_{\min}\epsilon}).$$

Compared to the Laplace noise, the additive factor in the approximation guarantee using the Gaussian noise is smaller by an order of $\sqrt{|V|\ln(|V|)}$. However, this improved accuracy comes at the price of achieving (ϵ,δ)-differential privacy, as opposed to ϵ-differential privacy using the Laplace noise.

G.2 Algorithm 1 of [3]

The DPMFW algorithm exploits Algorithm 1 of [3] for differentially private online linear optimization as a sub-routine. We explain this algorithm in more detail below. The algorithm is provided in Algorithm 4. Consider an online linear optimization problem over T rounds where at each round $t \in [T]$, the algorithm chooses an action $x_t \in \mathcal{X}$, $\mathcal{X}$ is the fixed domain set, and upon committing to this action, a loss vector ℓ_t is revealed and the algorithm incurs the loss $\langle \ell_t, x_t\rangle$. Algorithm 4 is identical to the well-known FTRL algorithm except for the fact that instead of $\sum_{s=1}^{t-1}\ell_s$, a noisy partial sum of the loss vectors $\tilde{L}_{t-1}$ is used in the update rule. This noisy partial sum is obtained using the Tree-Based Aggregation Protocol (TBAP) which was used in prior works as well [77, 112].

Algorithm 4 FTRL template for Online Linear Optimization [3]

```
Input: Noise distribution $\mathcal{D}$, regularizer $g(x)$.
Initialize an empty binary tree $B$ to compute differentially private estimates of $\sum_{s=1}^{t} \ell_s$.
Sample $n_0^1, \ldots, n_0^{\lceil \ln T \rceil}$ independently from $\mathcal{D}$.
$\tilde{L}_0 = \sum_{i=1}^{\lceil \ln T \rceil} n_0^i$.
for $t = 1, \ldots, T$ do
    Choose $x_t = \arg\min_{x \in \mathcal{X}} \big(\eta \langle x, \tilde{L}_{t-1} \rangle + g(x)\big)$.
    Observe $\ell_t$ and suffer a loss of $\langle \ell_t, x_t \rangle$.
    $(\tilde{L}_t, B) = \mathrm{TBAP}(\ell_t, B, t, \mathcal{D}, T)$.
end for
```

G.3 Missing proofs

G.3.1 Proof of Lemma 9.4.2

The upper bounds for R follow from $\|x\|_1 \leq \mathrm{rank}(\mathcal{M})$, $\forall x \in P(\mathcal{M})$ and $\|x\|_1 \leq \frac{1}{c_{\min}}$, $\forall x \in \{x \in [0,1]^{|V|} : c^T x \leq 1\}$. Consider the (i,j)-th entry of the Hessian of f. Let $m_F = \max_{i \in V} F(\{i\})$. We can write:

$$\begin{aligned}
|\nabla^2_{i,j} f(z)| &= |\mathbb{E}_{R\sim z}\big[F(R \cup \{i,j\}) - F(R \cup \{i\} \setminus \{j\}) - F(R \cup \{j\} \setminus \{i\}) + F(R \setminus \{i,j\})\big]| \\
&\leq \max\{F(\{i\}), F(\{j\})\} \\
&\leq m_F.
\end{aligned}$$

Thus, for all $k \in [K]$ and $j \in V$, using the mean-value theorem, we have:

$$\begin{aligned}
|\nabla_j f(x^{(k)} + \frac{1}{K} v_k) - \nabla_j f(x^{(k)})| &\leq \frac{1}{K} m_F |1^T v_k| \\
&= m_F \|\frac{1}{K} v_k\|_1.
\end{aligned}$$

Therefore, we can conclude $\|\nabla f(x^{(k)} + \frac{1}{K} v_k) - \nabla f(x^{(k)})\|_\infty \leq m_F \|\frac{1}{K} v_k\|_1$ and thus, $L \leq m_F$.

G.3.2 Proof of Lemma 9.4.5

First, note that by definition, the function g is monotone DR-submodular. Thus, similar to the proof of Lemma 9.4.1, we can write:

$$\begin{aligned}
g(x^{(k+1)}) - g(x^{(k)}) &\overset{(a)}{\geq} \frac{1}{K}\langle v_k, \nabla g(x^{(k)})\rangle - \frac{L}{2K^2}\|v_k\|_1^2 \\
&\overset{(b)}{\geq} \frac{1}{K}\langle x^*, \nabla g(x^{(k)})\rangle + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\
&\overset{(c)}{\geq} \frac{1}{K}\langle (x^* - x^{(k)}) \vee 0, \nabla g(x^{(k)})\rangle + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\
&\overset{(d)}{\geq} \frac{1}{K}\big(g(x^* \vee x^{(k)}) - g(x^{(k)})\big) + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2} \\
&\overset{(e)}{\geq} \frac{1}{K}\big(g(x^*) - g(x^{(k)})\big) + \frac{1}{K}\langle x^* - v_k, Y^{(k)}\rangle - \frac{LR^2}{2K^2},
\end{aligned}$$

where (a) is due to the L-smoothness of g, (b) follows from the update rule of the algorithm, (c) and (e) use the monotonicity of g, and (d) exploits concavity of g along non-negative directions. Using the definition of $G_{\mathcal{D}}$, if we take the expectation of both sides, and apply the inequality recursively for all $k \in [K]$, we obtain:

$$\mathbb{E}[g(x^{(K+1)})] - g(x^*) \geq (1 - \frac{1}{K})^K \big(\mathbb{E}[g(\underbrace{x^{(1)}}_{=0})] - g(x^*)\big) - G_{\mathcal{D}} - \frac{LR^2}{2K}.$$

Rearranging the terms and using the inequality $(1 - \frac{1}{K})^K \leq \frac{1}{e}$, we can write:

$$\mathbb{E}[g(x)] \geq (1 - \frac{1}{e})g(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K}.$$

Using the update rule of the algorithm, we have:

$$\ell^T x = \ell^T x^{(K+1)} = \frac{1}{K}\sum_{k=1}^{K} \ell^T v_k \geq \frac{1}{K}\sum_{k=1}^{K} \lambda = \lambda = \ell^T x^*,$$

where the inequality is due to the update rule of the algorithm for v_k. Also, using the definition of ℓ and DR-submodularity of f, the following holds:

$$\ell^T x^* = \sum_{i \in [|V|]} x_i^* \ell_i$$

$$\geq (1-\kappa_F) \sum_{i \in [|V|]} x_i^* \nabla_i f(0)$$
$$\geq (1-\kappa_F) f(x^*).$$

Putting the above inequalities together, we have:

$$\begin{aligned}
\mathbb{E}[f(x)] &= \mathbb{E}[g(x)] + \mathbb{E}[\ell^T x] \\
&\geq (1-\frac{1}{e})g(x^*) + \ell^T x^* - G_{\mathcal{D}} - \frac{LR^2}{2K} \\
&\geq (1-\frac{1}{e})f(x^*) - (1-\frac{1}{e})\ell^T x^* + \ell^T x^* - G_{\mathcal{D}} - \frac{LR^2}{2K} \\
&= (1-\frac{1}{e})f(x^*) + \frac{1}{e}\ell^T x^* - G_{\mathcal{D}} - \frac{LR^2}{2K} \\
&\geq (1-\frac{1}{e})f(x^*) + \frac{1}{e}(1-\kappa_F)f(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K} \\
&\geq (1-\frac{\kappa_F}{e})f(x^*) - G_{\mathcal{D}} - \frac{LR^2}{2K}.
\end{aligned}$$

