博士論文

Online Optimization with Limited Information
(限られた情報を用いたオンライン最適化)

伊藤　伸志

Online Optimization with Limited Information

Shinji Ito

March 2, 2020

Abstract

In this thesis, we study online optimization problems in which we can get only partial feedback on objective functions. Our aim is twofold: One is to provide sophisticated methods for information-limited online optimization problems that are computationally efficient and achieve small regrets, i.e., perform almost as well as optimal non-adaptive strategies. The other aim is to assess the "price of information" and the "price of computation" in the context of online optimization, i.e., how the feedback information and computational resources affect the optimality of decision making, so as to obtain a deeper understanding of the relationship among the following three concepts: information, computation, and optimality. To attain these goals, we consider different problem settings, for each of which we provide algorithms and analyze complexity.

This thesis presents algorithms for certain problem settings of information-limited online optimization, which include linear (combinatorial) optimization, submodular function minimization, convex optimization, and portfolio selection with combinatorial constraints. Our algorithms are superior to existing ones either in terms of regret upper bounds (how better decisions are) or of computational efficiency, or of both. We also offer lower bounds for regret and computational complexity as well. By comparing lower and upper bounds in each problem setting, we can assess the optimality of algorithms.

One highlight of the thesis is a computationally efficient algorithm for bandit linear optimization. Bandit optimization is an online optimization problem in which we can observe only the value of the objective function at the chosen solution. For this problem, there have been algorithms achieving optimal performance in terms of regret bounds. Such algorithms, however, require exponential-time computation for certain problems, and a computationally efficient optimal algorithm remains to be provided. We address this need by constructing a computationally efficient algorithm that achieves an optimal regret bound for general bandit linear optimization. Our analysis implies that the computational complexity of bandit linear optimization is equivalent to that for underlying offline optimization, i.e., we do not have to sacrifice computational efficiency even when feedback information is extremely limited.

We also consider submodular function minimization in which only a noisy evaluation oracle is available. For this problem, we provide a polynomial-time algorithm with a better error bound than existing algorithms. We also show that, under some assumptions, there is no room for improving our algorithm in terms of error bounds, by providing lower bounds of errors for arbitrary algorithms. In this, we have produced the first algorithm with optimal error (regret) bounds in the context of bandit submodular minimization.

This thesis covers other problem settings with nonlinear objectives, including bandit convex optimization and online portfolio selection. For bandit convex optimization, we present a novel algorithm that enjoys a minimax optimal regret bound under some assumptions, e.g., strong convexity and smoothness of objectives. Our algorithm is the first to achieve, under mild assumptions, an optimal regret bound of even for constrained problems. For online portfolio selection, we introduce a new problem setting in which the available combination of assets is restricted. We provide algorithms with regret upper bounds, and, by providing tight regret lower bounds, show that our algorithms are nearly optimal. Our regret analyses imply that the quantity of feedback information has a large impact on the quality of decision making, in online portfolio selection problems.

Acknowledgment

I have been supported by many people during my Ph.D. studies.

First and foremost, I am grateful for my advisor Satoru Iwata. He gave me many valuable comments and advice whenever I needed them. I have learned a lot from his faithful and ardent attitude toward research and education. In particular, the work in Chapter 5 could not be completed without his guidance.

I would like to thank my collaborators, Ryohei Fujimaki, Takuro Fukunaga, Daisuke Hatano, Naonori Kakimura, Ken-ichi Kawarabayashi, Hanna Sumita, Kei Takemura, and Akihiro Yabe. It was a great pleasure to discuss with them almost every week, from which numerous ideas and signs of progress have come. Many parts of this thesis, Chapters 3, 4, 6 and 8, result from joint works with these brilliant researchers. I am looking forward to furthering collaboration with them.

I thank Hiroshi Hirai, Hiroshi Imai, Kunihiko Sadakane, Taiji Suzuki and Akiko Takeda for reading this thesis, and for offering many helpful and insightful comments.

Many thanks to the members of Mathematical Informatics Laboratories #7. Discussion with Kaito Fujii and Tasuku Soma provided many inspirations for my research. Random topics with Taihei Oki and Nobutaka Shimizu were very interesting, and thanks to them, I could learn about many topics beyond my specialty.

I want to thank Masaaki Sugihara and Kazuo Murota, who supervised me during the course of undergraduate and master students. It was a great pleasure to start research under their supervision. They taught me a lot of things, including a perspective on research and life.

I was fortunate to be in the research environment of NEC Corporation. I would like to thank Yutaro Yamaguchi for giving me an opportunity to enter the company. I thank Ryohei Fujimaki and Satoshi Morinaga for inviting me to the company and for offering many topics to work on. My life in the company was particularly enjoyable thanks to Tatsuya Matsuoka, Naoto Ohsaka and Tomoya Sakai.

I want to thank the University of Tokyo for allowing me to study and do research. I gratefully appreciate the financial support of JST ACT-I.

Finally, I would like to thank my family, especially my parents Fumio and Toshie, for their moral support and warm encouragement.

Contents

Chapter 1

Introduction

Mathematical optimization plays a central role in a variety of research areas, including operations research, statistics, finance, signal processing, data analysis, and machine learning. In these areas, numerous methods have been developed for efficiently finding optimal or approximately optimal solutions. Most such methods are designed to receive explicit objective functions and constraints as input.

By way of contrast, in many real-world problems, no explicit objective may have been given at the moment of decision making, and in such cases, standard optimization methods would not directly apply. One typical example of this would be the *online prediction* problem. The goal of this problem is to predict the label of samples in a sequential manner, which would have a wide range of applications, such as recommendation systems and classification of spam email messages. Although many of these learning problems can be regarded as convex optimization problems, a learner is not able to assess the objective at the moment of prediction, and the objective often changes because the data, which provide objective functions, come sequentially. Similarly, in *online portfolio selection* problems, in which a player sequentially chooses ways of allocating assets, the price relatives (return on investment) that define objective functions are observed only after determining the investment, and, consequently, one cannot assess the objective before outputting solutions.

Such situations can be dealt with in the framework of *online optimization*, in which a player sequentially makes decisions before the objectives are revealed and improves strategies throughout multiple decision-making rounds. More precisely, in each round $t \in \{1, 2, \ldots\}$, a player chooses a solution x_t and then observes the objective function f_t repeatedly, aiming to minimize, or maximize, the sum of $f_t(x_t)$ for $t = 1, 2, \ldots, T$, where T stands for a given time horizon. In a standard setting, information about f_t is given just after choosing x_t. The performance of players is evaluated in terms of *regret*, which is defined as the difference between the achieved cumulative objective values $f_t(x_t)$ and that for an "optimal" strategy, which corresponds to a sort of benchmark. From this definition, smaller regrets imply that the performance of the output sequence is close to that with the optimal strategy. The definitions of benchmark optimal strategies are different depending on problem settings. A standard choice for it is the solution that minimizes the cumulative objective among all non-adaptive strategies.

Although online optimization is a promising approach for situations in which available information is restricted at the moment of decision making, we often face more difficult problems, ones in which the information is too restricted to apply a framework of standard online optimization. For example, in an application of online prediction problems, if the cost for observing features of examples is quite high, we need to make predictions with limited feature observations. Such scenarios may arise in the context of medical diagnoses of diseases [101], where each feature will correspond to the result of a medical test. This restriction means that we can observe only a part of the objective function even after choosing actions. A similar issue arises in online portfolio selection problems as well. In, for example, investment in commercial advertising or research and development, we cannot get feedback about the policies that have not been undertaken, which implies that we will not be able to assess all the values of the objective function even after the decision making has been completed. That is to say, in certain real-world applications, we will not be able to get complete information even after decision making has been done, and that standard online optimization methods will not directly apply.

This dissertation pursues two goals. One is to construct a sophisticated methodology for online optimization with limited information, in terms both of regret bounds (how better decisions are) and of computational efficiency, while the other is to properly assess the "price of information" and the "price of computation" in the context of online optimization, i.e., how the quantities of feedback information and computational resources affect the optimality of decision making, so as to obtain a deeper understanding of the relationship among the following three concepts: information, computation, and optimality. In an effort to attain these goals, we consider different problem settings, for each of which we provide algorithms and analyze complexity.

The contribution of this dissertation includes online algorithms for linear (combinatorial) optimization, submodular function minimization, convex optimization, and portfolio selection with combinatorial constraints. Our algorithms are superior to existing ones either in terms of regret upper bounds (or error bounds) or of computational efficiency, or of both. We offer lower bounds for regret and computational complexity as well. By comparing lower and upper bounds for different problem settings, we are able to assess the optimality of algorithms and evaluate the price of information and the price of computation. One highlight of the thesis is an oracle efficient algorithm for bandit linear optimization, which implies that the computational complexity of bandit linear (combinatorial) optimization is equivalent to that for underlying offline optimization, i.e., we do not have to sacrifice computational efficiency even when feedback information is extremely limited.

1.1 Review on Online Optimization

One of the most simple and fundamental examples of online optimization is the *prediction from expert advice problem* [11, 60, 23], in which the feasible region is a finite set of size d and the objective functions correspond to d-dimensional vectors. For this problem, Hannan [74] provided a framework of regret analyses in the 1950s. In terms of regret bounds, an optimal algorithm for the problem is the algorithm called *multiplicative weight update (MWU)* method. Algorithms similar to MWU can be found in the literature in the early 1950s in the context of game theory [29, 28, 144]. MWU has been independently rediscovered in other fields including computational geometry, e.g., Clarkson's algorithm for linear programming [43]), and machine learning, e.g., Winnow algorithm [124], Hedge algorithm and AdaBoost [60]). The method and analysis of MWU has provided basic tools for constructing many algorithms for other online optimization problems, including online convex optimization [34, 80], online combinatorial optimization [38, 52], online principal component analysis [165], and so on. MWU is used in Chapters 3 and 7 of this dissertation as well. For more details in MWU, refer to, e.g., the review by Arora et al. [11].

One of the most simple and fundamental examples of online optimization is the *prediction from expert advice* problem [11, 60, 23], in which the feasible region is a finite set of size d and the objective functions correspond to d-dimensional vectors. For this problem, Hannan [74] provided a framework of regret analyses in the 1950s. In terms of regret bounds, an optimal algorithm for the problem is referred to as a *multiplicative weight update (MWU).* Algorithms similar to MWU can be found in literature from the early 1950s in the context of game theory [29, 28, 144]. MWU has been independently rediscovered in other fields, including computational geometry, e.g., Clarkson's algorithm for linear programming [43], and machine learning, e.g., Winnow algorithm [124], Hedge algorithm and AdaBoost [60]. The method and analysis offered by MWU has provided basic tools for constructing many algorithms for other online optimization problems, including online convex optimization [34, 80], online combinatorial optimization [38, 52], and online principal component analysis The use of MWU is considered in Chapters 3 and 7 of this dissertation as well. For more details regarding MWU, refer to, e.g., the review by Arora et al. [11].

The *multi-armed bandit (MAB)* problem is a variant of the prediction from expert advice problem, in which only the reward (or loss) for a chosen action is observable in each round. The MAB framework was introduced in the 1930s by Thompson [159], and his proposed algorithm is now called *Thompson sampling.* Roughly two decades later, MAB was formally restated by Robbins [143], in which the notion of regret was introduced. In addition to Thompson sampling, the *upper confidence bound (UCB)* algorithm proposed by Lai [117] is also a popular algorithm for MAB. It has been shown that Thompson sampling, UCB algorithms, and their variants perform optimally in terms of certain some different measures, such as asymptotic optimality or minimax optimality, for stochastic settings [15, 105]. For non-stochastic settings, Auer et al. [16] proposed an algorithm called *EXP3* with a nearly minimax optimal regret bound, based on MWU. Though EXP3 retained a logarithmic gap between upper and lower bounds, this gap was shaved off by Audibert and Bubeck [12], who proposed a modified algorithm *INF* that achieves a completely tight regret bound. Besides its improved regret bound, the INF algorithm also turned out to have a remarkable side effect. Zimmert and Seldin [168] proved that a special case of INF works optimally not only in non-stochastic settings but also in stochastic settings. As a more general model in terms of observations, Alon et al. [10] considered MAB with *graph-structured feedback*, in which observable values for chosen actions are described by a directed graph. This work provided a complete characterization for minimax regret bounds depending on the underlying graph. For other results and recent progress in MAB, see., e.g., the book by Lattimore and Szepesvári [118].

For general online optimization problems with linear objective functions, Kalai and Vempala [100] presented a meta-algorithm called *follow the perturbed leader (FPL)* that achieves $O(\sqrt{T})$-regret by calling on an algorithm for underling (offline) optimization problems. For example, FPL achieves $O(\sqrt{T})$-regret for *online shortest path* problems [157, 17], by solving shortest path problems for T times. This implies that there exists a polynomial-time algorithm for an online optimization problem given that for the offline problem, and that the computational complexity of offline problems and of online problems are equivalent under polynomial-time reduction. Such reduction of online optimization to offline optimization has been extended to a more general setting in which only approximate algorithms are available [98, 82, 66]. For settings in which the observation is restricted, however, some open problems remain, as noted in [66, 78]. Details regarding open problems and recent progress are discussed in Chapter 3.

Online optimization problems with convex objective functions, or *online convex optimization*, have garnered much attention with their application to machine learning. For non-stochastic online convex optimization, Zinkevich [169] showed that the *online gradient descent* method enjoys an $O(\sqrt{T})$ regret bound. This bound turned out to be optimal order for general convex functions because an arbitrary algorithm suffers regrets of $\Omega(\sqrt{T})$ in the worst case even for the linear objective functions Cesa-Bianchi and Lugosi [37]. On the other hand, for special cases or problems with strongly-convex objective functions, one can obtain much better regret bounds. For example, Cover [47] considered the *online portfolio selection* problem, which is a special case of online convex optimization, and proposed an algorithm achieving $O(\log T)$-regret with an exponentially large computational cost. The computational efficiency was improved by Kalai and Vempala [100] later. These results were extended by Hazan et al. [81] who proposed the *online Newton step* algorithm, which achieves $O(\log T)$-regret for *exp-concave* objective functions with polynomial-time computation. Since the class of exp-concave functions includes strongly-convex functions and the objectives of online portfolio selection, the online newton step is useful for a broad class of problems.

Online sparse linear regression is an example of online optimization with limited information, which is an online learning problem in which a learner can observe only a limited number of feature values. This problem was posed as an open problem by Kale [101] and then was proved by Foster et al. [59], to be, in general, computationally hard. By way of contrast, under some assumptions, such as the restricted isometry property [35], computationally efficient algorithms have been proposed [102, 89]. Murata and Suzuki [134] have proposed a more sample-efficient algorithm for this problem.

In the context of more general problem settings of *online computation* including online optimization, the *competitive ratio* [57, 24], besides the regret, is adopted for measuring the performance of algorithms. The competitive ratio of an online algorithm is defined to be the ratio between its performance (the value of the objective function) and the offline algorithm's performance. In other words, the competitive analysis focuses on a multiplicative error while the regret analysis focuses on an additive error. When a competitive ratio of tends to 1, regret analysis provides information regarding its convergence rate, which implies that regret bounds offer a more detailed analysis in this case. In a study by Agrawal and Devanur [7], a connection between a class of online computation and online optimization has been established, by which regret bounds for online optimization lead to a competitive ratio for online packing. Fujii and Kashima [61] have recently proposed a stream-based online algorithm for adaptive submodular maximization [62, 68].

1.2 Contribution of This Dissertation

This dissertation considers certain problem settings in online optimization in which available information is restricted. In particular, we provide online optimization algorithms and complexity analyses for (i) linear optimization, including combinatorial optimization with bandit feedback, (ii) submodular function minimization with a stochastic evaluation oracle, (iii) convex optimization with bandit feedback, and (iv) portfolio selection with combinatorial constraints. The efficiency of the algorithms is evaluated in terms of two viewpoints: regret upper bounds, i.e., how much better the decisions are, and computational costs. Similarly, the complexity of the problems is evaluated in terms of information-theoretic regret lower bounds and computational complexity.

1.2.1 Linear Optimization

Online linear optimization is an online optimization problem with linear objective functions. Problem setting includes such combinatorial optimization problems as the online shortest path problem. In this problem, for example, a player is first given a directed graph without weights, as well as two vertices, s and t, corresponding, respectively, to source and destination. Throughout multiple rounds, the player repeatedly chooses an s-t path and then observes the weights of edges, which may change every round, and aims to minimize the sum of the weights of all paths. This shortest path problem is a special case of linear optimization since the weight of the path can be regarded as a value of a linear function in the indicator vector of the path as a set of edges. Similarly, online linear optimization includes various combinatorial optimization problems as minimum spanning trees, maximum weight matching, minimum cut, and knapsack problems.

We focus on online linear optimization with *bandit feedback*, which means that the player can observe only the value of the objective function $f_t(x_t)$ at the chosen solution. Online optimization problems with bandit feedback are called *bandit optimization problems.* In a bandit shortest path problem, the player can observe the weight of the chosen path alone in each round, in contrast to a situation in which the weights of all edges are observable, as in standard online shortest path problems. Bandit optimization problems are more difficult than standard online optimization problems, as available information is more restricted.

For bandit linear optimization, there have been algorithms with $O(\sqrt{T})$ regret upper bounds. Further, it has been known that any algorithm will suffer regrets of $\Omega(\sqrt{T})$ in the worst case. Hence, there is no room to improve the regret upper bound of $O(\sqrt{T})$, and algorithms with $O(\sqrt{T})$-regret are optimal algorithms in terms of worst-case regret. However, existing optimal algorithms require exponential-time computation for certain problems, and a computationally efficient optimal algorithm remains to be provided.

We address this need by constructing a computationally efficient algorithm that achieves an optimal regret bound for general bandit linear optimization. More precisely, our algorithm calls on an oracle that solves an offline optimization problem only polynomial times and runs in polynomial-time computation, except for the calling on of the oracle. This means that our algorithm runs in polynomial time under the minimum assumption that the corresponding offline optimization can be solved in polynomial time. Our results imply that the computational complexity of bandit optimization problems is equivalent to that of corresponding offline problems, as well, w.r.t. polynomial-time reduction.

1.2.2 Submodular Function Minimization

Submodular function minimization is an important problem in the research on combinatorial optimization, and several algorithms have been developed for it. Existing works are based on a computational model in which one can access an *evaluation oracle* that returns the value of the objective submodular function given a concrete value of the variable. On the other hand, we cannot always assess the exact value of the objective, and the observable value includes noise. To cope with such situations, we first consider submodular function minimization problems with stochastic noisy evaluation oracles.

Our problem setting is new, but it can be regarded as a special case of *bandit submodular minimization* in which we can observe values of submodular objective functions that may change every round and in which performance is evaluated in terms of regret. For this more general problem setting, Hazan and Kale [78] have proposed algorithms that achieve sublinear regret bounds, and regret lower bounds are also known. There has been, however, a gap of $O(T^{1/6})$ between the upper and lower bounds, which implies that there might remain room for improving the algorithms. The optimal rate of regret remains an open question.

Our contribution is to develop polynomial-time algorithms for submodular function minimization with noisy evaluation oracles. We show that they have better error bounds than existing algorithms and that, under some assumptions, there is no room for improving our algorithms in terms of accuracy. We provide lower bounds of errors for arbitrary algorithms that match the upper bounds obtained with our algorithms. In this we have produced the first algorithms with optimal error (regret) bounds in the context of bandit submodular minimization.

We also introduce the *price optimization problem*, an application of submodular function minimization. In this problem, we consider selling d types of products, and the goal is to find the best pricing strategy that maximizes the gross profit. A key observation here is the connection between supermodularity of the revenue and cross elasticity of demand, which means that submodular function minimization can be applied to the price optimization problem, under assumptions regarding cross elasticity of demand.

1.2.3 Convex Optimization

Online convex optimization problems are an important class of online optimization problems in that they have a wide range of applications, including online learning and online portfolio selection. For this class of problems, a simple method called the online gradient descent method [169] works well and achieves optimal regret bounds under certain conditions.

For bandit convex optimization, however, the optimal regret bound remains to be established. The current state-of-the-art has been reported by Bubeck et al. [34], which achieves $\tilde{O}(d^{9.5}\sqrt{T})$-regret, and they have conjectured that the optimal regret bound would be $\tilde{O}(d^{1.5}\sqrt{T})$, where d stands for the dimensionality of the feasible region. There have as yet, however, been no significant subsequent improvements.

We focus on an important special case of bandit convex optimization in which the objective functions are strongly convex and smooth. For this problem, there exists an algorithm [58, 5] that achieves $O(d\sqrt{T})$-regret under the strong assumption that the problem is unconstrained, i.e., that the feasible region is a linear space. This rate is optimal because a lower bound of $\Omega(d\sqrt{T})$ is known. For such constrained problems, however, the optimal regret bound has yet to be established. The best-known algorithm for constrained problems has been proposed by Hazan and Levy [80]; it has a regret bound of $\tilde{O}(d^{1.5}\sqrt{T})$ with an $O(d^{1/2})$-gap from the lower bound and does not have computationally efficient implementation for the arbitrary feasible region.