G.3.3 Proof of Theorem 9.5.1

Similar to the offline setting, assume that all utility functions $\{f_t\}_{t=1}^T$ are monotone DR-submodular and L-smooth with respect to the norm $\|\cdot\|_1$. We can write:

$$\begin{aligned}
f_t(x_t^{(k+1)}) &\geq f_t(x_t^{(k)}) + \frac{1}{K}\langle v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle - \frac{L}{2K^2}\|v_t^{(k)}\|_1^2 \\
&\geq f_t(x_t^{(k)}) + \frac{1}{K}\langle v_t^{(k)} - x^*, \nabla f_t(x_t^{(k)})\rangle + \frac{1}{K}\langle x^*, \nabla f_t(x_t^{(k)})\rangle - \frac{LR^2}{2K^2}.
\end{aligned}$$

We can use the DR-submodularity and monotonicity of the utility function f_t to write:

$$\begin{aligned}
\langle x^*, \nabla f_t(x_t^{(k)})\rangle &\geq \langle (x^* - x_t^{(k)}) \vee 0, \nabla f_t(x_t^{(k)})\rangle \\
&\geq f_t(x^* \vee x_t^{(k)}) - f_t(x_t^{(k)}) \\
&\geq f_t(x^*) - f_t(x_t^{(k)}).
\end{aligned}$$

Combining the above inequalities, we have:

$$f_t(x_t^{(k+1)}) \geq f_t(x_t^{(k)}) + \frac{1}{K}f_t(x^*) - \frac{1}{K}f_t(x_t^{(k)}) + \frac{1}{K}\langle v_t^{(k)} - x^*, \nabla f_t(x_t^{(k)})\rangle - \frac{LR^2}{2K^2}.$$

Rearranging the terms and taking sum over $t \in [T]$, we obtain:

$$\sum_{t=1}^{T}\big(f_t(x^*) - f_t(x_t^{(k+1)})\big) \leq (1 - \frac{1}{K})\sum_{t=1}^{T}\big(f_t(x^*) - f_t(x_t^{(k)})\big) + \frac{1}{K}\sum_{t=1}^{T}\langle x^* - v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle + \frac{LR^2T}{2K^2}.$$

Applying the above inequality recursively for all $k \in [K]$, we have:

$$\sum_{t=1}^{T}\big(f_t(x^*) - f_t(\underbrace{x_t^{(K+1)}}_{=x_t})\big) \leq \underbrace{(1 - \frac{1}{K})^K}_{\leq 1/e}\sum_{t=1}^{T}\big(f_t(x^*) - f_t(\underbrace{x_t^{(1)}}_{=0})\big) + \frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\langle x^* - v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle + \frac{LR^2T}{2K}.$$

Rearranging the terms, we can equivalently write:

$$R_T \leq \frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\langle x^* - v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle + \frac{LR^2T}{2K}.$$

Using Theorem 3.1 of [3] with the regularizer $g(x) = \sum_{i=1}^{|V|} x_i \ln(x_i)$, we have the following for all $k \in [K]$:

$$\mathbb{E}\big[\sum_{t=1}^{T}\langle x^* - v_t^{(k)}, \nabla f_t(x_t^{(k)})\rangle\big] \leq \mathcal{O}(R\sqrt{T \ln |V|}) + W_{\mathcal{D}},$$

where $W_{\mathcal{D}} := \mathbb{E}_{Z\sim\mathcal{D}'}\big[\max_{x\in P}\langle Z, x\rangle - \min_{x\in P}\langle Z, x\rangle\big]$ and $\mathcal{D}'$ is the distribution induced by the sum of $\lceil \ln T\rceil$ independent samples from $\mathcal{D} = \mathrm{Lap}^{|V|}(\lambda)$ or $\mathcal{D} = \mathcal{N}(0, \sigma^2 I)$. For matroid constraints, we can write:

$$\begin{aligned}\max_{x\in P}\langle Z, x\rangle - \min_{x\in P}\langle Z, x\rangle &\leq \|x\|_1\|Z\|_\infty + \|x\|_1\|Z\|_\infty \\ &= 2\|x\|_1\|Z\|_\infty \\ &\leq 2\mathrm{rank}(\mathcal{M})\lceil \ln T\rceil\|Y\|_\infty,\end{aligned}$$

where $Y \sim \mathcal{D}$. Therefore, $W_{\mathrm{Lap}} \leq 2\mathrm{rank}(\mathcal{M})\lceil \ln T\rceil\mathbb{E}\|Y\|_\infty$ holds. Similarly, we have $W_{\mathrm{Lap}} \leq \frac{2}{c_{\min}}\lceil \ln T\rceil\mathbb{E}\|Y\|_\infty$ for knapsack constraints. If $\mathcal{D} = \mathrm{Lap}^{|V|}(\lambda)$, we have:

$$\mathbb{E}\|Y\|_\infty \leq \mathcal{O}(\lambda \ln(|V|)).$$

If $\mathcal{D} = \mathcal{N}(0, \sigma^2 I)$, the following holds:

$$\mathbb{E}\|Y\|_\infty \leq \mathcal{O}(\sigma\sqrt{\ln(|V|)}).$$

Setting $\lambda = \frac{2m_F|V|\ln T\sqrt{2K\ln(1/\delta)}}{\epsilon}$ and using the result of Lemma 9.4.2, we have the following regret bound using the Laplace noise and under matroid and knapsack constraints respectively.

$$\mathbb{E}[R_T] \leq \mathcal{O}(\text{rank}(\mathcal{M})\sqrt{T\ln|V|}) + \frac{m_F(\text{rank}(\mathcal{M}))^2 T}{2K} + \mathcal{O}(\frac{\text{rank}(\mathcal{M})|V|\ln|V|\ln^2 T\sqrt{K\ln(1/\delta)}}{\epsilon}),$$
$$\mathbb{E}[R_T] \leq \mathcal{O}(\frac{\sqrt{T\ln|V|}}{c_{\min}}) + \frac{m_F T}{2c_{\min}^2 K} + \mathcal{O}(\frac{|V|\ln|V|\ln^2 T\sqrt{K\ln(1/\delta)}}{c_{\min}\epsilon}).$$

Also, we can use the advanced composition theorem to conclude that the algorithm is (ϵ, δ)-differentially private. Setting $\sigma^2 = \frac{8\beta^2 K\ln(1/\delta)}{\epsilon^2}\ln^2 T\ln(\frac{K\ln T}{\delta'})$, the regret bound using the Gaussian noise for matroid and knapsack constraints are as follows:

$$\mathbb{E}[R_T] \leq \mathcal{O}(\text{rank}(\mathcal{M})\sqrt{T\ln|V|}) + \frac{m_F(\text{rank}(\mathcal{M}))^2 T}{2K} + \mathcal{O}(\frac{\text{rank}(\mathcal{M})\sqrt{\ln|V|}\ln^2 T\sqrt{K\ln(1/\delta)\ln(\frac{K\ln T}{\delta'})}}{\epsilon}),$$
$$\mathbb{E}[R_T] \leq \mathcal{O}(\frac{\sqrt{T\ln|V|}}{c_{\min}}) + \frac{m_F T}{2c_{\min}^2 K} + \mathcal{O}(\frac{\sqrt{\ln|V|}\ln^2 T\sqrt{K\ln(1/\delta)\ln(\frac{K\ln T}{\delta'})}}{c_{\min}\epsilon}).$$

Similarly, the algorithm is $(\epsilon, \delta + \delta')$-differentially private using the Gaussian noise. Setting $K = \mathcal{O}(\sqrt{T})$ in the above inequalities, we obtain the regret bounds as stated.

Appendix H

APPENDIX OF CHAPTER 10

H.1 Missing proofs

H.1.1 Proof of Theorem 10.4.1

At time $t = 1$, we set $\pi_{S,1} = \frac{1}{\binom{K}{m}}$ for each set S where $|S| = m$ and update these weights as follows:

$$\pi_{S,t+1} = \pi_{S,t}(1 - \eta L_{S,t}),$$

where $L_{S,t} = \ell_{S,t} - \sum_{S':|S'|=m} \pi_{S',t}\ell_{S',t}$ and $\ell_{S,t} = \frac{1}{m}\sum_{i \in S} \ell_{i,t}$. If we denote the optimal set as S^*, we can write:

$$1 \geq \pi_{S^*,T+1} = \frac{1}{\binom{K}{m}} \prod_{t=1}^{T} (1 - \eta L_{S^*,t}).$$

Taking the natural logarithm of both sides, we have:

$$0 \geq -\ln \binom{K}{m} + \sum_{t=1}^{T} \ln(1 - \eta L_{S^*,t}).$$

We can use the inequality $\ln(1-x) \geq -x - x^2$ for $x \leq \frac{1}{2}$ (we choose η later such that this inequality holds) to obtain:

$$0 \geq -\ln \binom{K}{m} - \eta \sum_{t=1}^{T} L_{S^*,t} - \eta^2 \sum_{t=1}^{T} L_{S^*,t}^2.$$

Rearranging the terms and dividing both sides by η, we have:

$$-\sum_{t=1}^{T} L_{S^*,t} \leq \frac{\ln \binom{K}{m}}{\eta} + \eta \sum_{t=1}^{T} L_{S^*,t}^2.$$

Using the fact that $R_T = -\sum_{t=1}^{T} L_{S^*,t}$, the inequality $\binom{K}{m} \leq (\frac{Ke}{m})^m$ and $|L_{S,t}| \leq 1\ \forall S, t$, we can write:

$$R_T \leq \frac{m \ln(\frac{Ke}{m})}{\eta} + \eta T.$$

Setting $\eta = \sqrt{\frac{m \ln(\frac{Ke}{m})}{T}}$, we obtain the regret bound $\mathcal{O}(\sqrt{mT \ln(\frac{K}{m})})$. Note that the assumption $\eta L_{S^*,t} \leq \frac{1}{2}$ (used in the proof) holds if $T \geq 4m \ln(\frac{Ke}{m})$. Therefore, we assume T is large enough to satisfy this inequality.

H.1.2 Proof of Theorem 10.5.1

We can show that if $B = \max_z |\nu'(z)|$, the hazard rate of the distribution is bounded above by B as well. To see this, fix $x \geq 0$. Note that $\frac{f(-x)}{1-F(-x)} = \frac{f(x)}{1-F(-x)} \leq \frac{f(x)}{1-F(x)}$ for a symmetric zero-mean distribution and therefore, we only need to bound the hazard rate at $x > 0$ to bound $\text{haz}_{\mathcal{D}}$. We can write:

$$\begin{aligned}
\frac{f_{\mathcal{D}}(x)}{1 - F_{\mathcal{D}}(x)} &= \frac{f_{\mathcal{D}}(x)}{\int_x^\infty f_{\mathcal{D}}(z)dz} \\
&= \frac{\exp(-\nu(x))}{\int_x^\infty \exp(-\nu(z))dz} \\
&= \frac{1}{\int_x^\infty \exp(\nu(x) - \nu(z))dz} \\
&\overset{\text{(a)}}{=} \frac{1}{\int_x^\infty \exp((x - z)\nu'(z_x))dz} \\
&\overset{\text{(b)}}{\leq} \frac{1}{\int_x^\infty \exp(B(x - z))dz} \\
&= \frac{1}{(1/B)} \\
&= B,
\end{aligned}$$

where we have used the mean-value theorem in (a) and z_x is in the line segment between x and z. Also, note that since $x, z > 0$, $z_x > 0$ holds as well and therefore, $\nu'(z_x) > 0$. We have used this fact along with $x - z < 0$ to obtain (b). Therefore, we have $\text{haz}_{\mathcal{D}} \leq B$.

H.1.3 Proof of Theorem 10.5.2

Let's fix round $t \in [T]$ and expert $i \in [K]$. For $j \neq i$, denote $x_{j,0}^{(t)} = \sum_{s=1}^{t-1} \ell_{j,s} + p_{j,t}^2 + \eta\gamma_{j,t+1}$ as the total losses of expert j up to round t plus noise if $r_t = 0$. Similarly, we can define $x_{j,1}^{(t)} = \sum_{s=1}^{t-1} \ell_{j,s} + (1 - p_{j,t})^2 + \eta\gamma_{j,t+1}$. Define $X_0^{(t)}$ ($X_1^{(t)}$) as the m-th smallest value in

$\{x_{j,0}^{(t)}\}_{j\neq i}$ ($\{x_{j,1}^{(t)}\}_{j\neq i}$). Note that $|X_0^{(t)} - X_1^{(t)}| \leq 1$ holds because for each $j \neq i$, we have $|x_{j,0}^{(t)} - x_{j,1}^{(t)}| \leq 1$. Also, let $L = \sum_{s=1}^{t-1} \ell_{i,s}$. If $r_t = 0$, expert i is chosen at round $t+1$ if and only if $L + p_{i,t}^2 + \eta\gamma_{i,t+1} <= X_0^{(t)}$. Similarly, for the case $r_t = 1$, expert i is chosen if and only if $L + (1-p_{i,t})^2 + \eta\gamma_{i,t+1} <= X_1^{(t)}$. Rearranging the terms, we can write:

$$\begin{aligned} \eta\gamma_{i,t+1} + L - X_0^{(t)} &\leq -p_{i,t}^2 \quad \text{if } r_t = 0, \\ \eta\gamma_{i,t+1} + L - X_1^{(t)} &\leq -(1-p_{i,t})^2 \quad \text{if } r_t = 1. \end{aligned}$$

Given that $f(\gamma_{i,t+1}) \propto \exp(-\nu(\gamma_{i,t+1}))$, if we define $Y_0 = \eta\gamma_{i,t+1} + L - X_0^{(t)}$ and $Y_1 = \eta\gamma_{i,t+1} + L - X_1^{(t)}$, we have:

$$\begin{aligned} f_0(Y_0) &\propto \exp(-\nu(\frac{Y_0 - (L - X_0^{(t)})}{\eta})), \\ f_1(Y_1) &\propto \exp(-\nu(\frac{Y_1 - (L - X_1^{(t)})}{\eta})). \end{aligned}$$

Therefore, if F_0 and F_1 denote the cdf, we can write the probability of expert i being chosen at round $t+1$ as $F_0(-p_{i,t}^2)$ and $F_1(-(1-p_{i,t})^2)$ for the cases $r_t = 0$ and $r_t = 1$ respectively. Putting the above results together, we can write the expected utility of expert i (according to her belief $b_{i,t}$) at round t as follows:

$$\mathbb{E}_{r_t \sim \text{Bernoulli}(b_{i,t})}[U_{i,t}] = (1-b_{i,t})F_0(-p_{i,t}^2) + b_{i,t}F_1(-(1-p_{i,t})^2).$$

Taking the derivative and setting it to zero, we have:

$$\frac{d}{dp_{i,t}}\mathbb{E}_{r_t \sim \text{Bernoulli}(b_{i,t})}[U_{i,t}] = -2p_{i,t}(1-b_{i,t})f_0(-p_{i,t}^2) + 2(1-p_{i,t})b_{i,t}f_1(-(1-p_{i,t})^2) = 0$$

$$2f_1(-(1-p_{i,t})^2)\big(-p_{i,t}(1-b_{i,t})\frac{f_0(-p_{i,t}^2)}{f_1(-(1-p_{i,t})^2)} + (1-p_{i,t})b_{i,t}\big) = 0$$

$$-p_{i,t}(1-b_{i,t})\frac{\exp(-\nu(\frac{-p_{i,t}^2-(L-X_0^{(t)})}{\eta}))}{\exp(-\nu(\frac{-(1-p_{i,t})^2-(L-X_1^{(t)})}{\eta}))} + (1-p_{i,t})b_{i,t} = 0$$

$$-p_{i,t}(1-b_{i,t})\underbrace{\exp\big(\nu(\frac{-(1-p_{i,t})^2 - (L - X_1^{(t)})}{\eta}) - \nu(\frac{-p_{i,t}^2 - (L - X_0^{(t)})}{\eta})\big)}_{=A} + (1-p_{i,t})b_{i,t} = 0$$

Ideally, we want A to be as close to 1 as possible because if $A = 1$, $p^*_{i,t} = b_{i,t}$ and the algorithm would be incentive-compatible. In general, we have:

$$p^*_{i,t} = \frac{b_{i,t}}{b_{i,t} + (1 - b_{i,t})A}.$$

Note that this is not a closed-form solution for $p^*_{i,t}$ because A is also a function of $p_{i,t}$. We can observe that for $p_{i,t} < p^*_{i,t}$, the derivative of the utility function is positive, and for $p_{i,t} > p^*_{i,t}$, the derivative is negative.