Our contribution includes a new algorithm for bandit convex optimization with strongly-convex and smooth objective functions. This algorithm is the first to achieve, under mild assumptions, an optimal regret bound of $\tilde{O}(d\sqrt{T})$ even for constrained problems. Further, it runs in polynomial time under the minimal assumption that we have access to a membership oracle for the feasible region, i.e., the assumption that one can efficiently determine whether a given point is feasible or not.

1.2.4 Portfolio Selection

We also consider online portfolio selection, which in practice is an important special case of online convex optimization. Work on online portfolio selection was initiated by Cover [47], much progress has been made [81, 139, 53, 167] and there have been successful applications [122]. Among these studies, one of the most remarkable studies relates to the *online newton method* [81], which is the polynomial-time algorithm that achieves $O(d \log T)$-regret for online portfolio selection. Since there is a lower bound of $\Omega(d \log T)$ [139], the online newton method achieves an optimal regret bound.

A major restriction in existing algorithms comes from the requirement for the strong assumption that one can choose portfolios of arbitrary combinations of assets and that the price relatives (returns of investment) for all assets are observable. This assumption does not hold in many real-world applications, such as investment in commercial advertising, and, hence, in many cases, the existing algorithm cannot be directly applied.

To overcome this restriction, we introduce new problem settings featuring *online portfolio selection with combinatorial constraints*, and we propose relevant algorithms. In our problem settings, we are given a subset family of the available combination of assets, and managed portfolios consist of a combination available in an online manner. Our model includes two different settings: *full-feedback* and the *bandit-feedback*. In the former, one can observe price relatives for all assets, and in the latter, one can observe price relatives only for invested assets that are included in the chosen portfolio. We provide an algorithm with a regret upper bound for each problem setting, and, by providing tight regret lower bounds, show that our algorithms are nearly optimal in terms of regret bounds. Our regret bounds imply that the bandit-feedback setting is much harder than the full-feedback setting, as there is an exponentially large gap between optimal regret bounds for these two settings. Furthermore, our results include a lower bound for computational complexity. Under the assumption of $\mathbf{BPP} \not\subseteq \mathbf{NP}$, we cannot reduce the computational time of our algorithm into a polynomial.

1.3 Organization of This Dissertation

Chapter 1.3 introduce some existing results on online optimization, which are used in or related to the following chapters. Chapter 3 considers bandit linear optimization with bandit feedback. In this chapter, we present computationally efficient algorithms that achieve nearly optimal regret bounds. The contents in this chapter are included in the paper [92]. In Chapter 4, we study the statistical complexity of bandit combinatorial optimization, to provide tight regret lower bounds for various problem settings. Contents in Chapter 4 can be found in [91]. Chapter 5 considers submodular function minimization with noisy evaluation oracle. We provide algorithms and give lower bounds of complexity, and discuss the optimality of algorithms. The paper [85] includes the contents of this chapter. Chapter 6 introduces the price optimization problem [87, 88], an application of submodular function minimization. In Chapter 7, we move on to problems with convex objective functions. We propose a novel algorithm for bandit convex optimization with strongly-convex and smooth objectives and analyze regret bounds for the algorithm. The contents of this chapter are included in the paper [86]. Chapter 8 is devoted to online portfolio optimization with combinatorial constraints. The contents in this chapter has been published as the literature [90]. In Chapter 9, we conclude this dissertation and note remained open questions.

Chapter 2

Existing Results on Online Optimization

This chapter introduces the framework of online optimization and existing results.

2.1 Problem Setting of Online Optimization

Problems of online optimizations are specified by a *feasible region* $\mathcal{A}$ and a class of objective functions $\mathcal{F} \subseteq \{f : \mathcal{A} \to \mathbb{R}\}$. In an online optimization problem, a player, or a decision-maker, follows the protocol given in Algorithm 1: The player is first given the time horizon T corresponding to the number of *rounds* of decision making. In each round $t \in \{1, 2, \ldots, T\}$, the environment chooses the objective function $f_t \in \mathcal{F}$, and at the same time, the player chooses the action a_t. The environment then reveals feedback information on the objective f_t. The type of feedback depends on problem settings: in the standard *full-feedback* setting, complete information about f_t is revealed to the player, i.e., the player can assess $f_t(x)$ for all $x \in \mathcal{A}$ after deciding a_t. In the *bandit-feedback* setting, on the other hand, only the value of $f_t(a_t) \in \mathbb{R}$ is revealed. Besides these settings of feedback, other settings have been considered to model more various problems, e.g., semi-bandit feedback or graph-structured feedback. The player then incurs the loss of $f_t(a_t)$. The goal of the player is to minimize (or maximize) the cumulative loss $\sum_{t=1}^{T} f_t(a_t)$ over all rounds $t \in \{1, 2, \ldots, T\}$.

In terms of environments, we consider mainly two settings of *stochastic environments* and *non-stochastic (or adversarial) environments*. In the setting of stochastic environments, we assume that there is an unknown distribution $\mathcal{D}$ over $\mathcal{F}$, and that f_t follows $\mathcal{D}$ independently for $t \in \{1, 2, \ldots, T\}$. In the non-stochastic environment setting, we do not assume distributions of f_t, and f_t can changes arbitrarily. More precisely, we can consider two types of non-stochastic environment: *oblivious* one chooses $\{f_t\}_{t=1}^{T}$ arbitrarily before the game starts, and *reactive* one chooses each f_t depending on the history $f_1, a_1, \ldots, f_{t-1}, a_{t-1}$. The reactive non-stochastic environment models more general and harder problems than the oblivious one. In this thesis, we suppose the reactive one when dealing with non-stochastic settings.

The goal of online optimization is to provide a better algorithm for the player, of which

Algorithm 1 A template of online optimization

```
Require: Time horizon $T \in \mathbb{N}$, feasible region $\mathcal{A}$, class of objective functions $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{A}}$.
1: The player is given the time horizon $T$.
2: for $t = 1, 2, \ldots, T$ do
3:     The environment chooses the next objective function $f_t \in \mathcal{F}$, which is not revealed to the player at this moment.
4:     The player chooses the action $a_t \in \mathcal{A}$.
5:     The environment reveals information about $f_t$, e.g.,
           • the player obtains complete information about $f_t$ in the full-feedback setting (standard online optimization), or
           • the player observes only $f_t(a_t)$ in the bandit-feedback setting.
6:     The player incurs a loss of $f_t(a_t)$.
7: end for
```

performance is evaluated in terms of *regret* $R_T(a^*)$ defined by

$$R_T(a^*) = \sum_{t=1}^{T} f_t(a_t) - \sum_{t=1}^{T} f_t(a^*) \tag{2.1}$$

for arbitrary $a^* \in \mathcal{A}$. If the environment behaves randomly or if the algorithm is a randomized one, we consider the expectation $\mathbf{E}[R_T(a^*)]$ with respect to the randomness of the environment and the algorithm's internal randomization. This thesis focuses on the following *worst-case regret*:

$$\max_{\text{environment}} \max_{a^* \in \mathcal{A}} \mathbf{E}[R_T(a^*)]. \tag{2.2}$$

From the definition of the stochastic/non-stochastic environments, the worst-case regrets for different assumptions on environments satisfy the following:

$$\begin{aligned}
&\max_{\mathcal{D}\text{: distribution over } \mathcal{F}} \max_{a^* \in \mathcal{A}} \mathop{\mathbf{E}}_{\{f_t\}_t^T \sim \mathcal{D}^T} [R_T(a^*)] \quad \text{(stochastic environments)}\\
\le &\max_{\mathcal{D}\text{: distribution over } \mathcal{F}} \mathop{\mathbf{E}}_{\{f_t\}_t^T \sim \mathcal{D}^T} \left[\max_{a^* \in \mathcal{A}} R_T(a^*)\right] \quad \text{(stochastic environments)}\\
\le &\max_{\{f_t\}_{t=1}^T \in \mathcal{F}^T} \max_{a^* \in \mathcal{A}} \mathbf{E}[R_T(a^*)] \quad \text{(oblivious non-stochastic environments)}\\
= &\max_{\{f_t\}_{t=1}^T \in \mathcal{F}^T} \mathbf{E}\left[\max_{a^* \in \mathcal{A}} R_T(a^*)\right] \quad \text{(oblivious non-stochastic environments)}\\
\le &\max_{\{f_t\}_{t=1}^T\text{: reactive non-stochastic}} \max_{a^* \in \mathcal{A}} \mathbf{E}[R_T(a^*)] \quad \text{(reactive non-stochastic environments),}
\end{aligned}$$

where all the expectation is taken w.r.t. the algorithm's internal randomization and the environment's randomization. We note that, under the assumption that the complexity of $\mathcal{F}$ is small enough, the value $\mathbf{E}_{\{f_t\}_t^T \sim \mathcal{D}^T}[\max_{a^* \in \mathcal{A}} R_T(a^*)]$ is close to $\max_{a^* \in \mathcal{A}} \mathbf{E}_{\{f_t\}_t^T \sim \mathcal{D}^T}[R_T(a^*)]$.

2.2 Online Optimization and Offline Optimization

To contrast with online optimization problems, in this thesis, we use the term *offline optimization* to stand for the standard optimization problems of finding $a^* \in \arg\min_{a \in \mathcal{A}} f(a)$ given $f \in \mathcal{F}$. In this section, we see the relationship between online optimization and offline optimization.

If we have an algorithm for online optimization, then we have a fully polynomial-time randomized approximation scheme (FPRAS) for offline optimization, in terms of expected additive errors.

Proposition 2.2.1. *Suppose that there exists an algorithm for the stochastic online optimization problem specified by $\mathcal{A}$ and $\mathcal{F}$, such that $\mathbf{E}[R_T(a^*)] \leq b_T$ for all $a^* \in \mathcal{A}$. Then we have an algorithm for offline optimization such that given $f \in \mathcal{F}$, find an approximate solution $\hat{a} \in \mathcal{A}$ such that $\mathbf{E}[f(\hat{a})] \leq \min_{a \in \mathcal{A}} f(a) + b_T/T$.*

Proof. Suppose an environment for the online optimization defined by $f_t = f$ for all $t \in [T]$. Then, from the definition (2.1) of the regret, the output a_t of the online optimization algorithms satisfies

$$\frac{1}{T}\mathbf{E}\left[\sum_{t=1}^{T} f(a_t)\right] = \frac{1}{T}\left(\sum_{t=1}^{T} f(a^*) + \mathbf{E}[R_T(a^*)]\right) \leq \frac{1}{T}\sum_{t=1}^{T} f(a^*) + \frac{b_T}{T} = f(a^*) + \frac{b_T}{T}, \tag{2.3}$$

for arbitrary $a^* \in \mathcal{A}$. Accordingly, we have $\frac{1}{T}\mathbf{E}[\sum_{t=1}^{T} f(a_t)] \leq \min_{a \in \mathcal{A}} + b_T/T$. If we set $\hat{a} \in \arg\min\{f(a) \mid a \in \{a_t\}_{t=1}^{T}\}$, we have

$$\mathbf{E}[f(\hat{a})] = \frac{1}{T}\mathbf{E}\left[\sum_{t=1}^{T} f(\hat{a})\right] \leq \frac{1}{T}\mathbf{E}\left[\sum_{t=1}^{T} f(a_t)\right] \leq \min_{a \in \mathcal{A}} f(a) + \frac{b_T}{T}. \tag{2.4}$$

□

Corollary 2.2.1. *Let $\alpha, \beta \geq 0$, $\gamma \geq 1$ and $\delta \in (0, 1)$. Suppose that there exists an online optimization algorithm that achieves $\mathbf{E}[R_T] \leq O(n^{\alpha} T^{1-\delta})$ with $O(n^{\beta} T^{\gamma})$-time computation, where n is a parameter standing for the size of inputs. Then there exists an FPRAS for offline optimization such that, given $\varepsilon > 0$ and $f \in \mathcal{F}$, it outputs $\hat{a}$ satisfying $\mathbf{E}[f(\hat{a})] \leq \min_{a \in \mathcal{A}} f(a) + \varepsilon$ in $O(n^{(\beta + \alpha\gamma/\delta)} \varepsilon^{\gamma/\delta})$-time computation.*

This corollary implies that, if underlying offline optimization is computationally hard, it is hopeless to develop a computationally efficient algorithm that achieves a regret bound sublinear in T.

If the set $\mathcal{F}$ of objective functions consists of convex functions, the above result extends to the *stochastic optimization*. Given $\mathcal{A}$ and $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{A}}$, a problem instance of a stochastic optimization problem is specified by a distribution $\mathcal{D}$ over $\mathcal{F}$. In stochastic optimization, we are given T i.i.d. samples from $\mathcal{D}$, and search for the minimizer of $f^*(x) := \mathbf{E}_{f \sim \mathcal{D}}[f(x)]$. If all members of $\mathcal{F}$ are convex functions, then a stochastic online optimization algorithm for $\mathcal{F}$ leads to a stochastic optimization algorithm.

Proposition 2.2.2. *Assume that $\mathcal{A} \subseteq \mathbb{R}^d$ is a convex set and that $\mathcal{F}$ consists of convex functions. Suppose that there exists an algorithm for the stochastic online optimization problem specified by $\mathcal{A}$ and $\mathcal{F}$, such that $\mathbf{E}[R_T(a^*)] \leq b_T$ for all $a^* \in \mathcal{A}$. Then we have an algorithm for stochastic optimization such that given a distribution $\mathcal{D}$ over $\mathcal{F}$, find an approximate solution $\hat{a} \in \mathcal{A}$ such that $\mathbf{E}[f^*(\hat{a})] \leq \min_{a \in \mathcal{A}} f^*(a) + b_T/T$, where $f^*(x) = \mathbf{E}_{f \sim \mathcal{D}}[f(x)]$.*

Proof. Suppose a stochastic environment for the online optimization such that $f_t \sim \mathcal{D}$, i.i.d. for $t \in [T]$. Then, by a discussion similar to the proof of Proposition 2.2.1, we have

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left[f_t(a_t)\right] \leq f^*(a^*) + \frac{b_T}{T} \tag{2.5}$$

for arbitrary $a^* \in \mathcal{A}$. Since f_t and a_t are independent, we have

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left[f_t(a_t)\right] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}[f^*(a_t)] = \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} f^*(a_t)\right] \tag{2.6}$$

Define $\hat{a}$ to be the average of $a_1, \ldots, a_T$, i.e., let $\hat{a} = \frac{1}{T}\sum_{t=1}^{T} a_t$. Since f^*, a convex combination of convex functions, is a convex function as well, from Jensen's inequality, we have

$$\frac{1}{T}\sum_{t=1}^{T} f^*(a_t) \geq f^*\left(\frac{1}{T}\sum_{t=1}^{T} a_t\right) = f^*\left(\hat{a}\right). \tag{2.7}$$

By combining (2.5), (2.6) and (2.7), we obtain

$$\mathbf{E}[f^*(\hat{a})] \leq \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} f^*(a_t)\right] \leq \min_{a \in \mathcal{A}} f^*(a) + \frac{b_T}{T}. \tag{2.8}$$

□

Corollary 2.2.2. *Assume that $\mathcal{A} \subseteq \mathbb{R}^d$ is a convex set and that $\mathcal{F}$ consists of convex functions. Let $\alpha, \beta \geq 0$, $\gamma \geq 1$ and $\delta \in (0,1)$. Suppose that there exists an online optimization algorithm that achieves $\mathbf{E}[R_T] \leq O(n^\alpha T^{1-\delta})$ with $O(n^\beta T^\gamma)$-time computation, where n is a parameter standing for the size of inputs. Then there exists an FPRAS for stochastic optimization such that, given $\varepsilon > 0$ and $f \in \mathcal{F}$, it outputs $\hat{a}$ satisfying $\mathbf{E}[f^*(\hat{a})] \leq \min_{a \in \mathcal{A}} f^*(a) + \varepsilon$ in $O(n^{(\beta + \alpha\gamma/\delta)} \varepsilon^{\gamma/\delta})$-time computation.*

2.3 Online Gradient Descent Method

The *online gradient descent* method is an algorithm for online optimization, which works under the assumption that each objective function f_t is convex and that a subgradient of each f_t is available. Algorithm 2 shows how the online gradient descent proceeds.

The update rule of Algorithm 2 can be expressed as

$$a_{t+1} \in \arg\min_{a \in \mathcal{A}} \|a - a'_{t+1}\|_2^2 = \arg\min_{a \in \mathcal{A}} \|a - (a_t - \eta g_t)\|_2^2 = \arg\min_{a \in \mathcal{A}} \{2\eta g_t^\top (a - a_t) + \|a - a_t\|_2^2\}. \tag{2.9}$$

Noting this expression, we obtain the following regret bound:

Proposition 2.3.1. *Assume that each f_t is a convex function. For the output of Algorithm* (2), *the regret $R_T(a^*)$ is bounded as*

$$R_T(a^*) \leq \frac{1}{2\eta}\|a_1 - a^*\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T} \|g_t\|_2^2 \tag{2.10}$$

for arbitrary $a^* \in \mathcal{A}$.

Algorithm 2 Online gradient descent method

```
Require: Time horizon $T \in \mathbb{N}$, convex feasible region $\mathcal{A} \subseteq \mathbb{R}^d$, learning rate $\eta > 0$.
1: Set initial point $a_1 \in \mathcal{A}$ arbitrarily.
2: for $t = 1, 2, \ldots, T$ do
3:     Play $a_t$ and incur loss $f_t(a_t)$.
4:     Observe $g_t \in \partial f_t(a_t)$, a subgradient of $f_t$ at $a_t$.
5:     Update $a_t$ by $a'_{t+1} = a_t - \eta g_t$.
6:     Project $a'_{t+1}$ onto $\mathcal{A}$ by Euclidean projection, i.e., by $a_{t+1} \in \arg\min_{a \in \mathcal{A}} \|a - a'_{t+1}\|_2^2$.
7: end for
```

Proof. Since g_t is a subgradient of f_t at a_t, the regret $R_T(a^*)$ can be bounded as

$$R_T(a^*) = \sum_{t=1}^{T}(f_t(a_t) - f_t(a^*)) \leq \sum_{t=1}^{T} g_t^\top (a_t - a^*). \tag{2.11}$$

Since it holds that $\|a'_{t+1} - a^*\|_2^2 = \|a_t - \eta g_t - a^*\|_2^2 = \|a_t - a^*\|_2^2 - 2\eta g_t^\top (a_t - a^*) + \eta\|g_t\|_2^2$, we have

$$g_t^\top (a_t - a^*) \leq \frac{1}{2\eta}(\|a_t - a^*\|_2^2 - \|a'_{t+1} - a^*\|_2^2) + \frac{\eta}{2}\|g_t\|_2^2. \tag{2.12}$$

Since $a^* \in \mathcal{A}$, from the Pythagorean theorem (Theorem 2.1 in [77]), we have $\|a'_{t+1} - a^*\|_2 \geq \|a_{t+1} - a^*\|_2$. From this and (2.12), we have

$$g_t^\top (a_t - a^*) \leq \frac{1}{2\eta}(\|a_t - a^*\|_2^2 - \|a_{t+1} - a^*\|_2^2) + \frac{\eta}{2}\|g_t\|_2^2.$$

By taking the same of the above for $t = 1, \ldots, T$, we obtain

$$\begin{aligned}\sum_{t=1}^{T} g_t^\top (a_t - a^*) &\leq \frac{1}{2\eta}\sum_{t=1}^{T}(\|a_t - a^*\|_2^2 - \|a_{t+1} - a^*\|_2^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_2^2. \\ &= \frac{1}{2\eta}(\|a_1 - a^*\|_2^2 - \|a_{T+1} - a^*\|_2^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_2^2 \\ &\leq \frac{1}{2\eta}\|a_1 - a^*\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_2^2.\end{aligned}$$

By combining this and (2.11), we obtain (2.10). □

Corollary 2.3.1. *Assume that each f_t is a convex function, and that there exists $D, G > 0$ such that $\|a_1 - a^*\|_2 \leq D$ and $\|g_t\|_2 \leq G$ hold for all $a^* \in \mathcal{A}$ and for all $t \in [T]$. If η is chosen as $\eta = B\sqrt{T}/G$, then the regret is bounded as $R_T(a^*) \leq BG\sqrt{T}$ for all $a^* \in \mathcal{A}$.*

Note that the regret upper bounds in Proposition 2.3.1 and Corollary 2.3.1 apply even to the non-stochastic setting.

Algorithm 3 Multiplicative weight update method

```
Require: Time horizon $T \in \mathbb{N}$, the number $d \in \mathbb{N}$ of experts, learning rate $\eta > 0$.
1: Initialize the weight $w_1 = (w_{11}, w_{12}, \dots, w_{1d})^\top \in \mathbb{R}^d$ by $w_{1i} = 1$ for all $i \in [d]$.
2: for $t = 1, 2, \dots, T$ do
3:     Set the probabilistic strategy $p_t \in \Delta^d$ by $p_{ti} = w_{ti}/\|w_t\|_1$ for $i \in [d]$.
4:     Play $a_t = i$ with probability $p_{ti}$ and incur $f_t(a_t)$.
5:     Observe $f_t(1), f_t(2), \dots, f_t(d)$.
6:     Update $w_t$ by $w_{t+1,i} = w_{ti} \exp(-\eta f_t(i))$ for $i \in [d]$.
7: end for
```

2.4 Multiplicative Weight Update Method

In this section, we consider the *prediction from expert advice* problem. This problem is a simple special case of online optimization specified by $\mathcal{A} = [d]$, where d is a given integer parameter at least 2, and $\mathcal{F} = [0,1]^{\mathcal{A}} \cong [0,1]^d$.