We can bound A as follows: Let $g(u) = f(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta})$ where $a = L - X^{(t)}_1 + 1 - 2p_{i,t}$, $a' = L - X^{(t)}_0$ and $f(z) = \exp(-\nu(z))$. Taking derivative of g with respect to u, we obtain:

$$g'(u) = \frac{(a'-a)\nu'(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta})}{\eta} f(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta}) = \frac{(a'-a)\nu'(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta})}{\eta} g(u).$$

Therefore, we can write:

$$\begin{aligned}
\ln f(\frac{-p^2_{i,t} - a'}{\eta}) - \ln f(\frac{-p^2_{i,t} - a}{\eta}) &= \ln g(1) - \ln g(0) \\
&= \int_0^1 \frac{g'(u)}{g(u)} du \\
&= \int_0^1 \frac{(a'-a)\nu'(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta})}{\eta} du.
\end{aligned}$$

We have $a' - a = X^{(t)}_1 - X^{(t)}_0 + 2p_{i,t} - 1$. Therefore, $-2 \le a' - a \le 2$ holds. Moreover, we have $-B \le \nu'(\frac{-p^2_{i,t} - a - u(a'-a)}{\eta}) \le B$ due to Condition 10.5.1. Putting the above results together, the following holds:

$$\frac{-2B}{\eta} \le \ln A = \ln f(\frac{-p^2_{i,t} - a'}{\eta}) - \ln f(\frac{-p^2_{i,t} - a}{\eta}) \le \frac{2B}{\eta}.$$

Therefore, $A \in [\exp(\frac{-2B}{\eta}), \exp(\frac{2B}{\eta})]$. Let $h(p_{i,t}) = p_{i,t} - p^*_{i,t} = p_{i,t} - \frac{b_{i,t}}{b_{i,t} + (1-b_{i,t})A}$. Since $p^*_{i,t} \in [0, 1]$, we have $h(0) < 0$ and $h(1) > 0$. Taking the derivative of h with respect to $p_{i,t}$, we have:

$$\frac{dh}{dp_{i,t}} = 1 + \frac{b_{i,t}(1 - b_{i,t})A\big(\frac{2(1-p_{i,t})}{\eta}\nu'(\frac{-(1-p_{i,t})^2 - (L - X^{(t)}_1)}{\eta}) + \frac{2p_{i,t}}{\eta}\nu'(\frac{-p^2_{i,t} - (L - X^{(t)}_0)}{\eta})\big)}{(b_{i,t} + (1 - b_{i,t})A)^2}.$$

Using Condition 10.5.1, we have $1 - \frac{2Bb_{i,t}(1-b_{i,t})A}{\eta(b_{i,t}+(1-b_{i,t})A)^2} \leq \frac{dh}{dp_{i,t}}$. Therefore, we can write:

$$\begin{aligned}
\frac{dh}{dp_{i,t}} &\geq 1 - \frac{2Bb_{i,t}(1-b_{i,t})A}{\eta(b_{i,t}+(1-b_{i,t})A)^2} \\
&= 1 - \frac{2Bb_{i,t}(1-b_{i,t})A}{\eta(b_{i,t}^2+(1-b_{i,t})^2A^2+2b_{i,t}(1-b_{i,t})A)} \\
&\geq 1 - \frac{2Bb_{i,t}(1-b_{i,t})A}{2\eta b_{i,t}(1-b_{i,t})A} \\
&= 1 - \frac{B}{\eta} \\
&> 0.
\end{aligned}$$

We choose η later such that $\eta > B$ for the last inequality to hold. So, h is strictly increasing, there is exactly one solution $p_{i,t}^*$, and the derivative is positive below it and negative above it.

If we replace A with something larger in $p_{i,t}^*$, the value decreases and vice versa. Therefore, we can write:

$$\begin{aligned}
p_{i,t}^* &\leq \frac{b_{i,t}}{b_{i,t}+(1-b_{i,t})\exp(\frac{-2B}{\eta})} \\
&= \frac{b_{i,t}}{\exp(\frac{-2B}{\eta})+b_{i,t}(1-\exp(\frac{-2B}{\eta}))} \\
&\leq \frac{b_{i,t}}{\exp(\frac{-2B}{\eta})} \\
&\leq \frac{b_{i,t}}{1-\frac{2B}{\eta}} \\
&= \frac{\eta b_{i,t}}{\eta-2B} \\
&= b_{i,t} + \frac{2Bb_{i,t}}{\eta-2B} \\
&\leq b_{i,t} + \frac{2B}{\eta-2B}.
\end{aligned}$$

Similarly, we can lower bound $p_{i,t}^*$ as follows:

$$p_{i,t}^* \geq \frac{b_{i,t}}{b_{i,t}+(1-b_{i,t})\exp(\frac{2B}{\eta})}$$

$$
\begin{aligned}
&= \frac{b_{i,t}}{\exp(\frac{2B}{\eta}) - b_{i,t}(\exp(\frac{2B}{\eta}) - 1)} \\
&\geq \frac{b_{i,t}}{\exp(\frac{2B}{\eta})} \\
&= b_{i,t}\exp(\frac{-2B}{\eta}) \\
&\geq b_{i,t}(1 - \frac{2B}{\eta}) \\
&\geq b_{i,t} - \frac{2B}{\eta}.
\end{aligned}
$$

Putting the above results together, we conclude that $|p^*_{i,t} - b_{i,t}| \leq \frac{2B}{\eta - 2B}$ holds.

H.1.4 Proof of Theorem 10.5.3

Let $\eta > 0$ and $\mathcal{X}$ be the set of $\binom{K}{m}$ feasible sets of size m for this problem. Denote $L_t \in \mathbb{R}^K$ as the partial sum of losses for all experts before round t, i.e., $[L_t]_i = \sum_{s=1}^{t-1} \ell_{i,s}$. The update rule of FTPL at round $t \in [T]$ is $\pi_t = \arg\min_{x \in \mathcal{X}} \langle x, L_t + \eta\gamma_t \rangle$. First, we analyze the expected regret of the algorithm below.