The prediction from expert advice problem can be seen as a special case of online linear optimization, and hence, the online gradient descent method applies. Indeed, when we consider to manage a *probabilistic strategy* that is a probabilistic distribution p_t over $\mathcal{A} = [d]$, and choose an action a_t following p_t in each round, then the expected loss $\mathbf{E}[f_t(a_t)]$ is a linear function in p_t. Accordingly, we can apply Algorithm 2, the online gradient descent methods, and a sublinear regret bound can be achieved. The parameter D and G in Corollary 2.3.1 can be bounded as follows: The feasible region of p_t is congruent to $\Delta^d := \{x = (x_1, \dots, x_d)^\top \in [0,1]^d \mid x_1 + \dots + x_d = 1\}$ and $f_t \in [0,1]^d \in \mathcal{A}$. Since $\|p - p'\|_2 \le 1$ for all $p, p' \in \Delta^d$, we can assume $D = 1$. Similarly, since g_t in Algorithm 2 corresponds to the vector $(f_t(1), \dots, f_t(d))^\top \in [0,1]^d$, it holds that $\|g_t\|_2 \le \sqrt{d}$, and consequently, we can suppose $G = \sqrt{d}$. Hence, from Corollary 2.3.1, Algorithm 2 achieves the regret bound of $R_T(a^*) \le DG\sqrt{T} = \sqrt{dT}$.

For this problem, however, there is a better algorithm, the *multiplicative weight update* method, which is summarized in Algorithm 3. As shown from the following proposition, Algorithm 3 have a regret bound of $O(\sqrt{T \log d})$.

Proposition 2.4.1. *Suppose that $\eta f_t(i) \ge -1$ holds for all $t \in [T]$ and $i \in [d]$. The output of Algorithm 3 satisfies the following regret bound:*

$$\max_{i^* \in [d]} \mathbf{E}[R_T(i^*)] \le \frac{\log d}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{d} \mathbf{E}[p_{ti} f_t(i)^2]. \tag{2.13}$$

Proof. Fix $i^* \in [d]$ arbitrarily. The expectation of the regret $R_T(i^*)$ can be expressed as

$$\mathbf{E}[R_T(i^*)] = \mathbf{E}\left[\sum_{t=1}^{T} f_t(a_t) - \sum_{t=1}^{T} f_t(i^*)\right] = \mathbf{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{d} p_{ti} f_t(i) - \sum_{t=1}^{T} f_t(i^*)\right]. \tag{2.14}$$

Since $w_{T+1,i}$ can be expressed as $w_{T+1,i} = \exp(-\eta \sum_{t=1}^{T} f_t(i))$, the term $-\sum_{t=1}^{T} f_t(i^*)$ in (2.14) can be bounded as

$$-\sum_{t=1}^{T} f_t(i^*) = \frac{1}{\eta} \log w_{T+1,i^*} \le \frac{1}{\eta} \log \|w_{T+1}\|_1. \tag{2.15}$$

From the update rule $w_{t+1,i} = w_{ti} \exp(-\eta f_t(i))$, $\|w_{T+1}\|_1$ can be bounded as

$$\begin{aligned}
\|w_{T+1}\|_1 = \sum_{i=1}^d w_{T+1,i} = \sum_{i=1}^d w_{Ti} \exp(-\eta f_T(i)) \leq \sum_{i=1}^d w_{Ti}(1 - \eta f_T(i) + (\eta f_T(i))^2) \\
= \|w_T\|_1 \left(1 - \eta \sum_{i=1}^d p_{Ti} f_T(i) + \eta^2 \sum_{i=1}^d p_{Ti} f_T(i)^2\right) \\
\leq d \prod_{t=1}^T \left(1 - \eta \sum_{i=1}^d p_{ti} f_t(i) + \eta^2 \sum_{i=1}^d p_{ti} f_t(i)^2\right),
\end{aligned}$$

where the first inequality follows from $\exp(x) \leq 1 + x + x^2$ for $x \leq 1$ and the assumption of $\eta f_t(i) \geq -1$, the third equality follows from $p_{ti} = w_{ti}/\|w_t\|$, and the last inequality can be shown by applying the same operation recursively for $t = T-1, T-2, \ldots, 1$ and by substituting $\|w_t\|_1 = d$. Hence, we have

$$\begin{aligned}
\frac{1}{\eta} \log \|w_{T+1}\|_1 &\leq \frac{\log d}{\eta} + \frac{1}{\eta} \sum_{t=1}^T \log \left(1 - \eta \sum_{i=1}^d p_{ti} f_t(i) + \eta^2 \sum_{i=1}^d p_{ti} f_t(i)^2\right) \\
&\leq \frac{\log d}{\eta} - \sum_{t=1}^T \left(\sum_{i=1}^d p_{ti} f_j(i) - \eta \sum_{i=1}^d p_{ti} f_t(i)^2\right), \qquad (2.16)
\end{aligned}$$

where the last inequality follows from $\log(1 + x) \leq x$ for all $x > -1$. By combining (2.14), (2.15) and (2.16), we obtain

$$\mathbf{E}[R_T(i^*)] = \mathbf{E}\left[\sum_{t=1}^T \sum_{i=1}^d p_{ti} f_t(i) - \sum_{t=1}^T f_t(i^*)\right] \leq \frac{\log d}{\eta} + \eta \sum_{t=1}^T \sum_{i=1}^d \mathbf{E}[p_{ti} f_t(i)^2].$$

Since this holds for arbitrary $i^* \in [d]$, we have (2.13). □

Corollary 2.4.1. *Suppose that $f_t(i) \in [0, 1]$ holds for all $t \in [T]$ and for all $i \in [d]$. If we set $\eta = \sqrt{\frac{\log d}{T}}$, we have*

$$\max_{i^* \in [d]} \mathbf{E}[R_T(i^*)] \leq 2\sqrt{T \log d}. \qquad (2.17)$$

Proof. From $f_t(i) \leq 0$ and $\eta > 0$, the condition $\eta f_t(i) \leq -1$ follows, and consequently, Proposition 2.4.1 applies. From the condition $f_t(i) \in [0, 1]$, we have

$$\sum_{i=1}^d \mathbf{E}[p_{ti} f_t(i)^2] = \mathbf{E}\left[\sum_{i=1}^d p_{ti} f_t(i)^2\right] \leq \mathbf{E}\left[\sum_{i=1}^d p_{ti}\right] = 1.$$

From this and Proposition 2.4.1, we have

$$\max_{i^* \in [d]} \mathbf{E}[R_T(i^*)] \leq \frac{\log d}{\eta} + \eta T = 2\sqrt{T \log d},$$

where the equality follows from $\eta = \sqrt{\frac{\log d}{T}}$. □

Algorithm 4 Exp3 method for multi-armed bandit

```
Require: Time horizon $T \in \mathbb{N}$, the number $d \in \mathbb{N}$ of experts, learning rate $\eta > 0$.
1: Initialize the weight $w_1 = (w_{11}, w_{12}, \ldots, w_{1d})^\top \in \mathbb{R}^d$ by $w_{1i} = 1$ for all $i \in [d]$.
2: for $t = 1, 2, \ldots, T$ do
3:     Set the probabilistic strategy $p_t \in \Delta^d$ by $p_{ti} = w_{ti}/\|w_t\|_1$ for $i \in [d]$.
4:     Play $a_t = i$ with probability $p_{ti}$ and incur $f_t(a_t)$.
5:     Update $w_t$ by $w_{t+1,a_t} = w_{ti} \exp(-\eta f_t(a_t)/p_{ta_t})$ and $w_{t+1,i} = w_{t,i}$ for $i \in [d] \setminus \{a_t\}$.
6: end for
```

2.5 Non-Stochastic Multi-Armed Bandits

The *multi-armed bandit* problem is the sequential decision-making problem in which the player sequentially chooses an *arm* $a_t \in [d]$, which corresponds to the arm of slot machines, from $d \geq 2$ candidates, and observes the loss $f_t(a_t)$ for the chosen arm, in each round t. This can be seen as a variant of the prediction from experts problem. A major difference between these two problems is the feedback information; the prediction from experts problem allows the player to observe all of $f_t(1), f_t(2), \ldots, f_t(d)$ while one can observe $f_t(a_t)$ in the multi-armed bandit problem.

The *Exp3* method [16] is an algorithm for non-stochastic multi-armed bandit problems based on the multiplicative weight update method. In Exp3 method, we construct *unbiased estimators* of $f_t(i)$ for all $i \in [d]$ from the bandit feedback $f_t(a_t)$, and apply MWU with these unbiased estimators $\hat{f}_t(i)$ instead of $f_t(i)$. The values of $\hat{f}_t$ are given as follows: Suppose that a_t is chosen randomly so that $\mathrm{Prob}[a_t = i] = p_{ti}$. We then define $\hat{f}_t$ by

$$\hat{f}_t(i) = \begin{cases} \frac{1}{p_{ti}} f_t(i) & (i = a_t) \\ 0 & (i \in [d] \setminus \{a_t\}) \end{cases}. \tag{2.18}$$

The expectation of the function values of $\hat{f}_t$ is equal to f_t because we have

$$\mathbf{E}[\hat{f}_t(i)] = \mathrm{Prob}[a_t = i] \cdot \frac{1}{p_{ti}} f_t(i) = f_t(i)$$

for all $i \in [d]$. The algorithm can be summarized as Algorihm 4.

From Proposition 2.4.1 and the definition (2.18) of $\hat{f}_t$, Algorithm 4 has the following regret bound:

Proposition 2.5.1. *Suppose that $f_t(i) \in [0, 1]$ holds for all $t \in [T]$ and for all $i \in [d]$. If we set $\eta = \sqrt{\log d/(dT)}$, for the multi-armed bandit problem, Algorithm 4 has the following regret bound:*

$$\max_{i^* \in [d]} \mathbf{E}[R_T(i^*)] \leq 2\sqrt{Td \log d}. \tag{2.19}$$

Proof. Fix $i^* \in [d]$ arbitrarily. The expected regret $\mathbf{E}[R_T(i^*)]$ can be expressed as

$$\mathbf{E}[R_T(i^*)] = \mathbf{E}\left[\sum_{t=1}^T f_t(a_t) - \sum_{t=1}^T f_t(i^*)\right] = \mathbf{E}\left[\sum_{t=1}^T \sum_{i=1}^d p_{ti} \hat{f}_t(i) - \sum_{t=1}^T \hat{f}_t(i^*)\right], \tag{2.20}$$

where the second equality follows from the definition (2.18) of $\hat{f}_t$. Since $\hat{f}_t(i) \geq 0$ holds, the analysis in Proposition 2.4.1 applies, and consequently, the right-hand side of (2.20) can be bounded as

$$\begin{aligned}
&\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{d} p_{ti}\hat{f}_t(i) - \sum_{t=1}^{T}\hat{f}_t(i^*)\right] \leq \frac{\log d}{\eta} + \eta\sum_{t=1}^{T}\sum_{i=1}^{d}\mathbf{E}[p_{ti}\hat{f}_t(i)^2] = \frac{\log d}{\eta} + \eta\sum_{t=1}^{T}\mathbf{E}\left[\frac{f_t(a_t)^2}{p_{ta_t}}\right] \\
&= \frac{\log d}{\eta} + \eta\sum_{t=1}^{T}\mathbf{E}\left[\sum_{i=1}^{d}\mathrm{Prob}[a_t = i]\cdot\frac{f_t(i)^2}{p_{ti}}\right] = \frac{\log d}{\eta} + \eta\sum_{t=1}^{T}\mathbf{E}\left[\sum_{i=1}^{d} f_t(i)^2\right] \\
&\leq \frac{\log d}{\eta} + \eta dT = 2\sqrt{Td\log d}, \qquad (2.21)
\end{aligned}$$

where the first inequality follows from the proof of Proposition 2.4.1, the first equality follows from (2.18), the last inequality follows from the assumption of $f_t(i) \in [0,1]$, and the last equality follows from $\eta = \sqrt{\log d/(dT)}$. Combining (2.20) and (2.21), we obtain $\mathbf{E}[R_T(i^*)] \leq 2\sqrt{Td\log d}$. Since this holds for all $i^* \in [d]$, we have (2.19). □

Chapter 3

Algorithms for Online Linear Optimization with Bandit Feedback

We propose computationally efficient algorithms for *online linear optimization with bandit feedback*, in which a player chooses an *action vector* from a given (possibly infinite) set $\mathcal{A} \subseteq \mathbb{R}^d$, and then suffers a loss that can be expressed as a linear function in action vectors. Though there exist algorithms that achieve an optimal regret bound of $\tilde{O}(\sqrt{T})$ for T rounds (ignoring factors of $\mathrm{poly}(d, \log T)$), computationally efficient ways of implementing them have not yet been made clear, in particular when $|\mathcal{A}|$ is not bounded by a polynomial size in d. One standard way to pursue computational efficiency is to assume that we have an efficient algorithm referred to as *oracle* that solves (offline) linear optimization problems over $\mathcal{A}$. Under this assumption, the computational efficiency of a bandit algorithm can then be measured in terms of *oracle complexity*, i.e., the number of oracle calls. Our contribution is to propose algorithms that offer optimal regret bounds of $\tilde{O}(\sqrt{T})$ as well as low oracle complexity for both *non-stochastic settings* and *stochastic settings*. Our algorithm for non-stochastic settings has an oracle complexity of $\tilde{O}(T)$ and is the first algorithm that achieves both a regret bound of $\tilde{O}(\sqrt{T})$ and an oracle complexity of $\tilde{O}(\mathrm{poly}(T))$, given only linear optimization oracles. Our algorithm for stochastic settings calls the oracle only $O(\mathrm{poly}(d, \log T))$ times, which is smaller than the current best oracle complexity of $O(T)$ if T is sufficiently large.

3.1 Introduction

Online linear optimization with bandit feedback, or *bandit linear optimization*, is an important problem that has a wide range of applications. In it, a player is given $\mathcal{A} \subseteq \mathbb{R}^d$, referred to as a set of *action vectors*, and the number T of *rounds* of decision-making. In each round $t \in [T] := \{1, 2, \ldots, T\}$, the player chooses an action $a_t \in \mathcal{A}$, and then observes loss $\ell_t^\top a_t$, where $\ell_t \in \mathbb{R}^d$ is an unknown *loss vector* that can change over rounds. The bandit linear optimization includes a variety of important online decision-making problems as special cases. For example, given a graph $G = (V, E)$ and $s, t \in V$, by setting $\mathcal{A} \subseteq \mathbb{R}^{|E|}$ to be the set of all characteristic vectors of s-t paths, we can take into account *bandit shortest path* or *adaptive routing* [17]. In this setting, $\ell_t \in \mathbb{R}^{|E|}$ corresponds to (unknown) lengths of the edges, and bandit feedback $\ell_t^\top a_t$ represents the length of a chosen s-t path a_t. In addition to this application, bandit linear optimization includes bandit versions of such combinatorial optimization problems as minimum spanning tree, minimum cut, and knapsack problem, as well as continuous optimization problems such as linear programming and semidefinite programming.

The performance of the player is evaluated in terms of *regret* $R_T(a^*)$, defined as $R_T(a^*) = \sum_{t=1}^T \ell_t^\top a_t - \sum_{t=1}^T \ell_t^\top a^*$ for $a^* \in \mathcal{A}$, which represents the difference between the cumulative loss for the decision $\{a_t\}$ of the player and that for an arbitrarily fixed strategy a^*. The primary goal in bandit linear optimization is to achieve small regret for arbitrary $a^* \in \mathcal{A}$. It is known that there exist algorithms achieving regret bounds of $\tilde{O}(\sqrt{T})$,[1] as shown in Tables 3.1 and 3.2. In contrast, papers [12, 16, 44] showed that any algorithm will suffer at least $\Omega(\sqrt{T})$ regret in the worst case. Thus, algorithms with $\tilde{O}(\sqrt{T})$-regret bounds achieve *optimal* performance w.r.t. dependence on T.

Algorithms achieving an optimal $\tilde{O}(\sqrt{T})$-regret, however, have computational issues, especially when the action set $\mathcal{A}$ is exponentially large or is an infinite set. For example, well-known LinUCB methods [1, 51, 146] need to solve quadratic programming over $\mathcal{A}$, which costs time complexity of $\Omega(|\mathcal{A}|)$ if there are no additional assumptions. The ComBand algorithm [38] runs efficiently if there is an efficient sampling algorithm for $\mathcal{A}$ (such as k-sets, spanning trees, or bipartite perfect matchings), but such sampling algorithms are open for many important examples, including s-t paths. For the special case in which the convex hull of $\mathcal{A}$ can be expressed by c linear inequalities, CombExp [45] runs in $O(\mathrm{poly}(c, d)T)$-time. However, the size c of the linear inequality expression can be exponentially large for many examples.

This chapter aims to develop computationally efficient algorithms that achieve an $\tilde{O}(\sqrt{T})$ regret bound, under the assumption that we can call a *linear optimization oracle.* The oracle solves offline linear optimization problems over $\mathcal{A}$, i.e, given a loss vector $\ell \in \mathbb{R}^d$, the oracle outputs $a^* \in \arg\min_{a \in \mathcal{A}} \ell^\top a$. This assumption is standard in the context of online optimization [50, 100]. Under it, the computational efficiency of online optimization algorithms is evaluated in terms of *oracle complexity*, the number of oracle calls for the linear optimization oracle.

For online linear optimization with *full information*, in which a player can observe all entries of $\ell_t \in \mathbb{R}^d$ after choosing a_t, Kalai and Vempala [100] have proposed algorithms with an $\tilde{O}(\sqrt{T})$-regret bound and an oracle complexity of $O(T)$. By means of this algorithm, McMahan and Blum [131] and Dani and Hayes [50] showed that one can achieve $\tilde{O}(T^{2/3})$-regret and $O(T^{1/2})$-oracle complexity for bandit linear optimization. However, it has been an open question as to whether or not we can achieve $\tilde{O}(\sqrt{T})$-regret and $\tilde{O}(\mathrm{poly}(T))$-oracle complexity for bandit linear optimization, with only linear optimization oracles. We solve this open problem by proposing an algorithm that achieves $\tilde{O}(\sqrt{T})$-regret as well as $\tilde{O}(T)$-oracle complexity.

We separately consider here two different settings for bandit linear optimization; a *non-stochastic setting* and a *stochastic setting.* In the non-stochastic setting, we do not assume any generative models, but ℓ_t may be chosen in an adversarial manner, depending on the previous actions $a_1, \ldots, a_{t-1}$. The performance of an algorithm is measured in terms of the expectation of regret $R_T(a^*)$ w.r.t. the randomness of the algorithm's internal randomness and ℓ_t. In the stochastic setting, by way of contrast, the loss vectors ℓ_t are assumed to follow a probability distribution D over $\mathbb{R}^d$, i.i.d. for $t = 1, \ldots, T$.

In this chapter, we present computationally efficient algorithms achieving $O(\mathrm{poly}(d)\sqrt{T})$-regret. Specifically, we present algorithms with a small oracle complexity, i.e., algorithms that call the oracle as infrequently as possible. Our contribution is summarized in Tables 3.1 and

[1] In $\tilde{O}(\cdot)$ notation, we ignore factors of polynomials in d and $\log(T)$.

3.2.

For the non-stochastic setting, we propose an algorithm (Algorithm 5) that achieves a regret upper bound of $O(\sqrt{d^3 T \log T})$ in expectation and has $O(\mathrm{poly}(d, \log T)T)$-oracle complexity.

Theorem 3.1.1. *For the non-stochastic setting, Algorithm 5 satisfies the following:*

- *The output of the algorithm satisfies* $\mathbf{E}[R_T(a^*)] = O(\sqrt{d^3 T \log T})$ *for all* $a^* \in \mathcal{A}$.
- *The algorithm calls the linear optimization oracle* $O(\mathrm{poly}(d, \log T)T)$ *times.*
- *The computational time, except for that of the oracle, is of* $O(\mathrm{poly}(d, T))$.

As shown in Table 3.1, our Algorithm 5 achieves the smallest oracle complexity among algorithms with $\tilde{O}(\sqrt{T})$-regret. Noting that GeometricHedge assumes $\mathcal{A}$ to be a convex body, we can see that Algorithm 5 is the first algorithm that is applicable to discrete $\mathcal{A}$ and that achieves $\tilde{O}(\sqrt{T})$-regret and $\tilde{O}(\mathrm{poly}(T))$-oracle complexity.

Though the first algorithm in Table 3.1 with $\tilde{O}(T^{2/3})$-regret and $O(T^{2/3})$-oracle complexity might look incomparable to our results, algorithms with such bounds can be easily constructed from our Algorithm 5. In fact, by dividing T rounds into T/B blocks of size $B > 1$ and regarding each block as an individual round, we obtain the following statement:

Proposition 3.1.1. *Suppose there exists an algorithm with* $O(f(T))$*-regret and* $O(g(T))$*-oracle complexity. Then for arbitrary positive integer* B*, there exists an algorithm with* $O(B \cdot f(T/B))$*-regret and* $O(g(T/B))$*-oracle complexity.*

By setting the block size to be $B = \Theta(T^{1/3})$ and applying Algorithm 5, we can achieve $O(B\sqrt{T/B}) = \tilde{O}(T^{2/3})$-regret and $O(T/B) = \tilde{O}(T^{2/3})$-oracle complexity, which is equivalent to the performance in the uppermost result in Table 5. Note that Proposition 3.1.1 does *not* lead to an $\tilde{O}(\sqrt{T})$-regret algorithm given an $\tilde{O}(T^{2/3})$-regret algorithm conversely since the block size B must be at least 1.

For the stochastic setting, we propose an algorithm (Algorithm 6) that achieves a regret bound of $O(\sqrt{d^3 T \log(d \log T/\delta)})$ with probability $1 - \delta$ and has $O(\mathrm{poly}(d, \log T))$-oracle complexity, where $\delta \in (0, 1)$ is an arbitrary parameter.

Theorem 3.1.2. *Suppose* ℓ_t *follows a distribution over* $\mathbb{R}^d$*, i.i.d. for* $t = 1, 2, \ldots, T$*. Algorithm 6 then satisfies the following:*

- *The output of the algorithm satisfies* $R_T(a^*) = O(\sqrt{d^3 T \log(d \log T/\delta)})$ *for all* $a^* \in \mathcal{A}$*, with probability* $1 - \delta$.
- *The algorithm calls the linear optimization oracle* $O(\mathrm{poly}(d, \log T))$ *times.*
- *The computational time, except for that of the oracle, is of* $O(\mathrm{poly}(d, T))$.

A complete description of Algorithm 6 and a proof of this theorem are given in Section 3.5. As shown in Table 3.2, all existing algorithms achieving $\tilde{O}(\sqrt{T})$-regret require at least $\Omega(T)$ oracle complexity, and our Algorithm 6 is the first with an $\tilde{O}(\sqrt{T})$-regret bound and a sublinear oracle complexity in T.

[2]In this algorithm, $\mathcal{A}$ is assumed to be a convex body, and a membership oracle for $\mathcal{A}$ is assumed. Since we can construct a membership oracle from a linear optimization oracle and vice versa by a polynomial time reduction [150], the assumption regarding the oracle is equivalent to ours, modulo polynomial time reduction.

Table 3.1: Regret Bound and Oracle Complexity of for Non-Stochastic Bandit Linear Optimization

Algorithm	Regret Bound	Oracle Complexity
MV algorithm [50, 131] with FPL [100]	$\tilde{O}(T^{2/3})$	$O(T^{2/3})$
ComBand [38] GeometricHedge [52], Exp2 [13]	$\tilde{O}(T^{1/2})$	–
GeometricHedge with Volumetric Spanners[2][79]	$\tilde{O}(T^{1/2})$	$\tilde{O}(T^7)$
Algorithm 5 [This work]	$\tilde{O}(T^{1/2})$	$\tilde{O}(T)$

Table 3.2: Regret Bound and Oracle Complexity for Stochastic Bandit Linear Optimization

Algorithm	Regret Bound	Oracle Complexity
LinRel [14], LinUCB with ℓ_2-ball [1, 51, 146]	$\tilde{O}(T^{1/2})$	–
LinUCB with ℓ_1-ball [51],	$\tilde{O}(T^{1/2})$	$O(T)$
Linear Thompson sampling [3, 8]	$\tilde{O}(T^{1/2})$	$O(T)$
Algorithm 6 [This work]	$\tilde{O}(T^{1/2})$	$O(\mathrm{poly}(d, \log T))$

In both of Algorithms 5 and 6, we use the well-known techniques [150] of reduction among linear optimization, separation, and decomposition over a given convex body. Definitions of these three problems are given in Section 3.3. The reduction algorithms enable us to solve separation and decomposition problems by calling the linear optimization oracle $O(\mathrm{poly}(d))$ times. By means of these reduction techniques, Algorithms 5 and 6 maintain, respectively, supersets and subsets of the convex hull of $\mathcal{A}$ ($=: \mathrm{Conv}(\mathcal{A})$).

To construct Algorithm 5 for the non-stochastic setting, we extend a *cutting-plane approach* to our bandit-feedback setting. The cutting-plane approach, a way of reducing oracle complexity, has been applied only to full-information settings [82], not a bandit-feedback setting. A major difference between bandit-feedback and full-information settings is that the former needs *exploration*, i.e., chosen actions should be randomized with sufficiently large variance, while the latter does not need it and chooses actions deterministically. In full-information settings, hence, it suffices to focus on a deterministically chosen action alone. In contrast to this, to deal with the bandit-feedback setting, the difficulty lies in constructing a distribution of actions with a sufficiently large variance, for which cutting planes can be efficiently computed and the number of them can be bounded.

To this end, we design relevant probability distributions so that the cutting-plane approach works, which successfully reduces oracle complexity. More precisely, the cutting-plane approach maintains convex bodies $\mathcal{K}_t$ that include and approximate $\mathrm{Conv}(\mathcal{A})$, from which we choose *candidates* for actions, employing the support of the probability distribution of actions to choose. It is only when some candidates are invalid, i.e., when some are outside of $\mathrm{Conv}(\mathcal{A})$, that $\mathcal{K}_t$ is updated with a cutting plane excluding such an invalid candidate. In order to bound the number of oracle calls, we design candidates for actions that satisfy two conditions: the set of candidates has a bounded cardinality, and each candidate is sufficiently close to the weighted center of $\mathcal{K}_t$. Thanks to the first condition, we are able to efficiently decide if there exist invalid candidates. The second condition is essential for bounding the number of oracle calls in each update of $\mathcal{K}_t$.

Our Algorithm 6 for the stochastic setting is based on the framework of *phased elimination* *of actions*, in which T rounds are divided into *phases* that are segments of consequent rounds, and, in each phase, actions are eliminated so that only promising ones are left. Existing works employing this framework [14, 118, 161], are computationally inefficient, mainly due to the following two reasons: (i) We need to maintain a set of promising actions that may be an exponentially large combinatorial set, and, (ii) when choosing actions, we need to solve hard optimization problems, e.g., G-optimal design [118] or quadratic programming [14].

Our idea for resolving the first computational issue is to maintain the set of promising actions as a convex set instead of a subset of actions. The convex set here can be represented with only $O(\mathrm{poly}(d) \log T)$ linear inequalities, which implies that operations over it can be conducted efficiently. We resolve the second computational issue by combining *barycentric spanners* [17] and the decomposition technique over convex bodies [150], both of which are efficiently computed with $O(\mathrm{poly}(d))$ oracle calls. We show that thanks to these techniques, we can estimate the loss vector with enough accuracy to achieve an $\tilde{O}(\sqrt{T})$-regret bound. The oracle complexity is bounded as follows: In our algorithm, all a_t chosen in each phase are determined at the beginning of the phase, which means that the oracle complexity depends not on the number of rounds, but on the number of phases. The number of phases is of $O(\mathrm{poly}(d) \log T)$ and that of oracle calls in each phase is of $O(\mathrm{poly}(d, \log T))$, which results in overall $O(\mathrm{poly}(d, \log T))$-oracle complexity.

3.2 Problem Setting

The *bandit linear optimization problem* is a repeated game described as follows: Before the game starts, a player is given the number T of rounds and the dimensionality d of the *action set* $\mathcal{A} \subseteq \mathbb{R}^d$. In each round $t = 1, 2, \ldots, T$, the player chooses $a_t \in \mathcal{A}$ while an environment chooses a loss vector $\ell_t \in \mathbb{R}^d$, and then the player observes a loss $\ell_t^\top a_t$. The goal of the player is to achieve a small *regret* $R_T(a)$, which is defined for an arbitrary $a \in \mathcal{A}$ as $R_T(a) := \sum_{t=1}^T \ell_t^\top a_t - \sum_{t=1}^T \ell_t^\top a$.

We assume the action set $\mathcal{A}$ to be a compact set. Suppose that we have an algorithm for linear optimization over $\mathcal{A}$ for any vector $w \in \mathbb{R}^d$, which we call *linear optimization oracle* $\mathcal{O}_\mathcal{A} : \mathbb{R}^d \to \mathcal{A}$ that receives an input $w \in \mathbb{R}^d$ and returns a point $\mathcal{O}_\mathcal{A}(w) \in \mathcal{K}$ satisfying $w^\top \mathcal{O}_\mathcal{A}(w) = \min_{a \in \mathcal{A}} w^\top a$.

Assumption 3.2.1. We assume that there exist positive real numbers L and R such that: (a) $\|\ell_t\|_2 \leq L$ for all $t \in [T]$, and (b) $\|a\|_2 \leq R$ for all $a \in \mathcal{A}$. In addition, we assume that (c) $\mathcal{K} := \mathrm{Conv}(\mathcal{A})$ has a positive volume, i.e., $\mathrm{Vol}(\mathcal{K}) := \int_\mathcal{K} 1 \mathrm{d}x > 0$.

The first two assumptions (a) and (b) are standard in bandit linear optimization. If we are given a linear optimization oracle over $\mathcal{A}$, we can assume (c) without loss of generality. In fact, if $\mathcal{A}$ is included in a subspace with a smaller dimension than d, we can then detect it by calling the linear optimization oracle polynomial times (see, e.g., Corollary 14.1g in [150]), and we can make $\mathcal{K}$ full-dimensional by ignoring redundant dimensions.

3.3 Preliminary

3.3.1 Linear Optimization, Separation, and Decomposition

We define a *linear optimization problem* (LP), *separation problem* (SP), and *decomposition problem* (DP) for a compact convex body $\mathcal{P} \subseteq \mathbb{R}^d$ as follows:

Problem 3.3.1 (linear optimization problem, LP). Given a vector $w \in \mathbb{R}^d$, find a vector $x^* \in \mathcal{P}$ such that $w^\top x^* = \min_{x \in \mathcal{P}} w^\top x$.

Problem 3.3.2 (separation problem, SP). Given a vector $y \in \mathbb{R}^d$, decide whether y belong to $\mathcal{P}$ or not, and, in the latter case, find a vector $w \in \mathbb{R}^d$ such that $w^\top y < \min_{x \in \mathcal{P}} w^\top x$.

Problem 3.3.3 (decomposition problem, DP). Given a vector $x \in \mathcal{P}$, find vertices $x_0, \ldots, x_d$ of $\mathcal{P}$ and $\lambda_0, \ldots, \lambda_d \geq 0$ such that $x = \lambda_0 x_0 + \cdots + \lambda_d x_d$.

Ellipsoid methods provide reductions among these problems, which imply that

$$\text{LP: solvable} \iff \text{SP: solvable} \implies \text{DP: solvable}.$$

Theorem 3.3.1 (**Corollaries 14.1a**, **14.1b** and **14.1g** in [150])**.** *Suppose that $\mathcal{P} \subseteq \mathbb{R}^d$ is a polytope of which each vertex can be expressed by rationals with bit-lengths of at most φ, and that each entry of $x, y, w \in \mathbb{Q}^d$ is also a rational, with bit-length of at most φ. We then have the following:*

(a) *If there is an algorithm* SEP *that solves the separation problem, we can solve the linear optimization problem for $w \in \mathbb{Q}^d$ by calling* SEP *at most* $\mathrm{poly}(d, \varphi)$ *times.*

(b) *If there is an algorithm* OPT *that solves the linear optimization problem, we can solve the separation problem for $y \in \mathbb{Q}^d$ by calling* OPT *at most* $\mathrm{poly}(d, \varphi)$ *times.*

(c) *If there is an algorithm* OPT *that solves the linear optimization problem, we can solve the decomposition problem for $x \in \mathcal{P}$ by calling* OPT *at most* $\mathrm{poly}(d, \varphi)$ *times.*

Remark 3.3.1. For any $\varepsilon > 0$ and any real number $x \in [-1, 1]$, we can approximate x by a rational $\hat{x} \in \mathbb{Q}$ with a bit-length of at most $O(\log(1/\varepsilon))$ so that $|x - \hat{x}| \leq \varepsilon$. Hence, we can assume that φ in Theorem 3.3.1 is bounded as $\varphi = O(\log T)$ by ignoring $O(1/T)$ errors. This implies that the above reductions can be computed in $O(\mathrm{poly}(d, \log T))$ time.

3.3.2 Algorithms for Logconcave Distributions

If a probability distribution over convex body $\mathcal{P} \subseteq \mathbb{R}^d$ has a probability density function (PDF) $p : \mathcal{P} \to \mathbb{R}_{>0}$ such that $\log p$ is a concave function, we refer to it as a *logconcave distribution.* The following theorem means that, given the value oracle of a convex function $f : \mathcal{P} \to \mathbb{R}$, we can approximately sample a vector in $\mathcal{P}$ from a logconcave distribution $p(x) \propto \exp(-f(x))$.

Theorem 3.3.2 (Theorems 2.1 and 2.2 in [127], Lemma 10 in [79])**.** *Let $\mathcal{P} \subseteq \mathbb{R}^d$ be a convex body with non-zero Lubesgue measure, and let $f : \mathcal{P} \to \mathbb{R}$ be a convex function and let p be a logconcave distribution proportional to* $\exp(-f(x))$. *Suppose* $\varepsilon > 0$ *and* $\delta \in (0, 1)$. *Then, given access to the membership oracle of $\mathcal{P}$ and the value oracle of f, there is an algorithm which samples approximately from p such that: (i) the total variation distance between the produced distribution and p is at most ε, and (ii) after pre-processing for a time of $O(d^5 (\log d)^{O(1)})$, each sample can be produced in a time of $O(d^4/\varepsilon^4 \cdot (\log(d/\varepsilon))^{O(1)})$.*

As an implication of this theorem, we can efficiently approximate the mean $\mu(p) \in \mathbb{R}^d$ and the covariance matrix $\mathrm{Cov}(p) \in \mathbb{R}^{d \times d}$ of the distribution p. In fact, from Corollary 5.52 in [162] and standard concentration of logcancave distribution (see, e.g., Lemma 5.17 in [127]), it takes $(n \log(1/\delta)/\varepsilon)^{O(1)}$ samples to get a matrix $\hat{\Sigma}$ such that $(1 - \varepsilon)\mathrm{Cov}(p) \preceq \hat{\Sigma} \preceq (1 + \varepsilon)\mathrm{Cov}(p)$ with probability of at least $1-\delta$.[3] Similarly, we can get $\hat{\mu} \in \mathbb{R}^d$ such that $\|\hat{\mu} - \mu(p)\|_{\mathrm{Cov}(p)^{-1}} \leq \varepsilon$ from $(n \log(1/\delta)/\varepsilon)^{O(1)}$ samples.[4] Accordingly, we obtain the following corollary:

Corollary 3.3.1. *Suppose the same assumption as in Theorem 3.3.2. There is an algorithm that outputs a vector $\hat{\mu} \in \mathbb{R}^d$ and a symmetric matrix $\hat{\Sigma} \in \mathbb{R}^{d \times d}$ satisfying $\frac{1}{2}\mathrm{Cov}(p) \preceq \hat{\Sigma} \preceq 2\mathrm{Cov}(p)$ and $\|\hat{\mu} - \mu(p)\|_{\mathrm{Cov}(p)^{-1}} \leq \varepsilon$ with a probability of at least $1-\delta$. The computational time, the number of calls for the membership oracle of $\mathcal{P}$, and the value oracle of f are bounded by* $\mathrm{poly}(d, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$.

3.3.3 Barycentric Spanner

Definition 3.3.1. Let $S \in \mathbb{R}^d$ be a subset whose linear span is $\mathbb{R}^d$, and let $C > 1$. A set $X = \{x_1, \ldots, x_d\} \subseteq S$ is a *C-barycentric spanner* for S if every $x \in S$ may be expressed as a linear combination of elements of X using coefficients in $[-C, C]$.

Theorem 3.3.3 (**Proposition 2.4.** in [17])**.** *Suppose $\mathcal{P} \subseteq \mathbb{R}^d$ is a compact set not contained in any proper linear subspace. Given an algorithm OPT for LP, for any $C > 1$ we may compute a C-barycentric spanner for $\mathcal{P}$ in polynomial time, using $O(d^2 \log_C(d))$ calls to OPT. Its span is equal to $\mathbb{R}^d$.*

3.4 Algorithm for Non-stochastic Bandit Linear Optimization

Our algorithm uses the framework of a continuous multiplicative weight update (CMWU) [11, 47, 163]. A straightforward way of applying CMWU is to maintain probability distributions over $\mathcal{K} := \mathrm{Conv}(\mathcal{A})$, which, however, requires a large number of oracle calls. In fact, the algorithm by Hazan and Karnin [79] for bandit linear optimization over convex bodies calls an oracle $\tilde{O}(T^7)$ times. This inefficiency is due to that we need to sample from $\mathcal{K}$; the sampling algorithm in Theorem 3.3.2 requires $O(d^4/\varepsilon^4)$-oracle complexity.

We reduce oracle complexity by means of a *cutting-plane approach* [82]. In this approach, we maintain convex bodies $\mathcal{K}_t^{(j)}$ that include and approximate $\mathcal{K}$, and we update a distribution over $\mathcal{K}_t^{(j)}$ instead of $\mathcal{K}$. The advantage of this approach is that we can sample from $\mathcal{K}_t^{(j)}$ without calling an oracle. On the other hand, updating $\mathcal{K}_t^{(j)}$ requires oracle calls, and therefore, we need to bound the number of the updates as well as the number of oracle calls in each update. We design a strategy achieving these as follows: We set *candidates of actions* $\mathcal{E}_t^{(j)} \subseteq \mathcal{K}_t^{(j)}$, from which we choose action. When some actions among the candidates are invalid, i.e., outside of $\mathcal{K}$, we then reduce $\mathcal{K}_t^{(j)}$ by a cutting plane excluding such an invalid candidate. With this strategy, we need oracle calls to check if there exist invalid candidates. Our algorithm bounds the oracle complexity here by setting $\mathcal{E}_t^{(j)}$ to have $O(d)$ elements. Further, we design $\mathcal{E}_t^{(j)}$ so that its elements are sufficiently close to the weighted center of $\mathcal{K}_t^{(j)}$. This plays an important role in bounding the number of updates of $\mathcal{K}_t^{(j)}$. Indeed, when a convex body is updated by a cutting plane that excludes a point close to its center, its volume then decreases by a constant factor less than 1 (see, e.g., [127]). On the other hand, $\mathcal{K}_t^{(j)}$ always includes $\mathcal{K}$ with a positive volume, and hence, the volume of $\mathcal{K}_t^{(j)}$ cannot be smaller than that of $\mathcal{K}$, which implies that the number of updates is bounded.

[3] A similar argument can be found in Section 6.3 in [34].

[4] For a vector $x \in \mathbb{R}^d$ and a positive semidefinite matrix $A \in \mathbb{R}^{d \times d}$, denote $\|x\|_A := \sqrt{x^\top A x}$.

3.4.1 Algorithm

Our algorithm maintains a convex body $\mathcal{K}_t^{(j)} \subseteq \mathbb{R}^d$ such that $\mathcal{K}_1^{(0)} \supseteq \mathcal{K}_1^{(1)} \supseteq \cdots \supseteq \mathcal{K}_1^{(s_1)} = \mathcal{K}_2^{(0)} \supseteq \mathcal{K}_2^{(1)} \supseteq \cdots \supseteq \mathcal{K}_2^{(s_2)} \supseteq \cdots \supseteq \mathcal{K}_T^{(s_T)} \supseteq \mathcal{K} = \mathrm{Conv}(\mathcal{A})$, where t corresponds to the round, $j \in \{0, 1, \ldots\}$ is an index, and $s_t \in \{0, 1, \ldots, T\}$ is as will be defined later. It also updates a logconcave function $z_t : \mathbb{R}^d \to \mathbb{R}_{>0}$ in each round t based on the multiplicative weight update [11, 47, 163]. Before the first round, z_t is initialized to be a constant function $z_1(x) = 1$. Let $q_t^{(j)}$ denote the PDF of a distribution over $\mathcal{K}_t^{(j)}$ that is proportional to the function z_t, i.e.,

$$Z_t^{(j)} = \int_{\mathcal{K}_t^{(j)}} z_t(x) \mathrm{d}x, \quad q_t^{(j)}(x) = \begin{cases} \frac{z_t(x)}{Z_t^{(j)}} & \text{if } a \in \mathcal{K}_t^{(j)}, \\ 0 & \text{if } a \in \mathbb{R}^d \setminus \mathcal{K}_t^{(j)}. \end{cases} \tag{3.1}$$

Let us denote the mean and the covariance matrix for distribution of $q_t^{(j)}$ by $\mu_t^{(j)} \in \mathbb{R}^d$ and $\Sigma_t^{(j)} \in \mathbb{R}^{d \times d}$, respectively: $\mu_t^{(j)} = \mathbf{E}_{a \sim q_t^{(j)}}[a]$, $\Sigma_t^{(j)} = \mathbf{E}_{a \sim q_t^{(j)}}[(a - \mu_t^{(j)})(a - \mu_t^{(j)})^\top]$. From Corollary 3.3.1, we can compute estimators $\hat{\mu}_t^{(j)}$ and $\hat{\Sigma}_t^{(j)}$ of $\mu_t^{(j)}$ and $\Sigma_t^{(j)}$, respectively, such that

$$\frac{1}{2}\Sigma_t^{(j)} \preceq \hat{\Sigma}_t^{(j)} \preceq 2\Sigma_t^{(j)}, \quad \|\hat{\mu}_t^{(j)} - \mu_t^{(j)}\|_{(\Sigma_t^{(j)})^{-1}} \leq \varepsilon \tag{3.2}$$

with probability of at least $1 - \delta_t^{(j)}$, where $\varepsilon > 0$ and $\delta_t^{(j)} \in (0, 1)$, which will be defined later. Let $B_t^{(j)} = (b_{t1}^{(j)}, \ldots, b_{td}^{(j)}) \in \mathbb{R}^{d \times d}$ be a matrix such that $B_t^{(j)} B_t^{(j)\top} = \hat{\Sigma}_d^{(j)}$. Define $\mathcal{E}_t^{(j)} \subseteq \mathbb{R}^d$ as

$$\mathcal{E}_t^{(j)} = \left\{ \hat{\mu}_t^{(j)} + \frac{1}{4\mathrm{e}} b_{ti}^{(j)} \;\middle|\; i \in [d] \right\} \cup \left\{ \hat{\mu}_t^{(j)} - \frac{1}{4\mathrm{e}} b_{ti}^{(j)} \;\middle|\; i \in [d] \right\}. \tag{3.3}$$

In each round t, our algorithm checks if $\mathcal{E}_t^{(j)}$ is included in $\mathcal{K}$, and if not, it updates $\mathcal{K}_t^{(j)}$, as described in Step 7 of Algorithm 5, to exclude an element in $\mathcal{E}_t^{(j)} \setminus \mathcal{K}$. The set $\mathcal{E}_t^{(j)}$ is designed so that the following four conditions are satisfied:

1. The cardinality of $\mathcal{E}_t^{(j)}$ is bounded as $|\mathcal{E}_t^{(j)}| = O(d)$. Hence, we can decide if $\mathcal{E}_t^{(j)} \subseteq \mathcal{K}$ by $O(\mathrm{poly}(d))$ oracle calls.