Define $\phi_t(\theta) = \mathbb{E}_{\gamma_t \sim \mathcal{D}^K}[\min_{x \in \mathcal{X}} \langle x, \theta + \eta\gamma_t \rangle]$. Then, we have $\nabla\phi_t(L_t) = \mathbb{E}_{\gamma_t \sim \mathcal{D}^K}[\arg\min_{x \in \mathcal{X}} \langle x, L_t + \eta\gamma_t \rangle] = \mathbb{E}_{\gamma_t \sim \mathcal{D}^K}[\pi_t]$ and $\langle \nabla\phi_t(L_t), \ell_t \rangle = \mathbb{E}_{\gamma_t \sim \mathcal{D}^K}[\langle \pi_t, \ell_t \rangle]$. Using the Taylor's expansion of ϕ_t, we can write:

$$\phi_t(L_{t+1}) = \phi_t(L_t) + \langle \nabla\phi_t(L_t), \ell_t \rangle + \frac{1}{2}\langle \ell_t, \nabla^2\phi_t(L_t^{'})\ell_t \rangle = \phi_t(L_t) + \mathbb{E}_{\gamma_t \sim \mathcal{D}^K}[\langle \pi_t, \ell_t \rangle] + \frac{1}{2}\langle \ell_t, \nabla^2\phi_t(L_t^{'})\ell_t \rangle,$$

where $L_t^{'}$ is in the line segment between L_t and $L_{t+1} = L_t + \ell_t$. By definition, ϕ_t is the minimum of linear functions and therefore, it is concave. Thus, $\nabla^2\phi_t(L_t^{'})$ is negative semidefinite. Note that since γ_t is simply K i.i.d. samples of the noise distribution for all $t \in [T]$, we have $\phi_t(\theta) = \phi(\theta) = \mathbb{E}_{\gamma \sim \mathcal{D}^K}[\min_{x \in \mathcal{X}} \langle x, \theta + \eta\gamma \rangle]$. Taking the sum over $t \in [T]$ and rearranging the terms, we obtain:

$$\mathbb{E}[\sum_{t=1}^{T} \langle \pi_t, \ell_t \rangle] = \phi(L_{T+1}) - \phi(\underbrace{L_1}_{=0}) - \frac{1}{2}\sum_{t=1}^{T} \langle \ell_t, \nabla^2\phi_t(L_t^{'})\ell_t \rangle.$$

Using Jensen's inequality, we can write:

$$\phi(L_{T+1}) = \mathbb{E}_{\gamma \sim \mathcal{D}^K}[\min_{x \in \mathcal{X}} \langle x, L_{T+1} + \eta\gamma \rangle] \leq \min_{x \in \mathcal{X}} \mathbb{E}_{\gamma \sim \mathcal{D}^K}[\langle x, L_{T+1} + \eta\gamma \rangle] = \min_{x \in \mathcal{X}} \langle x, L_{T+1} \rangle.$$

Therefore, we can rearrange the terms and bound the expected regret as follows:

$$\mathbb{E}[R_T] = \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\sum_{t=1}^{T}\langle\pi_t, \ell_t\rangle] - \frac{1}{m}\min_{x\in\mathcal{X}}\langle x, L_{T+1}\rangle \leq -\frac{1}{m}\phi(0) - \frac{1}{2m}\sum_{t=1}^{T}\langle\ell_t, \nabla^2\phi_t(L_t^{'})\ell_t\rangle.$$

To bound the first term on the right-hand side, we can use the fact that $\mathcal{D}$ is symmetric to write:

$$-\phi(0) = -\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\min_{x\in\mathcal{X}}\langle x, \eta\gamma\rangle] = \eta\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\max_{x\in\mathcal{X}}\langle x, -\gamma\rangle] = \eta\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\max_{x\in\mathcal{X}}\langle x, \gamma\rangle].$$

Now, we move on to bound the second term in the regret bound. Given that the losses of the experts at each round $t \in [T]$ are bounded between 0 and 1, we have $\|\ell_t\|_\infty \leq 1$. Therefore, we have:

$$-\langle\ell_t, \nabla^2\phi_t(L_t^{'})\ell_t\rangle \leq \sum_{i,j\in[K]}|\nabla^2_{i,j}\phi_t(L_t^{'})|.$$

By definition, if we denote $\hat{\pi}(\theta) = \arg\min_{x\in\mathcal{X}}\langle x, \theta\rangle$, we have:

$$\nabla^2_{i,j}\phi(L_t^{'}) = \frac{1}{\eta}\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\hat{\pi}(L_t^{'} + \eta\gamma)_i\frac{d\nu(\gamma_j)}{d\gamma_j}].$$

Given the concavity of ϕ, diagonal entries of the Hessian $\nabla^2\phi$ are non-positive. We can also use $1^T\hat{\pi}(\theta) = m$ to show that the off-diagonal entries are non-negative and each row or column of the Hessian sums up to 0. Therefore, we have $\sum_{i,j\in[K]}|\nabla^2_{i,j}\phi(L_t^{'})| = -2\sum_{i=1}^{K}\nabla^2_{i,i}\phi(L_t^{'})$. Putting the above results together, we can bound the regret as follows:

$$\mathbb{E}[R_T] \leq \frac{\eta}{m}\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\max_{x\in\mathcal{X}}\langle x, \gamma\rangle] + \frac{1}{\eta m}\sum_{t=1}^{T}\sum_{i=1}^{K}\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\hat{\pi}(L_t^{'} - \eta\gamma)_i\frac{d\nu(\gamma_i)}{d\gamma_i}].$$

We can bound the first term as follows:

$$\begin{aligned}
\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\max_{x\in\mathcal{X}}\langle x, \gamma\rangle] &\leq \inf_{s:s>0}\frac{1}{s}\ln\big(\sum_{x\in\mathcal{X}}\mathbb{E}_{\gamma\sim\mathcal{D}^K}[\exp(s\langle x, \gamma\rangle)]\big)\\
&= \inf_{s:s>0}\frac{1}{s}\ln\big(\sum_{x\in\mathcal{X}}\prod_{i=1}^{K}\mathbb{E}_{\gamma_i\sim\mathcal{D}}[\exp(sx_i\gamma_i)]\big)\\
&= \inf_{s:s>0}\frac{1}{s}\ln\big(|\mathcal{X}|(\mathbb{E}_{\gamma_1\sim\mathcal{D}}[\exp(s\gamma_1)])^m\big)\\
&= \inf_{s:s>0}\frac{1}{s}\ln|\mathcal{X}| + \frac{m}{s}\ln\mathbb{E}_{\gamma_1\sim\mathcal{D}}[\exp(s\gamma_1)]
\end{aligned}$$

$$\begin{aligned}
&\leq \inf_{s:s>0} \frac{m}{s} \ln(\frac{Ke}{m}) + \frac{m}{s} \ln \mathbb{E}_{\gamma_1 \sim \mathcal{D}}[\exp(s\gamma_1)] \\
&\leq \mathcal{O}(m \ln(\frac{K}{m})) + \mathcal{O}(m) \\
&= \mathcal{O}(m \ln(\frac{K}{m})).
\end{aligned}$$

To bound the second term in the regret bound, we can use Condition 10.5.1 to write:

$$\sum_{i=1}^{K} \mathbb{E}_{\gamma\sim\mathcal{D}}[\hat{\pi}(L_t^{'} - \eta\gamma)_i \frac{d\nu(\gamma_i)}{d\gamma_i}] \leq B \sum_{i=1}^{K} \mathbb{E}_{\gamma\sim\mathcal{D}}[\hat{\pi}(L_t^{'} - \eta\gamma)_i] = B\mathbb{E}_{\gamma\sim\mathcal{D}}[\sum_{i=1}^{K} \hat{\pi}(L_t^{'} - \eta\gamma)_i] = Bm.$$

Putting the above results together, we have:

$$\mathbb{E}[R_T] \leq \mathcal{O}(\eta \ln(\frac{K}{m})) + \frac{BT}{\eta}.$$

Therefore, if we set $\eta = \sqrt{\frac{BT}{\ln(\frac{K}{m})}}$, the regret bound is $\mathcal{O}(\sqrt{BT\ln(\frac{K}{m})})$. In the proof of Theorem 10.5.2, we assumed that η is chosen such that $\eta > B$. So, we assume $T > B\ln(\frac{K}{m})$.