2. Each $y \in \mathcal{E}_t^{(j)}$ is sufficiently close to $\mu_t^{(j)}$, i.e., it satisfies $\|y - \mu_t^{(j)}\|_{(\Sigma_t^{(j)})^{-1}} \leq 1/(2\mathrm{e})$. This is important to bound the number of oracle calls.

3. The mean of $\mathcal{E}_t^{(j)}$ is equal to $\hat{\mu}_t^{(j)}$. This implies that if y follows a uniform distribution over $\mathcal{E}_t^{(j)}$, we then have $\mathbf{E}[\ell_t^\top y] = \ell_t^\top \hat{\mu}_t^{(j)} \approx \ell_t^\top \mu_t^{(j)} = \mathbf{E}[\ell_t^\top x]$ for $x \sim q_t^{(j)}$.

4. The covariance matrix Σ of a uniform distribution over $\mathcal{E}_t^{(j)}$ satisfies $\Sigma \succeq O(1/d^2) \cdot \Sigma_t^{(j)}$. Thanks to this, empirical estimates of ℓ_t based on $\mathcal{E}_t^{(j)}$ will have a sufficiently small variance.

The conditions 1 and 2 are used to bound the oracle complexity, and 3 and 4 are necessary to bound the regret. Once $\mathcal{E}_t^{(j)}$ is included in $\mathcal{K}$, our algorithm escapes the loop of updating $\mathcal{K}_t^{(j)}$. An integer s_t denotes the number of the updates in the round t. We denote $\mathcal{E}_t = \mathcal{E}_t^{(s_t)}$, $\hat{\Sigma}_t = \hat{\Sigma}_t^{(s_t)}$, $\hat{\mu}_t = \hat{\mu}_t^{(s_t)}$ and $B_t = B_t^{(s_t)}$. We randomly choose x from $\mathcal{E}_t$ as follows: choose $\sigma_t \in \{-1, 1\}$ and $i_t \in [d]$ uniformly at random, and define $x = \hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t}$. If we can play this x, then we can construct a good estimate of ℓ_t from the above condition 4, which leads a small

Algorithm 5 An oracle efficient algorithm for non-stochastic bandit linear optimization

```
Require: Learning rate $\eta > 0$, error bound $\varepsilon > 0$, time horizon $T \in \mathbb{N}$, $R > 0$ satisfying Assumption 3.2.1.
 1: Set $\mathcal{K}_1^{(0)} = B_\infty(0, R) = \{x \in \mathbb{R}^d \mid \|x\|_\infty \leq R\}$, and define $z_1 : \mathbb{R}^d \to \mathbb{R}_{>0}$ by $z_1(x) = 1$.
 2: for $t = 1, 2, \ldots, T$ do
 3:   for $j = 0, 1, 2, \ldots$ do
 4:     Compute $\mathcal{E}_t^{(j)}$ on the basis of (3.1) $\sim$ (3.3).
 5:     Solve SP for $\mathcal{P} = \mathcal{K}$ and for each $y \in \mathcal{E}_t^{(j)}$.
 6:     if There is a hyperplane $w \in \mathbb{R}^d$ s.t. $w^\top y < \min_{x \in \mathcal{K}} w^\top x$ for some $y \in \mathcal{E}_t^{(j)}$ then
 7:       Update $\mathcal{K}_t^{(j)}$ by $\mathcal{K}_t^{(j+1)} = \mathcal{K}_t^{(j)} \cap \{x \in \mathbb{R}^d \mid w^\top x \geq w^\top y\}$.
 8:     else
 9:       Set $s_t = j$ and $\mathcal{K}_t = \mathcal{K}_t^{(s_t)}$. Break the for loop w.r.t. the index $j$.
10:     end if
11:   end for
12:   Let $\hat{\mu}_t = \hat{\mu}_t^{(s_t)}$ and $b_{ti} = b_{ti}^{(s_t)}$ for $i \in [n]$, which are defined in (3.1) $\sim$ (3.3).
13:   Choose $i_t \in [d]$ and $\sigma_t \in \{1, -1\}$ uniformly at random.
14:   Solve DP for $\mathcal{P} = \mathcal{K}$ and $x = \hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t}$ to get a decomposition $x_{t0}, \ldots, x_{td} \in \mathcal{A}$ and
      $\lambda_{t0}, \ldots, \lambda_{td}$ such that $\hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t} = \lambda_{t0} x_{t0} + \cdots + \lambda_{td} x_{td}$.
15:   Play $a_t = x_{ts}$ with probability $\lambda_{ts}$ $(s = 0, \ldots, d)$, and receive loss $\ell_t^\top a_t$.
16:   Set $\hat{\ell}_t$ by (3.4) and update $z_t$ by $z_{t+1}(x) = z_t(x) \exp(-\eta \hat{\ell}_t^\top (x - \hat{\mu}_t))$.
17:   Set $\mathcal{K}_{t+1}^{(0)} = \mathcal{K}_t^{(s_t)}$.
18: end for
```

degree of regret. However, $x \in \mathcal{E}_t$ does not always belong to $\mathcal{A}$, particularly when $\mathcal{A}$ is discrete. To cope with this issue, we solve DP for this x and $\mathcal{P} = \mathcal{K}$ to derive a decomposition of x, i.e., compute $x_{t0}, \ldots, x_{td} \in \mathcal{K}$ and $\lambda_{t0}, \ldots, \lambda_{td} \geq 0$ as in Step 14. The algorithm then plays $a_t = x_{ti}$ with probability λ_{ti}, and obtains feedback of $\ell_t^\top a_t$. Based on this feedback, we compute an estimator $\hat{\ell}_t$ of the loss vector ℓ_t as

$$\hat{\ell}_t = 4\mathrm{e} d \sigma_t \ell_t^\top a_t \hat{\Sigma}_t^{-1} b_{ti_t}. \tag{3.4}$$

This is an unbiased estimator of ℓ_t, i.e., we have $\mathbf{E}[\hat{\ell}_t] = \ell_t$. The existence of $\hat{\Sigma}_t^{-1}$ follows from the definition of $\hat{\Sigma}$ and Assumption 3.2.1. In fact, from $\mathcal{A} \subseteq \mathcal{K}_t^{(j)}$ and Assumption 3.2.1, $\mathcal{K}_t^{(j)}$ has a positive volume and $q_t^{(j)}$ has a positive density over $\mathcal{K}_t^{(j)}$, which implies that the covariance matrix $\Sigma_t^{(j)}$ of $q_t^{(j)}$ is positive definite. From this and (3.2), $\hat{\Sigma}_t^{(j)}$ is positive definite for all t and j. The function z_t is updated by $z_{t+1}(x) = z_t(x) \exp(-\eta \hat{\ell}_t^\top (x - \hat{\mu}_t))$, where $\eta > 0$ is an input parameter standing for the learning rate, which will be optimized later. Let

$$\delta_t^{(j)} = 1/(T(j + 2 + \textstyle\sum_{i=1}^{t-1}(s_i + 1))(j + 3 + \sum_{i=1}^{t-1}(s_i + 1))). \tag{3.5}$$

To compute $\hat{\Sigma}_t^{(j)}$ and $\hat{\mu}_t^{(j)}$ satisfying (3.2) with probability at least $1 - \delta_t^{(j)}$, we use the algorithm in Corollary 3.3.1.

Let $S_T = \sum_{t=1}^T s_t$ denote the number of updates of $\mathcal{K}_t^{(j)}$. We show the following regret bound.

Theorem 3.4.1. *Define $\psi = \frac{1}{d} \log \frac{\mathrm{Vol}(B_\infty(0,R))}{\mathrm{Vol}(\mathcal{K})}$. Suppose a_t is given by Algorithm 5 with parameters $\varepsilon = \frac{1}{12\mathrm{e}T}$ and $\eta = \frac{1}{2\mathrm{e}LR} \min\{\sqrt{\frac{1+\psi+\log T}{dT}}, \frac{1}{2^4 d^{3/2}(1+\psi+\log T)}\}$. Then, for all $a^* \in \mathcal{A}$, we* *have*

$$\mathbf{E}[R_T(a^*)] \leq 2^7 \mathrm{e} L R d^{3/2} \max\{\sqrt{T(1+\psi+\log T)}, d(1+\psi+\log T)^2\} \left(1 - S_T/2^{10}\right). \qquad (3.6)$$

We note that ψ in the above theorem satisfies $\psi \leq \log \frac{R}{r}$ if $\mathcal{K}$ includes an ℓ_∞-ball of radius $r > 0$. The proof of this theorem is given in Section 3.4.3.

3.4.2 Oracle Complexity Analysis

Here we show that Algorithm 5 calls the linear optimization oracles only $O(\mathrm{poly}(d)T)$ times.

To implement Algorithm 5, the linear optimization oracle is required only in Steps 5 and 14. In Step 5, we need to solve SP to decide if there exists $x \in \mathcal{E}_t^{(j)}$ such that $x \notin \mathcal{K}$. From the definition (3.3) of $\mathcal{E}_t^{(j)}$, the number of elements in $\mathcal{E}_t^{(j)}$ is equal to $2d$ for each t and j, and, accordingly, the total number of solvings of SP is $\sum_{t=1}^T \sum_{j=0}^{s_t} |\mathcal{E}_t^{(j)}| = 2d \sum_{t=1}^T (s_t + 1) = 2d(T + S_T)$. The number S_T can be bounded as $S_T = O(T)$. Indeed, from Theorem 3.4.1, if $S_T > 2^{10}(1+T)$ then $\mathbf{E}[R_T(a^*)] < -2^7 \mathrm{e} L R T$, which contradicts to $R_T(a^*) = \sum_{t=1}^T \ell_t^\top (a_t - a^*) \geq -2LRT$. Consequently, the total number of solvings of SP is $O(dT)$. In Step 14, we solve DP in each round t, and hence the total number of solvings of DP is equal to T. Since we can solve SP and DP by calling the linear optimization oracle $\mathrm{poly}(d, \log T)$ times from Theorem 3.3.2 and Remark 3.3.1, we can implement Algorithm 5 so that it calls the linear optimization oracle $O(\mathrm{poly}(d, \log T)T)$ times.

3.4.3 Regret Analysis

We use the following two lemmas to prove Theorem 3.4.1

Lemma 3.4.1. *The conditional expectation of $\hat{\ell}_t$ defined by* (3.4)*, given ℓ_t and $\mathcal{E}_t$, satisfies*

$$\mathbf{E}[\hat{\ell}_t] = \ell_t.$$

Proof. Since $\sum_{s=0}^d \lambda_{ts} x_{ts} = \hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t}$, the expectation of a_t given σ_t, i_t satisfies

$$\mathbf{E}[a_t] = \hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t}.$$

Hence, we have

$$\mathbf{E}[\sigma_t b_{ti_t} a_t^\top] = \mathbf{E}\left[\sigma_t b_{ti_t} \left(\hat{\mu}_t + \frac{\sigma_t}{4\mathrm{e}} b_{ti_t}\right)^\top\right] = \frac{1}{4\mathrm{e}} \mathbf{E}[b_{ti_t} b_{ti_t}^\top] = \frac{1}{4\mathrm{e}d} \sum_{i=1}^d b_{ti} b_{ti}^\top = \frac{1}{4\mathrm{e}d} \hat{\Sigma}_t,$$

where the second equality comes from $\mathbf{E}[\sigma_t] = 0$ and $\sigma_t^2 = 1$, the third equality holds since i_t follows a uniform distribution over $[d]$, and the last equality follows from the definition of $\{b_{ti}\}_{i=1}^d$. From the above equation and the definition (3.4) of $\hat{\ell}_t$, we have

$$\mathbf{E}[\hat{\ell}_t] = 4\mathrm{e}d \hat{\Sigma}_t^{-1} \mathbf{E}[\sigma_t b_{ti_t} a_t^\top] \ell_t = \ell_t.$$

□

Lemma 3.4.2. *Suppose that a random variable X follows a logconcave distribution and that $\mathbf{E}[X^2] \leq 1/\alpha^2$ holds for given $\alpha \geq 1$. Then, we have*

$$\mathbf{E}[\exp(X)\mathbf{1}_{X>1}] \leq \frac{\exp(3-\alpha)}{1 - \exp(1-\alpha)}.$$

Proof. From Lemma 5.7 in [127], we have

$$\mathrm{Prob}[|X| \geq i] \leq \exp(-\alpha i + 1)$$

for all $i \geq 1$. Hence, we have

$$\begin{aligned}\mathbf{E}[\exp(X)\mathbf{1}_{X>1}] &= \sum_{i=1}^{\infty} \mathbf{E}[\exp(X)\mathbf{1}_{i\leq X\leq i+1}] \leq \sum_{i=1}^{\infty} \mathrm{Prob}[i \leq X \leq i+1]\exp(i+1) \\ &\leq \sum_{i=1}^{\infty} \mathrm{Prob}[|X| \geq i]\exp(i+1) \leq \sum_{i=1}^{\infty} \exp((1-\alpha)i+2) = \frac{\exp(3-\alpha)}{1-\exp(1-\alpha)}.\end{aligned}$$

□

To prove Theorem 3.4.1, we introduce some notations: In the following, we denote

$$f_t(x) = \ell_t^\top(x - \hat{\mu}_t), \quad \hat{f}_t(x) = \hat{\ell}_t^\top(x - \hat{\mu}_t). \tag{3.7}$$

Then we can express z_t as

$$z_t(x) = \exp\left(-\eta \sum_{i=1}^{t-1} \hat{f}_i(x)\right). \tag{3.8}$$

Then, the regret can be bounded by means of $Z_{T+1}^{(0)}$ defined in (3.1), as follows:

Lemma 3.4.3. *For all $a^* \in \mathcal{A}$ and $\gamma \in (0,1)$, we have*

$$\mathbf{E}[R_T(a^*)] \leq \frac{1}{\eta(1-\gamma)}\left(\mathbf{E}\log Z_{T+1}^{(0)} - \log \mathrm{Vol}(\mathcal{K}) + \eta\gamma LRT - d\log\gamma\right). \tag{3.9}$$

Proof. From $\mathbf{E}[a_t|\hat{\mu}_t] = \hat{\mu}_t$ and $\mathbf{E}[\hat{\ell}_t|\hat{\mu}_t] = \ell_t$, we have

$$\mathbf{E}\,R_T(a^*) = \mathbf{E}\sum_{t=1}^{T} \ell_t^\top(a_t - a^*) = \mathbf{E}\sum_{t=1}^{T} \ell_t^\top(\hat{\mu}_t - a^*) = \mathbf{E}\sum_{t=1}^{T} \hat{\ell}_t^\top(\hat{\mu}_t - a^*) = -\,\mathbf{E}\sum_{t=1}^{T} \hat{f}_t(a^*). \tag{3.10}$$

We consider evaluating the rightmost-hand side, by using $Z_{T+1}^{(0)}$. Define a convex body $\bar{\mathcal{K}}$ by $\bar{\mathcal{K}} := (1-\gamma)a^* + \gamma\mathcal{K}$. Since $\mathcal{K}$ is convex and $a^* \in \mathcal{K}$, we have $\bar{\mathcal{K}} \subseteq \mathcal{K}$. Hence, we have $\bar{\mathcal{K}} \subseteq \mathcal{K}_t^{(j)}$ for all t and j. Then, we have

$$\begin{aligned}\mathbf{E}\log Z_{T+1}^{(0)} &= \mathbf{E}\log\int_{\mathcal{K}_T} z_{T+1}(x)\mathrm{d}x \geq \mathbf{E}\log\int_{\bar{\mathcal{K}}} z_{T+1}(x)\mathrm{d}x \\ &= \mathbf{E}\log\int_{\mathcal{K}} \gamma^d z_{T+1}((1-\gamma)a^* + \gamma x)\mathrm{d}x \\ &= d\log\gamma + \mathbf{E}\log\int_{\mathcal{K}} \exp\left(-\eta\sum_{t=1}^{T}((1-\gamma)\hat{f}_t(a^*) + \gamma\hat{f}_t(x))\right)\mathrm{d}x \\ &= d\log\gamma - \eta(1-\gamma)\,\mathbf{E}\sum_{t=1}^{T}\hat{f}_t(a^*) + \mathbf{E}\log\int_{\mathcal{K}}\exp(-\eta\gamma\sum_{t=1}^{T}\hat{f}_t(x))\mathrm{d}x.\end{aligned} \tag{3.11}$$

The factor γ^d in the second equality comes from the change of variables $x \leftarrow (1-\gamma)a^* + \gamma x$. The last term in (3.11) can be bounded from below by means of $\bar{x} := \frac{\int_{\mathcal{K}} x \mathrm{d}x}{\int_{\mathcal{K}} 1 \mathrm{d}x} \in \mathcal{K}$. Indeed, since $\hat{f}_t$ is an affine function and exp is a convex function, from Jensen's inequality, we have

$$\begin{aligned}
&\mathbf{E} \log \int_{\mathcal{K}} \exp(-\eta\gamma \sum_{t=1}^{T} \hat{f}_t(x))\mathrm{d}x \geq \mathbf{E} \log \int_{\mathcal{K}} \exp(-\eta\gamma \sum_{t=1}^{T} \hat{f}_t(\bar{x}))\mathrm{d}x \\
&= -\eta\gamma\, \mathbf{E} \sum_{t=1}^{T} \hat{f}_t(\bar{x}) + \log \int_{\mathcal{K}} 1 \mathrm{d}x = -\eta\gamma\, \mathbf{E} \sum_{t=1}^{T} f_t(\bar{x}) + \log \int_{\mathcal{K}} 1\mathrm{d}x \geq -\eta\gamma LRT + \log \mathrm{Vol}(\mathcal{K}).
\end{aligned}$$

By combining this and (3.11), we obtain

$$-\eta(1-\gamma)\, \mathbf{E} \sum_{t=1}^{T} \hat{f}_t(a^*) \leq \mathbf{E} \log Z_{T+1}^{(0)} - \log \mathrm{Vol}(\mathcal{K}) + \eta\gamma LRT - d \log \gamma.$$

From this and (3.10), we have (3.9). □

The value $\mathbf{E} \log Z_{T+1}^{(0)}$ can be expressed as

$$\mathbf{E} \log Z_{T+1}^{(0)} = \log Z_1^{(0)} + \sum_{t=1}^{T} \mathbf{E} \log \frac{Z_{t+1}^{(0)}}{Z_t^{(0)}} \leq \log Z_1^{(0)} + \sum_{t=1}^{T} \left(\mathbf{E} \log \frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}} + \mathbf{E} \log \frac{Z_t^{(s_t)}}{Z_t^{(0)}} \right). \tag{3.12}$$

We can evaluate $\mathbf{E} \log \frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}}$ and $\mathbf{E} \log \frac{Z_t^{(s_t)}}{Z_t^{(0)}}$ as in Lemmas 3.4.4 and 3.4.5, respectively.

Lemma 3.4.4. *Under the assumption that* (3.2) *and* $\eta \leq \frac{1}{2^3 \mathrm{e} LRd^{3/2}(4+\log T)}$ *holds for all* $t \in [T]$, *we have*

$$\mathbf{E} \log \frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}} \leq LR\varepsilon\eta + 2^5(\mathrm{e}dLR\eta)^2 + \frac{1}{T}. \tag{3.13}$$

Proof. We denote $q_t := q_t^{(s_t)}$, which is defined in (3.1). Let $\mu_t \in \mathbb{R}^d$ and $\Sigma_t \in \mathbb{R}^{d\times d}$ denote the mean and the covariance matrix of q_t, respectively. Define $\tilde{f}_t : \mathbb{R}^d \to \mathbb{R}$ by $\tilde{f}_t(x) = \hat{\ell}_t^\top (x - \mu_t)$. From the definition (3.7) of $\hat{f}_t$, we have $\hat{f}_t(x) = \tilde{f}_t(x) + \hat{\ell}_t^\top(\mu_t - \hat{\mu}_t)$. From the definitions (3.8) and (3.1) of z_t and Z_t^j, we can express $\frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}}$ as follows:

$$\begin{aligned}
\frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}} &= \int_{\mathcal{K}_t} \frac{z_{t+1}(x)}{Z_t^{(s_t)}} \mathrm{d}x = \int_{\mathcal{K}_t} \frac{z_t(x)}{Z_t^{(s_t)}} \exp(-\eta \hat{f}_t(x)) \mathrm{d}x \\
&= \mathop{\mathbf{E}}_{x\sim q_t} \exp(-\eta \hat{f}_t(x)) = \exp(-\eta \hat{\ell}_t^\top (\mu_t - \hat{\mu}_t)) \cdot \mathop{\mathbf{E}}_{x\sim q_t} \exp(-\eta \tilde{f}_t(x)).
\end{aligned} \tag{3.14}$$

Since $\exp(x) \leq 1 + x + x^2$ holds for $x \leq 1$, $\mathop{\mathbf{E}}_{x\sim q_t} \exp(-\eta\tilde{f}_t(x))$ can be bounded as

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim q_t} \exp(-\eta\tilde{f}_t(x)) &\leq \mathop{\mathbf{E}}_{x\sim q_t} [(1 - \eta\tilde{f}_t(x) + \eta^2 \tilde{f}_t(x)^2)\mathbf{1}_{-\eta\tilde{f}_t(x)\leq 1}] + \mathop{\mathbf{E}}_{x\sim q_t} [\exp(-\eta\tilde{f}_t(x))\mathbf{1}_{-\eta\tilde{f}_t(x)>1}] \\
&\leq \mathop{\mathbf{E}}_{x\sim q_t} [(1 - \eta\tilde{f}_t(x) + \eta^2\tilde{f}_t(x)^2)] + \mathop{\mathbf{E}}_{x\sim q_t} [\exp(-\eta\tilde{f}_t(x))\mathbf{1}_{-\eta\tilde{f}_t(x)>1}].
\end{aligned} \tag{3.15}$$

The first term in (3.15) can be, from the definition of μ_t, Σ_t and $\tilde{f}_t$, expressed as follows:

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim q_t}[(1-\eta\tilde{f}_t(x)+\eta^2\tilde{f}_t(x)^2)] &= 1-\eta\mathop{\mathbf{E}}_{x\sim q_t}[\hat{\ell}_t^\top(x-\mu_t)]+\eta^2\mathop{\mathbf{E}}_{x\sim q_t}[\hat{\ell}_t^\top(x-\mu_t)(x-\mu_t)\hat{\ell}_t^\top] \\ &= 1+\eta^2\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t. \qquad (3.16)\end{aligned}$$

To bound the second term in (3.15), we use Lemma 3.4.2 with $X=-\eta\tilde{f}_t(x)$: If x follows q_t, a logconcave distribution, and if $\hat{\ell}_t$ is fixed, i.e., ℓ_t, σ_t, i_t and a_t are fixed, then $\tilde{f}_t(x)=\hat{\ell}_t(x-\mu_t)=4\mathrm{e}d\sigma_t\ell_t^\top a_t b_{ti_t}\hat{\Sigma}_t^{-1}(x-\mu_t)$ follows a logconcave distribution because logconcavity is preserved under linear transformations (see, e.g., [142]). Furthermore, we have[5]

$$\begin{aligned}&\mathop{\mathbf{E}}_{x\sim q_t}[(\eta\tilde{f}_t(x))^2]=\eta^2\mathop{\mathbf{E}}_{x\sim q_t}[\hat{\ell}_t^\top(x-\mu_t)(x-\mu_t)^\top\hat{\ell}_t]=\eta^2\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t=(4\mathrm{e}d\eta\ell_t^\top a_t)^2 b_{ti_t}^\top\hat{\Sigma}_t^{-1}\Sigma_t\hat{\Sigma}_t^{-1}b_{ti_t} \\ &\le 2(4\mathrm{e}d\eta LR)^2 b_{ti_t}^\top\hat{\Sigma}_t^{-1}b_{ti_t}=2(4\mathrm{e}d\eta LR)^2\hat{\Sigma}_t^{-1}\bullet(b_{ti_t}b_{ti_t}^\top)\le 2^5(\mathrm{e}d\eta LR)^2\hat{\Sigma}_t^{-1}\bullet\hat{\Sigma}_t=2^5(\mathrm{e}\eta LR)^2d^3,\end{aligned}$$

where the first, second and third equalities come from the definitions of $\tilde{f}_t$, Σ_t and $\hat{\ell}_t$, respectively, the first inequality follows from the condition $\Sigma_t \preceq 2\hat{\Sigma}_t$ given in (3.2), and the second inequality follows from that $\hat{\Sigma}_t=\sum_{i=1}^d b_{ti}b_{ti}^\top$. Hence, under the assumption that $2^5(\mathrm{e}\eta LR)^2d^3(4+\log T)^2\le 1$ holds, it follows from Lemma 3.4.2 that

$$\mathop{\mathbf{E}}_{x\sim q_t}[\exp(-\eta\tilde{f}_t(x))\mathbf{1}_{-\eta\hat{f}_t(x)>1}]\le\frac{\exp(-1-\log T)}{1-\exp(-3-\log T)}\le\frac{1}{T}.$$

Combining this, (3.15) and (3.16), we obtain

$$\mathop{\mathbf{E}}_{x\sim q_t}\exp(-\eta\tilde{f}_t(x))\le 1+\eta^2\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t+\frac{1}{T}.$$

From this, (3.14), Lemma 3.4.1 and the fact that $\log(1+x)\le x$ holds for $x\ge 0$, we have

$$\begin{aligned}\mathbf{E}\log\frac{Z_{t+1}^{(0)}}{Z_t^{(s_t)}}&\le-\eta\,\mathbf{E}\,\ell_t^\top(\mu_t-\hat{\mu}_t)+\mathbf{E}\log\left(1+\eta^2\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t+\frac{1}{T}\right)\\&\le-\eta\,\mathbf{E}\,\ell_t^\top(\mu_t-\hat{\mu}_t)+\mathbf{E}\left[\eta^2\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t\right]+\frac{1}{T}.\end{aligned}$$

From (3.2), we have

$$|\ell_t^\top(\mu_t-\hat{\mu}_t)|\le\|\ell_t\|_2\|\mu_t-\hat{\mu}_t\|_2\le L\|\mu_t-\hat{\mu}_t\|_{\Sigma_t^{-1}}\|\Sigma_t\|_2\le LR\varepsilon,$$

where $\|\Sigma_t\|_2$ stands for the ℓ_2 operator norm, i.e., the largest eigenvalue of Σ_t, and the last inequality holds because Σ_t is the covariance matrix of distribution over a region included in $B_\infty(0,R)$. Furthermore, we have

$$\begin{aligned}\mathbf{E}\left[\hat{\ell}_t^\top\Sigma_t\hat{\ell}_t\right]&\le(4\mathrm{e}dLR)^2\Sigma_t^{-1}\bullet\mathbf{E}[b_{ti_t}b_{ti_t}^\top]=(4\mathrm{e}dLR)^2\Sigma_t^{-1}\bullet\left(\frac{1}{d}\sum_{i=1}^d b_{ti}b_{ti}^\top\right)\\&=(4\mathrm{e}dLR)^2\Sigma_t^{-1}\bullet\left(\frac{1}{d}\hat{\Sigma}_t\right)\le 2(4\mathrm{e}dLR)^2.\end{aligned}$$

By combining the above three inequalities, we obtain (3.13). □

[5] $A\bullet B$ means the Frobenius inner product of matrices A and B, i.e., its value is defined to be the trace of $A^\top B$.

Lemma 3.4.5. *Suppose $\varepsilon \leq 1/(12\mathrm{e})$. For all $t \in [T]$ and $j \in \{0, 1, \ldots, s_t - 1\}$, under the assumption of* (3.2) *we have*

$$\frac{Z_t^{(j+1)}}{Z_t^{(j)}} \leq \left(1 - \frac{1}{2\mathrm{e}}\right). \tag{3.17}$$

Proof. Let (w, b) denote the hyperplane that the algorithm chooses for updating $\mathcal{K}_t^{(j)}$, which means that $\mathcal{K}_t^{(j+1)} = \mathcal{K}_t^{(j)} \cap \{x \in \mathbb{R}^d \mid w^\top x \geq b\}$. Then, we have

$$1 - \frac{Z_t^{(j+1)}}{Z_t^{(j)}} = \frac{\int_{\mathcal{K}_t^{(j)} \setminus \mathcal{K}_t^{(j+1)}} z_t(a) \mathrm{d}a}{\int_{\mathcal{K}_t^{(j)}} z_t(a) \mathrm{d}a} = \operatorname*{Prob}_{x \sim q_t^{(j)}}[w^\top x < b].$$

Since (w, b) satisfies $w^\top x \leq b$ for some $x \in \mathcal{E}_t^{(j)}$, there exists $i \in [d]$ and $\sigma \in \{1, -1\}$ such that

$$w^\top \left(\hat{\mu}_t^{(j)} + \frac{\sigma}{4\mathrm{e}} b_{ti}^{(j)}\right) \leq b.$$

Combining the above equality and inequality, we obtain

$$\begin{aligned}
1 - \frac{Z_t^{(j+1)}}{Z_t^{(j)}} &\geq \operatorname*{Prob}_{x \sim q_t^{(j)}} \left[w^\top x < w^\top \left(\hat{\mu}_t^{(j)} + \frac{\sigma}{4\mathrm{e}} b_{ti}^{(j)}\right)\right] \\
&= \operatorname*{Prob}_{x \sim q_t^{(j)}} \left[w^\top \left(x - \mu_t^{(j)}\right) < w^\top \left(\hat{\mu}_t^{(j)} - \mu_t^{(j)} + \frac{\sigma}{4\mathrm{e}} b_{ti}^{(j)}\right)\right].
\end{aligned}$$

The value $w^\top \left(\hat{\mu}_t^{(j)} - \mu_t^{(j)} + \frac{\sigma}{4\mathrm{e}} b_{ti}^{(j)}\right)$ can be bounded as

$$\begin{aligned}
\left|w^\top \left(\hat{\mu}_t^{(j)} - \mu_t^{(j)} + \frac{\sigma}{4\mathrm{e}} b_{ti}^{(j)}\right)\right| &\leq \|w\|_{\hat{\Sigma}_t^{(j)}} \left(\|\hat{\mu}_t^{(j)} - \mu_t^{(j)}\|_{\hat{\Sigma}_t^{(j)-1}} + \frac{1}{4\mathrm{e}} \|b_{ti}^{(j)}\|_{\hat{\Sigma}_t^{(j)-1}}\right) \\
&\leq \|w\|_{\hat{\Sigma}_t^{(j)}} \left(\varepsilon + \frac{1}{4\mathrm{e}}\right) \leq \sqrt{2} \|w\|_{\Sigma_t^{(j)}} \left(\varepsilon + \frac{1}{4\mathrm{e}}\right) \leq \frac{\|w\|_{\Sigma_t^{(j)}}}{2\mathrm{e}},
\end{aligned}$$

where the first inequality comes from the Cauchy–Schwarz inequality, the second inequality follows from (3.2) and that $\Sigma_t^{(j)} = \sum_{i=1}^d b_{ti}^{(j)} b_{ti}^{(j)\top}$, the third inequality follows from $\hat{\Sigma}_t^{(j)} \succeq 2\Sigma_t^{(j)}$ in (3.2), and the last inequality comes from the assumption of $\varepsilon < 1/(12\mathrm{e})$. From the above two inequations, we have

$$1 - \frac{Z_t^{(j+1)}}{Z_t^{(j)}} \geq \operatorname*{Prob}_{x \sim q_t^{(j)}} \left[\frac{w^\top (x - \mu_t^{(j)})}{\|w\|_{\Sigma_t^{(j)}}} < -\frac{1}{2\mathrm{e}}\right].$$

When x follows $q_t^{(j)}$, $y := \frac{w^\top (x - \mu_t^{(j)})}{\|w\|_{\Sigma_t^{(j)}}}$ follows a distribution with mean 0 and variance 1 since $w^\top x$ has mean $w^\top \mu_t^{(j)}$ and variance $w^\top \Sigma_t^{(j)} w$. Moreover, the PDF of y is a logconcave function because logconcavity is preserved under linear transformations [142]. Since we have $\mathrm{Prob}[y \leq 0] \geq 1/\mathrm{e}$ from Lemma 5.4 in [127] and $\mathrm{Prob}[-1/(2\mathrm{e}) \leq y \leq 0] \leq 1/(2\mathrm{e})$ from Lemma 5.5 in [127], we have $\mathrm{Prob}[y < -1/(2\mathrm{e})] \geq 1/(2\mathrm{e})$. □

Combining (3.12) and Lemmas 3.4.4 and 3.4.5, under the assumption that (3.2) holds for all $t \in [T]$ and $j \in \{0, 1, \ldots, s_t\}$, we have

$$\begin{aligned}
\mathbf{E} \log Z_{T+1}^{(0)} &\leq \log Z_1^{(0)} + T \left(LR\varepsilon\eta + 2^5 (\mathrm{e}dLR\eta)^2 + \frac{1}{T}\right) + \sum_{t=1}^T s_t \log \left(1 - \frac{1}{2\mathrm{e}}\right) \\
&\leq \log Z_1^{(0)} + LRT\varepsilon\eta + 2^5 T (\mathrm{e}dLR\eta)^2 + 1 - \frac{S_T}{5},
\end{aligned} \tag{3.18}$$

where we denote $S_T = \sum_{t=1}^{T} s_t$ and the second inequality follows from $\log\left(1 - \frac{1}{2\mathrm{e}}\right) \leq \frac{1}{5}$. Define $\delta :=$ Prob[there exists $t \in [T]$ and $j \leq s_t$ such that (3.2) does not hold]. From the definition (3.5) of $\delta_t^{(j)}$, we have $\delta \leq \sum_{t=1}^{T} \sum_{j=0}^{s_t} \leq \sum_{k=2}^{\infty} \frac{1}{Tk(k+1)} = \frac{1}{2T}$. Since (3.18) holds with probability at least $1 - \delta$, and it always holds that $R_T(a^*) \leq 2LRT$, from Lemma 3.4.3, we have

$$
\begin{aligned}
&\mathbf{E}[R_T(a^*)] \\
&\leq \frac{1-\delta}{1-\gamma}\left(2^5 T(\mathrm{e}dLR)^2\eta + \frac{1}{\eta}\left(\log\frac{Z_1^{(0)}}{\mathrm{Vol}(\mathcal{K})} + 1 - d\log\gamma - \frac{S_T}{5}\right) + LRT(\varepsilon+\gamma)\right) + 2\delta LRT \\
&\leq \frac{1}{1-\gamma}\left(2^5 T(\mathrm{e}dLR)^2\eta + \frac{1}{\eta}\left(\log\frac{Z_1^{(0)}}{\mathrm{Vol}(\mathcal{K})} + 1 - d\log\gamma - \frac{S_T}{10}\right) + LRT(\varepsilon+\gamma)\right) + LR,
\end{aligned}
$$

where the second inequality comes form $0 \leq \delta \leq \frac{1}{2T}$. Let $\psi := \frac{1}{d}\log\frac{Z_1^{(0)}}{\mathrm{Vol}(\mathcal{K})} = \frac{1}{d}\log\frac{\mathrm{Vol}(B_\infty(0,R))}{\mathrm{Vol}(\mathcal{K})}$. By setting $\varepsilon = \gamma = \frac{1}{12eT}$, we obtain

$$
\mathbf{E}[R_T(a^*)] \leq 2\left(2^5 T(\mathrm{e}dLR)^2\eta + \frac{1}{\eta}\left(d(5+\psi+\log T) - \frac{S_T}{10}\right) + 2LR\right).
$$

In addition, setting $\eta = \frac{1}{2\mathrm{e}LR}\min\left\{\sqrt{\frac{1+\psi+\log T}{dT}}, \frac{1}{2^4 d^{3/2}(1+\psi+\log T)}\right\}$ we obtain

$$
\mathbf{E}[R_T(a^*)] \leq 2^7 \mathrm{e}LRd^{3/2}\max\left\{\sqrt{T(1+\psi+\log T)}, d(1+\psi+\log T)^2\right\}\left(1 - \frac{S_T}{2^{10}}\right),
$$

which means that (3.6) holds. □

3.5 Algorithm for Stochastic Bandit Linear Optimization

In this section, we present an algorithm for stochastic bandit linear optimization, where we assume that ℓ_t follows a distribution D over $\mathbb{R}^d$, i.i.d. for $t = 1, 2, \ldots, T$. We denote $\ell^* = \mathop{\mathbf{E}}_{\ell \sim D}[\ell] \in \mathbb{R}^d$ and $\xi_t = \ell_t - \ell^*$.

3.5.1 Algorithm

Our algorithm is summarized in Algorithm 6. In the algorithm, a parameter $\delta > 0$ controls the probability of achieving a small regret. The rounds are divided into $K = O(\log T)$ *phases* so that the k-th phase consists of $\Theta(2^k)$ rounds for each $k \in \{1, \ldots, K\}$.

When the k-th phase begins, the algorithm maintains an action set $\mathcal{P}_k$. This action set is initialized by $\mathcal{P}_1 = \mathrm{Conv}(\mathcal{A})$, and is defined recursively by (3.20) (thus $\mathcal{P}_1 \supseteq \mathcal{P}_2 \supseteq \cdots \supseteq \mathcal{P}_K$). $\mathcal{P}_k$ is designed so that the value of $\ell^{*\top}x$ is smaller for all $x \in \mathcal{P}_k$ as k gets larger (see Lemma 3.5.2). At the beginning of the k-th phase, the algorithm computes a 2-barycentric spanner $X_k = \{x_{k1}, \ldots, x_{kd}\}$ of $\mathcal{P}_k$. We can construct a good estimate of ℓ^* if each element of X_k can be chosen as an action. However, elements in X_k do not always belong to $\mathcal{A}$, especially when $\mathcal{A}$ is discrete. To cope with this issue, our algorithm decomposes each x_{ki} into the points $x_{ki0}, \ldots, x_{kid} \in \mathcal{A}$ with weight $\lambda_{ki0}, \ldots, \lambda_{kid} \geq 0$ so that $\lambda_{ki0} + \cdots + \lambda_{kid} = 1$ and $\lambda_{ki0}x_{ki0} + \cdots + \lambda_{kid}x_{kid} = x_{ki}$. It then plays x_{kij}, $T_{kij} \propto \lambda_{kij}$ times, for each $j = 0, 1, \ldots, d$. We denote the action played at the t-th round by a_t. The algorithm computes an empirical estimate $\hat{\ell}_k$ of ℓ^*, based on the feedback obtained in k-th phase, as defined in (3.19).