H.1.5 Proof of Theorem 10.6.1

For $i \in [m]$ and $t \in [T]$, define:

$$\phi_{i,t}(S) = (1 - \frac{1}{m})^{m-i} g_t(S) + h_t(S).$$

For all $i \in [m]$, we can write

$$\begin{aligned}
\phi_{i,t}(S_{i,t}) - \phi_{i-1,t}(S_{i-1,t}) &= (1-\frac{1}{m})^{m-i} g_t(S_{i,t}) + h_t(S_{i,t}) - (1-\frac{1}{m})^{m-(i-1)} g_t(S_{i-1,t}) - h_t(S_{i-1,t}) \\
&= (1-\frac{1}{m})^{m-i}(g_t(S_{i,t}) - g_t(S_{i-1,t})) + h_t(v_{i,t}) + \frac{1}{m}(1-\frac{1}{m})^{m-i} g_t(S_{i-1,t}).
\end{aligned}$$

Taking the sum over $t \in [T]$, we obtain:

$$\begin{aligned}
\sum_{t=1}^{T}(\phi_{i,t}(S_{i,t}) - \phi_{i-1,t}(S_{i-1,t})) &= (1-\frac{1}{m})^{m-i} \sum_{t=1}^{T}(g_t(S_{i,t}) - g_t(S_{i-1,t})) + \sum_{t=1}^{T} h_t(v_{i,t}) \\
&\quad + \frac{1}{m}(1-\frac{1}{m})^{m-i} \sum_{t=1}^{T} g_t(S_{i-1,t}) \\
&= (1-\frac{1}{m})^{m-i} \sum_{t=1}^{T} g_t(v_{i,t}|S_{i-1,t}) + \sum_{t=1}^{T} h_t(v_{i,t})
\end{aligned}$$

$$+\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}g_t(S_{i-1,t}).$$

If the regret of $\mathcal{A}_i$ is bounded above by $R_T^{(i)}$ and the optimal benchmark solution is $\mathrm{OPT}=\{v_1^*,\dots,v_m^*\}$, for all $j\in[m]$, we can write:

$$\sum_{t=1}^{T}\big((1-\frac{1}{m})^{m-i}g_t(v_j^*|S_{i-1,t})+h_t(v_j^*)\big)-\sum_{t=1}^{T}\big((1-\frac{1}{m})^{m-i}g_t(v_{i,t}|S_{i-1,t})+h_t(v_{i,t})\big)\le R_T^{(i)}.$$

Putting the above inequalities together, we have:

$$\begin{aligned}\sum_{t=1}^{T}(\phi_{i,t}(S_{i,t})-\phi_{i-1,t}(S_{i-1,t}))\ge{}&\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}\sum_{j=1}^{m}g_t(v_j^*|S_{i-1,t})+\frac{1}{m}\sum_{t=1}^{T}\sum_{j=1}^{m}h_t(v_j^*)\\&+\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}g_t(S_{i-1,t})-R_T^{(i)}.\end{aligned}$$

We can use submodularity and monotonicity of g_t to write:

$$g_t(\mathrm{OPT})-g_t(S_{i-1,t})\le g_t(\mathrm{OPT}\cup S_{i-1,t})-g_t(S_{i-1,t})=\sum_{j=1}^{m}g_t(v_j^*|S_{i-1,t}\cup\{v_1^*,\dots,v_{j-1}^*\})\le\sum_{j=1}^{m}g_t(v_j^*|S_{i-1,t}).$$

We can combine the last two inequalities to write:

$$\begin{aligned}\sum_{t=1}^{T}(\phi_{i,t}(S_{i,t})-\phi_{i-1,t}(S_{i-1,t}))\ge{}&\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}(g_t(\mathrm{OPT})-g_t(S_{i-1,t}))+\frac{1}{m}\sum_{t=1}^{T}\sum_{j=1}^{m}h_t(v_j^*)\\&+\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}g_t(S_{i-1,t})-R_T^{(i)}\\={}&\frac{1}{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}g_t(\mathrm{OPT})+\frac{1}{m}\sum_{t=1}^{T}\sum_{j=1}^{m}h_t(v_j^*)-R_T^{(i)}.\end{aligned}$$

Taking the sum over $i\in[m]$, we have:

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{m}(\phi_{i,t}(S_{i,t})-\phi_{i-1,t}(S_{i-1,t}))\ge{}&\frac{1}{m}\sum_{i=1}^{m}(1-\frac{1}{m})^{m-i}\sum_{t=1}^{T}g_t(\mathrm{OPT})+\sum_{t=1}^{T}\sum_{j=1}^{m}h_t(v_j^*)-\sum_{i=1}^{m}R_T^{(i)}\\={}&(1-(1-\frac{1}{m})^m)\sum_{t=1}^{T}g_t(\mathrm{OPT})+\sum_{t=1}^{T}h_t(\mathrm{OPT})-\sum_{i=1}^{m}R_T^{(i)}\\\ge{}&(1-\frac{1}{e})\sum_{t=1}^{T}g_t(\mathrm{OPT})+\sum_{t=1}^{T}h_t(\mathrm{OPT})-\sum_{i=1}^{m}R_T^{(i)}.\end{aligned}$$

On the other hand, we have:

$$\sum_{t=1}^{T}\sum_{i=1}^{m}(\phi_{i,t}(S_{i,t})-\phi_{i-1,t}(S_{i-1,t})) = \sum_{t=1}^{T}(\phi_{m,t}(S_{m,t})-\phi_{0,t}(S_{0,t})) = \sum_{t=1}^{T}(g_t(S_t)+h_t(S_t)) = \sum_{t=1}^{T} f_t(S_t).$$

Combining the last two inequalities, we obtain:

$$\begin{aligned}\sum_{t=1}^{T} f_t(S_t) &\geq (1-\frac{1}{e})\sum_{t=1}^{T} g_t(\text{OPT}) + \sum_{t=1}^{T} h_t(\text{OPT}) - \sum_{i=1}^{m} R_T^{(i)} \\ &= (1-\frac{1}{e})\sum_{t=1}^{T} f_t(\text{OPT}) - (1-\frac{1}{e})\sum_{t=1}^{T} h_t(\text{OPT}) + \sum_{t=1}^{T} h_t(\text{OPT}) - \sum_{i=1}^{m} R_T^{(i)} \\ &= (1-\frac{1}{e})\sum_{t=1}^{T} f_t(\text{OPT}) + \frac{1}{e}\sum_{t=1}^{T} h_t(\text{OPT}) - \sum_{i=1}^{m} R_T^{(i)} \\ &\geq (1-\frac{1}{e})\sum_{t=1}^{T} f_t(\text{OPT}) + \frac{1-c_f}{e}\sum_{t=1}^{T} f_t(\text{OPT}) - \sum_{i=1}^{m} R_T^{(i)} \\ &= (1-\frac{c_f}{e})\sum_{t=1}^{T} f_t(\text{OPT}) - \sum_{i=1}^{m} R_T^{(i)},\end{aligned}$$

where $f = \sum_{t=1}^{T} f_t$. Rearranging the terms, we obtain the $(1-\frac{c_f}{e})$-regret bound of the online distorted greedy algorithm as follows:

$$(1-\frac{c_f}{e})\sum_{t=1}^{T} f_t(\text{OPT}) - \sum_{t=1}^{T} f_t(S_t) \leq \sum_{i=1}^{m} R_T^{(i)}.$$

To see why the online greedy algorithm is incentive-compatible, note that at round $t \in [T]$, the loss of expert $j \in [K]$ for the instance $\mathcal{A}_i;\ i \in [K]$ is:

$$-f_t(j|S_{i-1,t}) = \prod_{k\in S_{i-1,t}} \ell_{k,t}(\ell_{j,t}-1).$$

If we use $\pi_{j,t+1}^{(i)}$ to denote the probability of expert j being chosen at round $t+1$ by $\mathcal{A}_i$, we can write:

$$\pi_{j,t+1}^{(i)} = \pi_{j,t}^{(i)}\big(1-\eta\prod_{k\in S_{i-1,t}}\ell_{k,t}(\ell_{j,t}-\sum_{s=1}^{K}\pi_{s,t}^{(i)}\ell_{s,t})\big).$$

$\pi_{j,t+1}^{(i)}$ is linear in $\ell_{j,t}$ and expert j does not have control over the term $\prod_{k\in S_{i-1,t}}\ell_{k,t}$ (because $S_{i-1,t}$ is unknown to the expert). Therefore, to maximize $\pi_{j,t+1}^{(i)}$, expert j can only aim to minimize $\mathbb{E}_{r_t \sim \text{Bern}(b_{j,t})}[\ell_{j,t} - \sum_{s=1}^{K} \pi_{s,t}^{(i)} \ell_{s,t}]$. Given that the quadratic loss function is proper, we can conclude that $\mathcal{A}_i$ is incentive-compatible. Moreover, since the online greedy algorithm simply outputs the predictions of $\{\mathcal{A}_i\}_{i=1}^{K}$, this algorithm is incentive-compatible as well. For the case with $c_f = 0$, the loss of expert $j \in [K]$ for each instance $\mathcal{A}_i;\ i \in [K]$ in both online greedy algorithm and online distorted greedy algorithm is simply $\frac{1}{m}(\ell_{j,t} - 1)$ and the same argument holds.