Algorithm 6 An oracle efficient algorithm for stochastic bandit linear optimization

```
Require: Action set $\mathcal{A} \subseteq \mathbb{R}^d$, positive real numbers $L$ and $R$ satisfying Assumption 3.2.1, $\delta \in (0,1)$.
1:  Set $\mathcal{P}_1 = \mathrm{Conv}(\mathcal{A})$ and $t_{1,1,0} = 0$.
2:  for $k = 1, 2, \ldots, K$ do
3:      Let $X_k = \{x_{k1}, \ldots, x_{kd}\} \subseteq \mathcal{P}_k$ be a 2-barycentric spanner for $\mathcal{P}_k$.
4:      Set $\zeta_k = 2^{2k+9} d^2 \log\left(\frac{2dk(k+1)}{\delta}\right)$.
5:      for $i = 1, \ldots, d$ do
6:          Solve DP for $\mathcal{P} = \mathrm{Conv}(\mathcal{A})$ and $x = x_{ki}$ to get a decomposition $x_{ki0}, \ldots, x_{kid} \in \mathcal{A}$ and $\lambda_{ki0}, \ldots, \lambda_{kid}$ such that $x_{ki} = \lambda_{ki0} x_{ki0} + \cdots + \lambda_{kid} x_{kid}$.
7:          for $j = 0, \ldots, d$ do
8:              Set $T_{kij} = \lceil \zeta_k \lambda_{kij} \rceil, t_{ki,j+1} = t_{kij} + T_{kij}$.
9:              Choose action $a_t = x_{kij}$ exactly $T_{kij}$ times, from the $(t_{kij}+1)$-th round to the $(t_{ki,j+1})$-th round.
10:         end for
11:         Set $t_{k,i+1,0} = t_{ki,d+1}$.
12:     end for
13:     Calculate empirical estimate $\hat{\ell}_k$ of $\ell^*$ by
            $$V_k = \sum_{i=1}^{d} \sum_{j=0}^{d} T_{kij} x_{kij} x_{kij}^\top, \quad \hat{\ell}_k = V_k^{-1} \sum_{i=1}^{d} \sum_{j=0}^{d} \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (\ell_t^\top x_{kij}) x_{kij}. \qquad (3.19)$$
14:     Solve LP for $\mathcal{P} = \mathcal{P}_k$ and $w = \hat{\ell}_k$ to find a vector $x_k^* \in \arg\min_{x \in \mathcal{P}_k} \hat{\ell}_k^\top x$.
15:     Update $\mathcal{P}_k$ by
            $$\mathcal{P}_{k+1} = \{x \in \mathcal{P}_k \mid \hat{\ell}_k^\top (x - x_k^*) \le LR2^{-k}\}. \qquad (3.20)$$
16: end for
```

We note that $\mathrm{Vol}(\mathcal{P}_k) > 0$ holds for all k, which implies that we can apply the algorithm in Theorem 3.3.3 to $\mathcal{P}_k$ and that V_k is invertible. In fact, we have $\mathrm{Vol}(\mathcal{P}_1) > 0$ from Assumption 3.2.1 and we can show $\mathrm{Vol}(\mathcal{P}_k) > 0$ by induction in k, from the definition (3.20). The linear span of a barycentric spanner $X_k = \{x_{k1}, \ldots, x_{kd}\}$ coincides with that of $\mathcal{P}_k$ (see, e.g., [17]), which is equal to $\mathbb{R}^d$ since $\mathrm{Vol}(\mathcal{P}_k) > 0$. Hence, we have $V_k \succeq \zeta_k \sum_{i=1}^d \sum_{j=0}^d \lambda_{kij} x_{kij} x_{kij}^\top \succeq \zeta_k \sum_{i=1}^d x_{ki} x_{ki}^\top \succ O$, which means that V_k is nonsingular.

Algorithm 6 satisfies the following regret bound:

Theorem 3.5.1. *Suppose that ℓ_t follows an i.i.d. distribution for $t = 1, \ldots, T$ with $T \ge 2$, and that $\{a_t\}_{t=1}^T$ is given by Algorithm 6. With probability at least $1 - \delta$, the regret is bounded as follows:*

$$\max_{a \in \mathcal{A}} R_T(a) \le 2^{12} LR \sqrt{d^3 T \log\left(\frac{d \log T}{\delta}\right)}. \qquad (3.21)$$

The proof of this theorem is given in Section 3.5.3.

3.5.2 Oracle Complexity Analysis

Step 3 in Algorithm 6 requires constructing a 2-barycentric spanner. From Theorem 3.3.3, we can construct it by calling an algorithm that solves LP for $\mathcal{P} = \mathcal{P}_k$, $O(\mathrm{poly}(d))$ times. Theorem 3.3.1 (a) and Remark 3.3.1 imply that we can solve LP, by solving SP $O(\mathrm{poly}(d, \log T))$ times. SP for $\mathcal{P} = \mathcal{P}_k$ can be solved by the following procedure:

1. Decide if $y \in \mathcal{P}_1$ or not, and, in the latter case, output a vector $w \in \mathbb{R}^d$ such that $w^\top y < \min_{x \in \mathcal{P}_1} w^\top x$. From Theorem 3.3.1 (b), this can be done by calling the LP oracle for $\mathcal{A}$ $O(\mathrm{poly}(d))$ times. In the former case, i.e., if $y \in \mathcal{P}_1$, go to the next step.

2. For $j = 1, \ldots, k-1$, if $\hat{\ell}_j^\top (y - x_k^*) > LR2^{-j}$, output $\hat{\ell}_j$. If $\hat{\ell}_j^\top (y - x_k^*) \leq LR2^{-j}$ for all $j = 1, \ldots, k-1$, it means that $y \in \mathcal{P}_k$.

This procedure calls the LP oracle for $O(\mathrm{poly}(d, \log T))$ times and runs in $O(\mathrm{poly}(d, K)) = O(\mathrm{poly}(d, \log T))$ times. Hence, we have an efficient implementation of Step 3 that calls the LP oracle for $\mathcal{A}$ $O(\mathrm{poly}(d, \log T))$ times. Similarly, Steps 6 and 14 in Algorithm 6 can be executed by calling the LP oracle for $\mathcal{A}$ $O(\mathrm{poly}(d, \log T))$ times. The other steps are free from access to the oracle and can be efficiently implemented. Since the number K of iterations w.r.t. k is bounded as in (3.31), the number of oracle calls for solving LP over $\mathcal{A}$ is of $O(\mathrm{poly}(d, \log T) K) = O(\mathrm{poly}(d, \log T))$.

3.5.3 Regret Analysis

In the proof of Theorem 3.5.1 we may assume that

$$T > 2d\zeta_1 = 2^{12} d^3 \log \left(\frac{4d}{\delta} \right). \tag{3.22}$$

Indeed, if $T \leq 2d\zeta_1$, then we see $R_T(a) \leq 2LRT \leq 2LR\sqrt{2d\zeta_1 T}$, which means that (3.21) holds.

To prove the above theorem, we start with analyzing the error of the estimators $\hat{\ell}_k$ defined by (3.19):

Lemma 3.5.1. *With probability at least $1 - \delta$, for all $k \in \{1, 2, \ldots\}$ and $x \in \mathcal{P}_k$, we have*

$$|(\hat{\ell}_k - \ell^*)^\top x| \leq 2^{-1-k} LR. \tag{3.23}$$

Proof. Since $X_k = \{x_{k1}, \ldots, x_{kd}\}$ is a 2-barycentric spanner for $\mathcal{P}_k$, for all $x \in \mathcal{P}_k$, there exists a vector $w = (w_1, \ldots, w_d)^\top \in [-2, 2]^d$ such that $x = w_1 x_{k1} + \cdots + w_d x_{kd}$. By means of this w, $(\hat{\ell}_k - \ell^*)^\top x$ can be expressed as

$$\begin{aligned}
(\hat{\ell}_k - \ell^*)^\top x &= \left(\sum_{i=1}^{d} \sum_{j=0}^{d} \sum_{t=t_{kij}+1}^{t_{ki,j+1}} \ell_t^\top x_{kij} x_{kij}^\top V_k^{-1} - \ell^{*\top} \right) \sum_{s=1}^{d} w_s x_{ks} \\
&= \left(\sum_{i=1}^{d} \sum_{j=0}^{d} \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (\ell_t - \ell^*)^\top x_{kij} x_{kij}^\top V_k^{-1} \right) \sum_{s=1}^{d} w_s x_{ks} \\
&= \sum_{s=1}^{d} w_s \sum_{i=1}^{d} \sum_{j=0}^{d} \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (\xi_t^\top x_{kij}) x_{kij}^\top V_k^{-1} x_{ks},
\end{aligned} \tag{3.24}$$

where the first equality comes form the definition (3.19) of $\hat{\ell}_k$, the second equality comes from $\sum_{i=1}^d \sum_{j=0}^d \sum_{t=t_{kij}+1}^{t_{ki,j+1}} x_{kij} x_{kij}^\top V_k^{-1} = \sum_{i=1}^d \sum_{j=0}^d T_{kij} x_{kij} x_{kij}^\top V_k^{-1} = I$, and the last equality comes from $\xi_t = \ell_t - \ell^*$. We give a uniform bound for this value by the following claim: with probability at least $1-\delta$, it holds for all $k = 1, 2, \ldots$ and $s = 1, \ldots, d$ that

$$\left| \sum_{i=1}^d \sum_{j=0}^d \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (\xi_t^\top x_{kij}) x_{kij}^\top V_k^{-1} x_{ks} \right| \leq \frac{LR}{d 2^{k+2}}. \tag{3.25}$$

Since the expectation of $\xi_t = \ell_t - \ell^*$ is equal to 0, and since $\|\xi_t\|_2 \leq \|\ell_t\|_2 + \|\ell^*\|_2 \leq 2L$ and $\|x_{kij}\|_2 \leq R$ hold from Assumption 3.2.1, $\{\xi_t^\top x_{kij}\}$ are independent random variables with mean 0 and absolute value at most $2LR$. Hence, from Hoeffding's inequality, it holds with probability at least $1 - \frac{\delta}{dk(k+1)}$ that

$$\begin{aligned}
&\left| \sum_{i=1}^d \sum_{j=0}^d \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (\xi_t^\top x_{kij}) x_{kij}^\top V_k^{-1} x_{ks} \right| \\
&\leq 2LR \sqrt{8 \log\left(\frac{2dk(k+1)}{\delta}\right) \sum_{i=1}^d \sum_{j=0}^d \sum_{t=t_{kij}+1}^{t_{ki,j+1}} (x_{kij}^\top V_k^{-1} x_{ks})^2} \\
&= 2LR \sqrt{8 \log\left(\frac{2dk(k+1)}{\delta}\right) x_{ks}^\top V_k^{-1} \left(\sum_{i=1}^d \sum_{j=0}^d \sum_{t=t_{kij}+1}^{t_{ki,j+1}} x_{kij} x_{kij}^\top \right) V_k^{-1} x_{ks}} \\
&= 2LR \sqrt{8 \log\left(\frac{2dk(k+1)}{\delta}\right) x_{ks}^\top V_k^{-1} x_{ks}}.
\end{aligned} \tag{3.26}$$

The value $x_{ks}^\top V_k^{-1} x_{ks}$ is bounded as $x_{ks}^\top V_k^{-1} x_{ks} \leq \zeta_k^{-1}$. Indeed, we have

$$\begin{aligned}
V_k &\succeq \sum_{j=0}^d T_{ksj} x_{ksj} x_{ksj}^\top \succeq \zeta_k \sum_{j=0}^d \lambda_{ksj} x_{ksj} x_{ksj}^\top \\
&= \zeta_k \left(x_{ks} x_{ks}^\top + \sum_{j=0}^d \lambda_{ksj} (x_{ksj} - x_{ks})(x_{ksj} - x_{ks})^\top \right) \succeq \zeta_k x_{ks} x_{ks}^\top,
\end{aligned}$$

where the first inequality comes from the definition (3.19) of V_k, the second inequality comes from $T_{ksj} = \lceil \zeta_k \lambda_{ksj} \rceil \geq \zeta_k \lambda_{ksj}$ (Step 8 of Algorithm 6), and the equality comes from that $\lambda_{ks0} + \cdots + \lambda_{ksd} = 1$ and that $\lambda_{ks0} x_{ks0} + \cdots + \lambda_{ksd} x_{ksd} = x_{ks}$. This inequality indicates

$$0 \leq (V_k^{-1} x_{ks})^\top (V_k - \zeta_k x_{ks} x_{ks}^\top)(V_k^{-1} x_{ks}) = x_{ks}^\top V_k^{-1} x_{ks} (1 - \zeta_k x_{ks}^\top V_k^{-1} x_{ks}),$$

from which we have $x_{ks}^\top V_k^{-1} x_{ks} \leq \zeta_k^{-1}$.

Plugging this bound on $x_{ks}^\top V_k^{-1} x_{ks}$ into (3.26) and the definition of ζ_k (Step 4 of Algorithm 6) show that the inequality (3.25) holds with probability at least $1 - \frac{\delta}{dk(k+1)}$ for each k and s. Hence, we have

$$\begin{aligned}
\text{Prob[(3.25) does not hold for some } k \text{ and } s \text{]} &\leq \sum_{k=1}^K \sum_{s=1}^d \text{Prob[(3.25) does not hold for } k \text{ and } s \text{]} \\
&\leq \sum_{k=1}^K \sum_{s=1}^d \frac{\delta}{dk(k+1)} = \delta,
\end{aligned}$$

which means that, with probability at least $1-\delta$, (3.25) holds for all k and s. Combining this, (3.24), and $|w_s| \leq 2$ for all $s \in [d]$, we obtain (3.23). $\square$

Lemma 3.5.2. *Fix $a^* \in \arg\min_{a \in \mathcal{A}} \ell^{*\top} a$. With probability at least $1-\delta$, for all k, we have*

$$a^* \in \mathcal{P}_k, \text{ and } \ell^{*\top} x - \ell^{*\top} a^* \leq 2^{2-k} LR \text{ for all } x \in \mathcal{P}_k. \tag{3.27}$$

Proof. From Lemma 3.5.1, we can assume that (3.23) holds for all k and $x \in \mathcal{P}_k$. Under this assumption, we show (3.27) by induction in k. We can confirm that (3.27) holds for $k=1$. Indeed, $a^* \in \mathcal{P}_1$ follows from $\mathcal{A} \subseteq \mathcal{P}_1$, and $\ell^{*\top} x - \ell^{*\top} a^* \leq \|\ell^*\|_2 \|x - a^*\|_2 \leq 2LR$ follows from $\|\ell^*\|_2 \leq L$ and $\|x\|_2 \leq R$ for $x \in \mathcal{P}_1$. Suppose that (3.27) holds for $k = s$. Then, $\mathcal{P}_{s+1}$, defined by (3.20), contains a^* because

$$\hat{\ell}_s^\top a^* \leq \ell^{*\top} a^* + 2^{-1-s} LR \leq \ell^{*\top} x_s^* + 2^{-1-s} LR \leq \hat{\ell}_s^\top x_s^* + 2^{-s} LR,$$

where the first and the third inequalities come from (3.23) and the second inequality comes from $\ell^{*\top} a^* = \min_{a \in \mathcal{A}} \{\ell^{*\top} a\} = \min_{x \in \mathcal{P}_1} \{\ell^{*\top} x\} \leq \ell^{*\top} x_s^*$. Furthermore, for all $x \in \mathcal{P}_{s+1}$, we have

$$\ell^{*\top} x \leq \hat{\ell}_s^\top x + 2^{-s-1} LR \leq \hat{\ell}_s^\top x_s^* + 3 \cdot 2^{-s-1} LR \leq \hat{\ell}_s^\top a^* + 3 \cdot 2^{-s-1} LR \leq \ell^{*\top} a^* + 2^{-s+1} LR,$$

where the first and the fourth inequalities come from (3.23), the second inequality comes from the definition (3.20) of $\mathcal{P}_{k+1}$, and the third inequality comes from $a^* \in \mathcal{P}_s$ and $x_s^* \in \arg\min_{x \in \mathcal{P}_s} \hat{\ell}_s^\top x$. Hence, (3.27) holds for $k = s+1$. By induction in k, (3.27) is proven to hold for all positive integers k. $\square$

Let T_k denote the number of rounds in the k-th phase, i.e.,

$$T_k = \sum_{i=1}^{d} \sum_{j=0}^{d} T_{kij} = t_{k+1,1,0} - t_{k,1,0}. \tag{3.28}$$

From the definition of $T_{kij} = \lceil \zeta_k \lambda_{kij} \rceil$ (Step 8 of Algorithm 6), we have $\zeta_k \lambda_{kij} \leq T_{kij} < \zeta_k \lambda_{kij} + 1$. Combining this and the condition that $\sum_{j=0}^{d} \lambda_{kij} = 1$, we obtain

$$d\zeta_k \leq T_k \leq d\zeta_k + d(d+1). \tag{3.29}$$

Let K be the index of phases such that the K-th phase includes T-th round, i.e., K is the number such that $\sum_{k=1}^{K-1} T_k < T \leq \sum_{k=1}^{K} T_k$. Note that $K \geq 2$ follows from the assumption 3.22. From (3.29) and the definition of ζ_k (Step 4 in Algorithm 6), we have

$$T > T_{K-1} \geq d\zeta_{K-1} = 2^{2K+7} d^3 \log\left(\frac{2dK(K-1)}{\delta}\right) \geq (2^{K+3})^2 d^3 \log\left(\frac{2dK(K+1)}{\delta}\right),$$

where the last inequality follows from $2\log\left(\frac{2dK(K-1)}{\delta}\right) = \log\left(\frac{2dK(K-1)}{\delta}\right)^2 \geq \log\left(\frac{2dK(K+1)}{\delta}\right)$. This inequality implies the bound on 2^K and K, as follows:

$$2^K \leq \frac{1}{2^3}\left(\frac{T}{d^3}\right)^{\frac{1}{2}} \left(\log\left(\frac{2dK(K+1)}{\delta}\right)\right)^{-\frac{1}{2}}, \tag{3.30}$$

$$K \leq \frac{1}{2}\log_2(T) - 3. \tag{3.31}$$

By means of these inequalities, we can bound the value $\sum_{t=1}^{T} \ell^{*\top}(a_t - a^*)$, for the output a_t of Algorithm 6, and $a^* \in \arg\min_{a \in \mathcal{A}} \ell^{*\top} a$:

$$
\begin{aligned}
\sum_{t=1}^{T} \ell^{*\top}(a_t - a^*) &\leq \sum_{k=1}^{K} \sum_{t=t_{k,1,0}+1}^{t_{k+1,1,0}} \ell^{*\top}(a_t - a^*) \\
&\leq \sum_{k=1}^{K} 2^{2-k} T_k L R && \text{(Lemma 3.5.2, } a_t \in \mathcal{P}_k) \\
&\leq dLR \sum_{k=1}^{K} 2^{2-k}(\zeta_k + d + 1) && \text{(from (3.29))} \\
&= dLR \sum_{k=1}^{K} \left(2^{k+11} d^2 \log\left(\frac{2dk(k+1)}{\delta} \right) + 2^{2-k}(d+1) \right) && \text{(Step 4 in Algo. 6)} \\
&\leq dLR \left(2^{K+12} d^2 \log\left(\frac{2dK(K+1)}{\delta} \right) + 4(d+1) \right) \\
&\leq dLR \left(2^9 d^2 \left(\frac{T}{d^3} \log\left(\frac{2dK(K+1)}{\delta} \right) \right)^{\frac{1}{2}} + 4(d+1) \right) && \text{(from (3.30))} \\
&\leq dLR \left(2^9 \sqrt{dT \log\left(\frac{d(\log_2 T)^2}{\delta} \right)} + 4(d+1) \right) && \text{(from (3.31))} \\
&\leq dLR \left(2^{10} \sqrt{dT \log\left(\frac{d \log T}{\delta} \right)} + 4(d+1) \right). && (3.32)
\end{aligned}
$$

By combining this and the following lemma, we obtain an upper bound on the regret $R_T(a) = \sum_{t=1}^{T} \ell_t^\top (a_t - a)$.

Lemma 3.5.3. *Let $a^* \in \arg\min_{a \in \mathcal{A}} \ell^{*\top} a$. With probability at least $1 - \delta$, it holds for all $a \in \mathcal{A}$ that*

$$
R_T(a) \leq \sum_{t=1}^{T} \ell^{*\top}(a_t - a^*) + 8LR\sqrt{dT \log\left(\frac{2d}{\delta} \right)}. \tag{3.33}
$$

Proof. We show (3.33) by proving the following two inequalities:

$$
\sum_{t=1}^{T} \ell_t^\top a_t - \sum_{t=1}^{T} \ell^{*\top} a_t \leq LR\sqrt{8T \log\left(\frac{2}{\delta} \right)}, \tag{3.34}
$$

$$
\sum_{t=1}^{T} \ell^{*\top} a^* - \sum_{t=1}^{T} \ell_t^\top a \leq LR\sqrt{8dT \log\left(\frac{2d}{\delta} \right)}. \tag{3.35}
$$

Denote $X_\tau := \sum_{t=1}^{\tau} (\ell_t - \ell^*)^\top a_t$. Since $\{X_\tau\}_{\tau=0}^{T}$ is a martingale such that $|X_{\tau+1} - X_\tau| \leq 2LR$, from Azuma's inequality, with probability $1 - \frac{\delta}{2}$, we have $X_T \leq LR\sqrt{8T \log\left(\frac{2}{\delta}\right)}$, which means that (3.34) holds. Similarly, from Hoeffding's inequality, we have

$$
\left\| \sum_{t=1}^{T} (\ell_t - \ell^*) \right\|_2 \leq L\sqrt{dT \log\left(\frac{2d}{\delta} \right)} \tag{3.36}
$$

with probability at least $1 - \frac{\delta}{2}$. Under this condition, we have

$$-\sum_{t=1}^{T} \ell_t^\top a \leq -\sum_{t=1}^{T} \ell^{*\top} a + LR\sqrt{dT \log\left(\frac{2d}{\delta}\right)} \leq -\sum_{t=1}^{T} \ell^{*\top} a^* + LR\sqrt{8dT \log\left(\frac{2d}{\delta}\right)},$$

where the first inequality follows from (3.36) and $\|a\|_2 \leq R$, and the second inequality follows from $a^* \in \arg\min_{a \in \mathcal{A}} \ell^{*\top} a$. Hence, we have (3.35) for all $a \in \mathcal{A}$ with probability at least $1 - \frac{\delta}{2}$. Since each of (3.34) and (3.35) holds with probability $1 - \frac{\delta}{2}$, both (3.34) and (3.35) hold with probability $1-\delta$. Then, by taking the sum of each side of (3.34) and (3.35), we obtain (3.33). □

By combining (3.32), Lemma 3.5.3 and (3.22), we obtain

$$R_T(a) \leq 2^{11} LR\sqrt{d^3 T \log\left(\frac{d \log T}{\delta}\right)}$$

with probability at least $1 - 2\delta$. Replacing δ with $\delta/2$, we obtain (3.21).