H.1.6 Proof of Theorem 10.6.2

First, assume that the losses are non-negative. Using the update rule of the algorithm, we can write:

$$\begin{aligned}
1 &\geq \pi_{i^*,T+1} = \pi_{i^*,1} \prod_{t=1}^{T} (1 - \eta L_{i^*,t}) \\
1 &\geq \frac{1}{K} \prod_{t:L_{i^*,t} \geq 0} (1 - \eta L_{i^*,t}) \prod_{t:L_{i^*,t} < 0} (1 - \eta L_{i^*,t}).
\end{aligned}$$

We can use the inequalities $1 - \eta x \geq (1-\eta)^x$ for $x \in [0,1]$ and $1 - \eta x \geq (1+\eta)^{-x}$ for $x \in [-1, 0]$ to write:

$$\begin{aligned}
1 &\geq \frac{1}{K} \prod_{t:L_{i^*,t} \geq 0} (1-\eta)^{L_{i^*,t}} \cdot \prod_{t:L_{i^*,t} < 0} (1+\eta)^{-L_{i^*,t}} \\
0 &\geq -\ln K + \sum_{t:L_{i^*,t} \geq 0} L_{i^*,t} \ln(1-\eta) - \sum_{t:L_{i^*,t} < 0} L_{i^*,t} \ln(1+\eta).
\end{aligned}$$

Given the inequalities $\ln(1-x) \geq -x - x^2$ and $\ln(1+x) \geq x - x^2$ for $x \leq \frac{1}{2}$, we have:

$$\begin{aligned}
0 &\geq -\ln K + (-\eta - \eta^2) \sum_{t:L_{i^*,t} \geq 0} L_{i^*,t} - (\eta - \eta^2) \sum_{t:L_{i^*,t} < 0} L_{i^*,t} \\
0 &\geq -\ln K - \eta \sum_{t=1}^{T} L_{i^*,t} - \eta^2 \sum_{t=1}^{T} |L_{i^*,t}| \\
R_T = -\sum_{t=1}^{T} L_{i^*,t} &\leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} |L_{i^*,t}|.
\end{aligned}$$

Given that $L_{i^*,t} = \ell_{i^*,t} - \pi_t^T \ell_t$, we can write:

$$R_T \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^{T} |\ell_{i^*,t} - \pi_t^T \ell_t| \leq \frac{\ln K}{\eta} + \eta (\sum_{t=1}^{T} |\ell_{i^*,t}| + \sum_{t=1}^{T} |\pi_t^T \ell_t|) = \frac{\ln K}{\eta} + \eta(|L_T^*| + |L_T|),$$

where $|L_T^*|$ and $|L_T|$ are the cumulative absolute loss of the benchmark and the algorithm respectively. Therefore, if we set $\eta = \min\{1, \sqrt{\frac{\ln K}{|L_T| + |L_T^*|}}\}$, we obtain an $\mathcal{O}(\sqrt{\max\{|L_T|, |L_T^*|\} \ln K} + \ln K)$ regret bound. Given that $R_T = L_T - L_T^*$, if $|L_T| = L_T < L_T^* = |L_T^*|$, the regret is negative and if $|L_T| = L_T \geq L_T^* = |L_T^*|$, the regret bound is $\mathcal{O}(\sqrt{|L_T| \ln K} + \ln K)$. Therefore, the regret bound $\mathcal{O}(\sqrt{|L_T| \ln K} + \ln K)$ always holds.

H.1.7 Proof of Theorem 10.7.1

Given that ℓ is a (strictly) proper loss function, for $p \in [0,1]$ and $r \in \{0,1\}$, we have:

$$\ell(p, r) = -G(p) + (p - r)G'(p).$$

where G is a (strictly) convex function. To guarantee that experts converge to the Nash equilibrium, we have to ensure that the loss of each expert is convex. Taking the derivative of the expected loss with respect to $p_{i,t}$, we have:

$$\begin{aligned}
\frac{d}{dp_{i,t}} \mathbb{E}_{r_t \sim \mathrm{Bern}\left(f_{i,t}(\pi_t^T p_t)\right)} \ell(p_{i,t}, r_t) &= \pi_{i,t} f'_{i,t}(\pi_t^T p_t) \ell(p_{i,t}, 1) + f_{i,t}(\pi_t^T p_t) \ell'(p_{i,t}, 1) \\
&\quad - \pi_{i,t} f'_{i,t}(\pi_t^T p_t) \ell(p_{i,t}, 0) + (1 - f_{i,t}(\pi_t^T p_t)) \ell'(p_{i,t}, 0) \\
&= f_{i,t}(\pi_t^T p_t) \big(\ell'(p_{i,t}, 1) - \ell'(p_{i,t}, 0)\big) + \ell'(p_{i,t}, 0) \\
&\quad + \pi_{i,t} f'_{i,t}(\pi_t^T p_t) \big(\ell(p_{i,t}, 1) - \ell(p_{i,t}, 0)\big) \\
&= -\pi_{i,t} f'_{i,t}(\pi_t^T p_t) G'(p_{i,t}) - f_{i,t}(\pi_t^T p_t) G''(p_{i,t}) + p_{i,t} G''(p_{i,t}).
\end{aligned}$$

Taking the second derivative of the expected loss, we can write:

$$\begin{aligned}
\frac{d^2}{dp_{i,t}^2} \mathbb{E}_{r_t \sim \mathrm{Bern}(f_{i,t}(\pi_t^T p_t))} \ell(p_{i,t}, r_t) &= -\pi_{i,t}^2 f''_{i,t}(\pi_t^T p_t) G'(p_{i,t}) - \pi_{i,t} f'_{i,t}(\pi_t^T p_t) G''(p_{i,t}) + p_{i,t} G'''(p_{i,t}) \\
&\quad + G''(p_{i,t}) - \pi_{i,t} f'_{i,t}(\pi_t^T p_t) G''(p_{i,t}) - f_{i,t}(\pi_t^T p_t) G'''(p_{i,t}) \\
&= \big(p_{i,t} - f_{i,t}(\pi_t^T p_t)\big) G'''(p_{i,t}) + \big(1 - 2\pi_{i,t} f'_{i,t}(\pi_t^T p_t)\big) G''(p_{i,t})
\end{aligned}$$

$$- \pi_{i,t}^2 f''_{i,t}(\pi_t^T p_t) G'(p_{i,t}).$$

Given that G is convex, we have $G''(p) \geq 0$. Therefore, if $f_{i,t}$ is α-Lipschitz where $\alpha < 0.5$, we can argue that the second term in the second derivative $\big(1 - 2\pi_{i,t} f'_{i,t}(\pi_t^T p_t)\big) G''(p_{i,t})$ is non-negative. Assuming that the third derivative of G (G''') and the second derivative of $f_{i,t}$ ($f''_{i,t}$) are zero (i.e., G is a polynomial of degree at most 2 and f is an affine function), we can make sure that the other two terms are zero. For instance, for quadratic loss functions, we have $G(p) = p^2 - p$. If we consider the drift model with $\alpha < 0.5$, $\mathbb{E}_{r_t \sim \text{Bern}\left(f_{i,t}(\pi_t^T p_t)\right)} \ell(p_{i,t}, r_t)$ is a convex function of $p_{i,t}$.

H.1.8 Proof of Theorem 10.7.2

Proof For a fixed $i \in [K]$ and $t \in [T]$, we have:

$$\begin{aligned}
\frac{d}{dp_{i,t}} \mathbb{E}_{r_t \sim \text{Bern}\left(f_{i,t}(\pi_t^T p_t)\right)} \ell(p_{i,t}, r_t) &= -\alpha \pi_{i,t}(2p_{i,t} - 1) + 2(p_{i,t} - \alpha \pi_t^T p_t - (1-\alpha) b_{i,t}) = 0 \\
\big(2(1 - \alpha\pi_{i,t}) - 2\alpha\pi_{i,t}\big) p_{i,t} &= 2(1-\alpha) b_{i,t} - \alpha\pi_{i,t} + 2\alpha \sum_{j \neq i} \pi_{j,t} p_{j,t} \\
p_{i,t} &= \frac{1-\alpha}{1 - 2\alpha\pi_{i,t}} b_{i,t} + \frac{\alpha}{1 - 2\alpha\pi_{i,t}} \Big(\sum_{j \neq i} \pi_{j,t} p_{j,t} - 0.5\pi_{i,t}\Big).
\end{aligned}$$

Therefore, $p_{i,t} = \min\{0, \frac{1-\alpha}{1-2\alpha\pi_{i,t}} b_{i,t} + \frac{\alpha}{1-2\alpha\pi_{i,t}} (\sum_{j \neq i} \pi_{j,t} p_{j,t} - 0.5\pi_{i,t})\}$. Using the optimality condition, we can write:

$$\begin{aligned}
&\big(-\alpha\pi_{i,t}(2p_{i,t} - 1) + 2(p_{i,t} - f_{i,t}(\pi_t^T p_t))\big)(f_{i,t}(\pi_t^T p_t) - p_{i,t}) \geq 0 \\
&2(f_{i,t}(\pi_t^T p_t) - p_{i,t})^2 \leq -\alpha\pi_{i,t}(2p_{i,t} - 1)(f_{i,t}(\pi_t^T p_t) - p_{i,t}) \leq \alpha\pi_{i,t} \underbrace{|2p_{i,t} - 1|}_{\leq 1} |f_{i,t}(\pi_t^T p_t) - p_{i,t}|.
\end{aligned}$$

Thus, $|f_{i,t}(\pi_t^T p_t) - p_{i,t}| \leq \frac{\alpha\pi_{i,t}}{2}$ holds.

On the other hand, we have:

$$\begin{aligned}
\frac{d}{dp_{i,t}} \mathbb{E}_{r_t \sim \text{Bern}\left(f_{i,t}(p_{i,t})\right)} \ell(p_{i,t}, r_t) &= 2(p_{i,t} - \alpha p_{i,t} - (1-\alpha) b_{i,t}) - \alpha(2p_{i,t} - 1) = 0 \\
\big(2(1-\alpha) - 2\alpha\big) p_{i,t} &= 2(1-\alpha) b_{i,t} - \alpha \\
p_{i,t} &= \frac{1-\alpha}{1-2\alpha} b_{i,t} - \frac{0.5\alpha}{1-2\alpha}.
\end{aligned}$$

So, $p^*_{i,t} = \min\{0, \frac{1-\alpha}{1-2\alpha} b_{i,t} - \frac{0.5\alpha}{1-2\alpha}\}$. We can use the optimality conditions here as well to write:

$$\begin{aligned}
&\big(2(p^*_{i,t} - f_{i,t}(p^*_{i,t}))) - \alpha(2p^*_{i,t} - 1)\big)(f_{i,t}(p^*_{i,t}) - p^*_{i,t}) \geq 0 \\
&2(f_{i,t}(p^*_{i,t}) - p^*_{i,t})^2 \leq -\alpha(2p^*_{i,t} - 1)(f_{i,t}(p^*_{i,t}) - p^*_{i,t}) \leq \alpha \underbrace{|2p^*_{i,t} - 1|}_{\leq 1} |f_{i,t}(p^*_{i,t}) - p^*_{i,t}|.
\end{aligned}$$

Thus, $|f_{i,t}(p^*_{i,t}) - p^*_{i,t}| \leq \frac{\alpha}{2}$ holds. Putting the above results together, we can bound $|p_{i,t} - f_{i,t}(p^*_{i,t})|$ as follows:

$$\begin{aligned}
|p_{i,t} - f_{i,t}(p^*_{i,t})| &= |p_{i,t} - p^*_{i,t} + p^*_{i,t} - f_{i,t}(p^*_{i,t})| \\
&\leq |p_{i,t} - p^*_{i,t}| + |f_{i,t}(p^*_{i,t}) - p^*_{i,t}| \\
&\leq |\frac{1-\alpha}{1-2\alpha\pi_{i,t}} b_{i,t} + \frac{\alpha}{1-2\alpha\pi_{i,t}} (\sum_{j\neq i} \pi_{j,t} p_{j,t} - 0.5\pi_{i,t}) - \frac{1-\alpha}{1-2\alpha} b_{i,t} + \frac{0.5\alpha}{1-2\alpha}| + \frac{\alpha}{2} \\
&= |\frac{1-\alpha}{(1-2\alpha\pi_{i,t})(1-2\alpha)} \big(1 - 2\alpha - 1 + 2\alpha\pi_{i,t}\big) b_{i,t} \\
&+ \alpha\big(\frac{\sum_{j\neq i} \pi_{j,t} p_{j,t} - 0.5\pi_{i,t}}{1-2\alpha\pi_{i,t}} + \frac{0.5}{1-2\alpha}\big)| + \frac{\alpha}{2} \\
&\leq \frac{2\alpha(1-\alpha)(1-\pi_{i,t})}{(1-2\alpha\pi_{i,t})(1-2\alpha)} b_{i,t} + \alpha\big(\frac{\sum_{j\neq i} \pi_{j,t} p_{j,t} - 0.5\pi_{i,t}}{1-2\alpha\pi_{i,t}} + \frac{0.5}{1-2\alpha} + \frac{1}{2}\big).
\end{aligned}$$

Assuming $\alpha < 0.25$, we have:

$$|p_{i,t} - f_{i,t}(p^*_{i,t})| \leq 8\alpha b_{i,t} + 3.5\alpha = (8b_{i,t} + 3.5)\alpha \leq 11.5\alpha.$$

Using the regret bound of the WSU algorithm of [89], we have:

$$\sum_{t=1}^{T} \ell(\sum_{j=1}^{K} \pi_{j,t} p_{j,t}, r_t) - \min_{i\in[K]} \sum_{t=1}^{T} \ell(p_{i,t}, r_t) \leq \mathcal{O}(\sqrt{T \ln K}).$$

For quadratic loss function $\ell(p, r) = (p - r)^2$, the function is 2-Lipschitz. Therefore, the following inequalities hold for all $t \in [T]$ and $i \in [K]$:

$$|\ell(f_{i,t}(\pi_t^T p_t), r_t) - \ell(p_{i,t}, r_t)| \leq 2|f_{i,t}(\pi_t^T p_t) - p_{i,t}| \leq \alpha\pi_{i,t} \leq \alpha,$$

$$|\ell(p_{i,t}, r_t) - \ell(f_{i,t}(p^*_{i,t}), r_t)| \leq 2|p_{i,t} - f_{i,t}(p^*_{i,t})| \leq 23\alpha.$$

Putting the above results together, the regret bound is $\mathcal{O}(\sqrt{T \ln K}) + \mathcal{O}(\alpha T)$. Therefore, plugging in the value of α, we obtain the $\mathcal{O}(\sqrt{T \ln K})$ regret bound as stated. ■