Chapter 4

Tight Regret Bounds for Bandit Combinatorial Optimization

Bandit combinatorial optimization is a bandit framework in which a player chooses an action in a given finite set $\mathcal{A} \subseteq \{0,1\}^d$ and suffers a loss that is the inner product of the chosen action and an unobservable loss vector in $\mathbb{R}^d$ in each round. This chapter aims to reveal what property makes the bandit combinatorial optimization hard. Recently, Cohen et al. [44] showed a lower bound $\Omega(\sqrt{dk^3T/\log T})$ of the regret, where k is the maximum ℓ_1-norm of action vectors, and T is the number of rounds. Their lower bound was constructed via continuous strongly-correlated distribution of losses. Our main result is to improve their bound to $\Omega(\sqrt{dk^3T})$ by a factor of $\sqrt{\log T}$, which can be done by means of strongly-correlated losses with *binary* values. The bound derives better regret bounds for three specific examples of bandit combinatorial optimization: the multitask bandit, the bandit ranking and the multiple-play bandit. In particular, our bound for the bandit ranking answers to an open problem posed in [44]. In addition, we show that the problem becomes easier without correlations among entries of loss vectors. In fact, if each entry of loss vectors is an independent random variable, then one can achieve a regret of $\tilde{O}(\sqrt{dk^2T})$, which is $\sqrt{k}$ times smaller than the lower bound shown above. Our results indicate that correlation among losses is essential to having a large regret.

4.1 Introduction

This chapter investigates the *bandit combinatorial optimization* problem defined as follows: A player is given a finite *action set* $\mathcal{A} \subseteq \{a \in \{0,1\}^d \mid \|a\|_1 = k\}$ and the number T of rounds for decision-making. In each round $t = 1, 2, \ldots, T$, the player chooses an *action* a_t from $\mathcal{A}$. At the same time, the environment privately chooses a *loss vector* $\ell_t = [\ell_{t1}, \ldots, \ell_{td}]^\top \in [0,1]^d$, and the player observes the loss $\ell_t^\top a_t$ incurred by the action a_t. The goal of the player is to minimize the expected cumulative loss $\mathbf{E}[\sum_{t=1}^T \ell_t^\top a_t]$, where the expectation is taken w.r.t. the player's internal randomization. The performance of her algorithm is measured in terms of the *regret* R_T defined by $R_T = \max_{a \in \mathcal{A}} \mathbf{E}\left[\sum_{t=1}^T \ell_t^\top a_t - \sum_{t=1}^T \ell_t^\top a\right]$.

Our focus is on the minimax regret, the worst-case regret attained by optimal algorithms, which can be expressed as $\mathcal{R}_T := \min_{\text{algorithm}} \max_{\{\ell_t\}_{t=1}^T \subseteq [0,1]^d} R_T$. The minimax regret can be bounded from above by designing algorithms. The current best upper bound is $\mathcal{R}_T = O(\sqrt{dk^3T\log(\mathrm{e}d/k)})$, as reported by a number of papers [13, 32, 38, 52, 79]. By way of contrast, lower bounds of the minimax regret can be proven via constructing a probabilistic distribution of loss vectors for which any algorithm suffers a certain degree of regret. To give a lower bound, Audibert et al. [13] constructed a probabilistic distribution of loss vectors for which arbitrary algorithms suffer a regret of $\Omega(\sqrt{dk^2T})$, and they conjectured that this is tight, i.e., $\mathcal{R}_T = \Theta(\sqrt{dk^2T})$. More recently, however, Cohen et al. [44] showed the lower bound of $\mathcal{R}_T = \Omega(\sqrt{dk^3T/\log T})$, which would disprove the above-mentioned conjecture, and they have decreased the gap between the upper and lower bounds to $O(\sqrt{\log(\mathrm{e}d/k)\log T})$ consisting of logarithmic terms alone.

The input distribution constructed by Cohen et al. [44] for deriving the lower bound has unique characteristics that cannot be found in previous studies, such as lower bounds for a multi-armed bandit [16], a combinatorial semi-bandit [32, 119, 160] and a combinatorial bandit [13]. In previous studies on lower bounds, only binary inputs and an arm-wise independent distribution have been considered, i.e., $\ell_{t1}, \ldots, \ell_{td}$ are mutually independent $\{0, 1\}$-valued discrete random variables. Such inputs have been shown to give tight lower bounds for multi-armed bandits [16] and combinatorial semi-bandits [13, 119]. In contrast to these, Cohen et al. [44] introduced loss vectors following a continuous distribution over $[0, 1]^d$ that have a strong correlation among d entries. Furthermore, the lower bound given by Cohen et al. [44] includes a $1/\sqrt{\log T}$ term which does not appear in the other lower bounds for bandit problems. In addition, their lower bounds apply to special cases such as the multitask bandit and the bandit ranking problem. Their results are, however, restricted to the problems under certain parameter constraints, and consequently, the tight bounds for some important special cases, including the problem referred to as *bandit ranking with full permutations*, are left open.

Such characteristics w.r.t. the input distribution given by Cohen et al. [44] lead to the following research questions:

Q. 1 Is the $1/\sqrt{\log T}$ factor in the lower bound given by Cohen et al. [44] redundant or inevitable?

Q. 2 Does a continuous distribution of loss vectors make the problem essentially harder than a discrete (binary) distribution? If we restrict to the loss vectors in $\{0, 1\}^d$, then the player can see the *number of good arms* ($i \in [d]$ s.t. $l_{ti} = 0$) in the chosen arms S_t, which may, or may not, be more informative than real values.

Q. 3 Does the correlation of loss among different arms make the problem essentially harder than arm-wise independent loss?

Q. 4 Can we obtain tight lower bounds for the special cases such as the bandit ranking problem with full permutations, resolving the open question in [44]?

4.2 Main Results

Our main result is to answer the above four questions. First, we improve the regret lower bound given by Cohen et al. [44] to $\Omega(\sqrt{dk^3T})$, by a factor of $\sqrt{\log T}$, as shown in Table 4.1. Such bounds can be proven by constructing a distribution of *strongly-correlated* losses taking *binary* values. We show the bounds for two specific examples of bandit combinatorial optimization that

Table 4.1: Regret bounds $\mathcal{R}_T$ for bandit combinatorial optimization.

Assumption	Upper bound by Algorithms	Lower bound
No assumption	$O(\sqrt{dk^2T\log\lvert\mathcal{A}\rvert})$ $= O(\sqrt{dk^3T\log(\mathrm{e}d/k)})$ ([32] and [38])	$\Omega(\sqrt{dk^3T/\log T})$ by $\ell_t \in [0,1]^d$ ([44]), $\Omega(\sqrt{dk^3T})$ by $\ell_t \in \{0,1\}^d$ **(Theorems 4.2.1** and **4.2.2)**
Independent losses	$O(\sqrt{dkT\log\lvert\mathcal{A}\rvert\log T})$ $= O(\sqrt{dk^2T\log(\mathrm{e}d/k)\log T})$ **(Algorithm 7** and **Theorem 4.2.3)**	$\Omega(\sqrt{dk^2T})$ by $\ell_t \in \{0,1\}^d$ ([13])

are of practical importance: the *multitask bandit problem*, the *bandit ranking problem* (Theorem 4.2.1), and the *multiple-play bandit problem* (Theorem 4.2.2). This result gives answers to **Q. 1** and **Q. 2** in Section 4.1: The $1/\sqrt{\log T}$ factor in the lower bound is redundant, and the difference between continuous-valued and discrete-valued losses has no large impact on the hardness of the problem. This also answers to **Q. 4**, an open problem posed in [44].

The multitask bandit problem [38, 44] is a bandit framework in which the player tries to solve k instances of the n-armed bandit problem. This is a special case of bandit combinatorial optimization with $d = kn$ and

$$\mathcal{A} = \left\{ a \in \{0,1\}^d \;\middle|\; \sum_{i=(j-1)n+1}^{jn} a_i = 1 \quad (j \in [k]) \right\}. \tag{4.1}$$

In the bandit ranking problem or online ranking problem [83] with bandit feedback problem, the goal of the player is to find a maximum matching in the complete bipartite graph $K_{k,n}$ with $d = kn$ edges, where $k \in [n]$. The set of all maximum matching can be expressed by

$$\mathcal{A} = \left\{ a \in \{0,1\}^d \;\middle|\; \sum_{i=(j-1)n+1}^{jn} a_i = 1 \ (j \in [k]),\ \sum_{i=1}^{k} a_{(i-1)n+j} \leq 1 \ (j \in [n]) \right\}. \tag{4.2}$$

For these problems, we give the following regret lower bound.

Theorem 4.2.1 (multitask bandit, bandit ranking)**.** *Suppose that $\mathcal{A}$ is defined by* (4.1) *or* (4.2) *and $n \geq 2$. There is a probability distribution D over $\{0,1\}^d$ for which the following holds: If ℓ_t is drawn from D for $t = 1, \ldots, T$ independently, the regret for any algorithm satisfies $\mathbf{E}[R_T] = \Omega(\min\{\sqrt{dk^3T}, k^{3/4}T\})$, where the expectation is taken w.r.t. the randomness of ℓ_t.*

For the bandit ranking problem, Cohen et al. [44] have shown a lower bound of $\Omega(\sqrt{\frac{dk^3T}{\log T}})$ under the assumption of $n \geq 2k$, and the full-permutation case ($k = n$) has been left as an open problem, as mentioned in their conclusion. Theorem 4.2.1 answers to this open problem: Even if $k = n$, the minimax regret is of $\mathcal{R}_T = \tilde{\Theta}(\sqrt{dk^3T}) = \tilde{\Theta}(\sqrt{k^5T})$, ignoring a $\sqrt{\log k}$ factor. Theorem 4.2.1 can also be extended to the online shortest path problem [17], via the standard reduction from multitask bandit to the online shortest path. See e.g., [44] for details of the reduction. The proof of Theorem 4.2.1 is given in Sections 4.4.3 and 4.4.5.

The multiple-play bandit problem [38, 112, 116, 160] is another bandit framework in which the player can choose arbitrary k arms from a set of d arms in each round. This problem corresponds to $\mathcal{A} = \binom{[d]}{k} := \{a \in \{0,1\}^d \mid \|a\|_1 = k\}$.

Theorem 4.2.2 (mutiple-play bandit)**.** *Suppose that* $\mathcal{A} = \binom{[d]}{k}$. *There is a probability distribution* D *over* $\{0,1\}^d$ *for which the following holds: If* ℓ_t *is drawn from* D *for* $t = 1, \ldots, T$ *independently, the regret for any algorithm will satisfy* $\mathbf{E}[R_T] = \Omega\left(\min\left\{(\frac{d-k}{d})^2\sqrt{dk^3T}, \frac{d-k}{d}k^{3/4}T\right\}\right)$, *where the expectation is taken w.r.t. the randomness of* ℓ_t.

The above lower bound means that $\mathcal{R}_T = \Omega(\sqrt{dk^3T})$ for $T = \Omega(dk^{3/2})$ and $d = \Omega(k)$. Note that existing works [13, 44, 119] gave weaker lower bounds only for the case of $d \geq 2k$, while ours are valid for general d and k. The proof of Theorem 4.2.2 is given in Section 4.4.4.

A basic idea for proving a nearly tight bound is to construct an environment where all entries of ℓ_t are strongly correlated with one another, which has been introduced by Cohen et al. [44]. If losses are strongly correlated, the observed value $\ell_t^\top a$ has a larger variance; For example, the variance is of order k if all the entries are independent, while it can be of order k^2 if all the entries take the same value. When the observed values $\ell_t^\top a$ have larger variances, the KL divergence among the values for different actions a is small, which implies that no algorithm can detect "good" actions well. Cohen et al. [44] constructed such an environment by means of normal distributions, which improve the lower bound by an $\tilde{O}(\sqrt{k})$ factor. On the other hand, unfortunately, their bound includes a redundant $(\log T)^{-1/2}$ factor due to the unbounded support of normal distributions.[1] We remark that their technique has been used recently for proving a lower bound for *bandit PCA* [113], which includes a redundant $(\log T)^{-1/2}$ factor too, for the same reason as the above.

To shave off the $(\log T)^{-1/2}$ factor, this chapter introduces a novel class of discrete distributions over $\{0,1\}^d$ so that entries of loss vectors are bounded and strongly correlated. In order to make the losses correlated, we consider d Bernoulli distributions that share the parameter, by which the observed value has a large variance of $O(k^2)$. It is, however, not straightforward to set "good" actions in this approach. The previous work [44] simply decreases the mean parameter in normal distribution to set "good" actions, but it does not work as it causes large KL divergences between good actions and the others in our distribution. In the present work, we adjust the parameter of Bernoulli distributions carefully with the intention of having small KL divergence, which successfully improves the regret lower bound. Our idea is potentially used to improve the idea of [44] even in other problems.

Second, we show that the correlation among losses is essential to having a large regret. In fact, if each entry of loss vectors is an independent random variable, then one can achieve a regret of $\tilde{O}(\sqrt{dk^2T})$ as below, which is $\sqrt{k}$ times smaller than the lower bounds in Theorems 4.2.1 and 4.2.2. This gives an answer to **Q. 3**: The correlation among losses makes the problem essentially harder, as the minimax regret bound gets larger by a factor of $\tilde{\Theta}(\sqrt{k})$.

Theorem 4.2.3 (smaller regret bound for arm-wise independent loss)**.** *There exists an algorithm that achieves* $\mathbf{E}[R_T] = O(\sqrt{dk^2T\log T\log\frac{ed}{k}})$ *for* $T = \Omega(d^3)$, *under the assumption that* ℓ_t *follows a distribution of mutually independent* d *random variables in* $[0,1]$, *i.i.d. for* $t = 1, 2, \ldots, T$.

This upper bound is nearly tight; Theorem 5 in [13] implies that any algorithm suffers $\mathbf{E}[R_T] = \Omega(\sqrt{dk^2T})$ in the worst case under the same assumption as in Theorem 4.2.3.[2] By

[1]To keep ℓ_t in the bounded region $[0,1]^d$ with high probability, the variances of normal distributions need to be kept sufficiently small, which makes the KL divergence large.

[2]Although the original statement in [13] does not include the independence assumption, we can confirm that it is satisfied in their proof.

combining this result and Theorem 4.2.3, we obtain the following corollary:

Corollary 4.2.1. *Under the same assumption as in Theorem 4.2.3, the minimax regret in bandit combinatorial optimization is of order* $\tilde{\Theta}(\sqrt{dk^2T})$*, where we ignore logarithmic factors in* d *and* T*.*

To prove Theorem 4.2.3, we analyze regret upper bounds for *stochastic linear bandits*, which are a generalization of bandit combinatorial optimization with stochastic environments. In stochastic linear bandits, a player is given a finite set $\mathcal{A} \subseteq \mathbb{R}^d$ of d-dimensional vectors. In each round, the player chooses $a_t \in \mathcal{A}$ and receives loss $L_t = \ell^{*\top} a_t + \eta_t$, where η_t is the noise, which we assume is conditionally α-subgaussian. We also assume that $\sup_{a,b\in\mathcal{A}} \ell^{*\top}(a-b) \leq L$. We observe that bandit combinatorial optimization with the assumption in Theorem 4.2.3 is a special case of stochastic linear bandits with $\alpha = \sqrt{k}/2$ and $L = k$.

For stochastic linear bandits with $\alpha = 1$ and $L = 1$, Lattimore and Szepesvári [118] provided an algorithm that achieves $R_T = O(\sqrt{dT \log \frac{|\mathcal{A}| \log T}{\delta}})$.[3] This upper bound, however, does not directly lead to Theorem 4.2.3 because their bound stands only for the case of $\alpha = 1$ and $L = 1$; If we directly apply their result, we have $R_T = O(\sqrt{dk^2T \log \frac{|\mathcal{A}| \log T}{\delta}}) = \tilde{O}(\sqrt{dk^3T})$ by multiplying losses by $1/k$. This is $\tilde{\Omega}(\sqrt{k})$ times larger than the bound in Theorem 4.2.3.

To cope with this issue, we modify their algorithm so that we can provide a more refined analysis for the case of arbitrary α and L. The differences between our Algorithm 7 given in Section 4.5.1 and Algorithm 12 in [118] are summarized as follows:

- They deal with only the case in which the noise η_t has a bounded variance, i.e., $\alpha = 1$. To deal with the case for a general α, we modify the definition (4.31) of T_k in their algorithm.
- They assume that the suboptimality gap $\max_{a,b\in\mathcal{A}}\{\ell^{*\top}(a-b)\}$ is bounded by 1. To cope well with changing suboptimality gaps, we modify the definition of ε_t in their algorithm.
- They basically consider maximization problems, while we consider minimization (This results in no essential differences).

We show that Algorithm 7 achieves the following regret bound:

Theorem 4.2.4. *For any input parameters* $\delta > 0$ *and* $\varepsilon_1 > 0$*, with a probability of at least* $1 - \delta$*, the output of Algorithm 7 satisfies*

$$\max_{a\in\mathcal{A}} \sum_{t=1}^{T} \ell^{*\top}(a_t - a) \leq 9\alpha\sqrt{dT \log \frac{|\mathcal{A}| \log T}{\delta}} + L\frac{2d\alpha^2}{\varepsilon_1^2} \log \frac{2|\mathcal{A}|}{\delta} + (L + \varepsilon_1)d^2. \tag{4.3}$$

Theorem 4.2.4 means that the upper bound L of $\ell^{*\top} a_t$ does not affect the leading term of the regret upper bound, but α does. By substituting $\alpha = \sqrt{k}/2$ and $L = k$ into the bound in Theorem 4.2.4, we obtain Theorem 4.2.3.

4.3 Related Work

Bandit combinatorial optimization was introduced by McMahan and Blum [131] and Awerbuch and Kleinberg [17]. They proposed algorithms achieving regret of $\tilde{O}(T^{3/4})$ and $\tilde{O}(T^{2/3})$,

[3] In their book, the proof is left for the reader as an exercise.

respectively, ignoring dependence on d and logarithmic factors in T. Algorithms with better regret bounds have been proposed in several papers [13, 32, 38, 52]. These algorithms achieve $R_T = O(\sqrt{dk^3 T \log(ed/k)})$ in our problem setting. Recently, computationally efficient algorithms achieving sublinear regret have also been considered in, e.g., [38, 45, 79, 148].

In terms of lower bounds for bandit combinatorial optimization, Audibert et al. [13] showed that $R_T = \Omega(\sqrt{dk^2 T})$, and they conjectured that this lower bound was tight. Very recent work by Cohen et al. [44], however, disproved the conjecture showing $R_T = \Omega(\sqrt{dk^3 T / \log T})$.

Combinatorial semi-bandit optimization is a variant of bandit combinatorial optimization, in which the player can observe not only the total loss $\ell_t^\top a_t$ but also the entry ℓ_{ti} for each chosen arm $i \in S_t$. This problem was introduced by György et al. [71] in the context of the online shortest path problem, i.e., they considered the case in which $\mathcal{A}$ is a set of all subsets of edges constructing a path in a given graph. For general action sets $\mathcal{A} \subseteq \binom{[d]}{k}$, Audibert et al. [13] proposed an algorithm achieving regret of $O(\sqrt{dkT})$, and showed that this is minimax optimal, i.e., there is an action set $\mathcal{A} \subseteq \binom{[d]}{k}$ such that $R_T = \Omega(\sqrt{dkT})$. For the multiple-play bandit problem, i.e., the case of $\mathcal{A} = \binom{[d]}{k}$, with semi-bandit feedback, Uchiya et al. [160] showed that $R_T = \Omega(\sqrt{dT})$, but it remained open whether this is tight, until very recent work by Lattimore et al. [119] provided the proof that $R_T = \Omega(\sqrt{dkT})$.

The study of stochastic linear bandits dates back to work by Abe and Long [2]. They and Auer [14] considered the case of finite action sets that can change every round. Bandit combinatorial optimization with a stochastic environment can be seen as a special case of stochastic linear bandits in which the action set is included in $\binom{[d]}{k}$ and does not change every round. Auer [14] introduced a technique of dividing rounds to achieve $R_T = O(\sqrt{dT(\log(|\mathcal{A}| T \log T))^3})$, under the assumption of bounded loss. We remark that a similar technique is used in Algorithm 7. A similar technique was used for spectral bandits given by Valko et al. [161], in which they eliminated unpromising arms over several phases.

4.4 Lower Bounds

In this section, we give proofs for Theorems 4.2.1 and 4.2.2. First, we revisit proofs in the previous work, Theorem 5 in [13] and Lemma 4 in [44], which show regret lower bounds of the order $\Omega(\sqrt{dk^2 T})$ and $\Omega(\sqrt{dk^3 T / \log T})$ for multitask bandits, respectively. From their proofs, we can observe that regret lower bounds can be derived from upper bounds on KL divergences that are determined by distributions of loss vectors. Second, we construct a distribution of loss vectors so that the corresponding KL divergence is small enough. Combining these two results, we obtain Theorem 4.2.1, an improved lower bound for multitask bandits. Finally, we extend the proof for multitask bandit to prove Theorem 4.2.2 for multiple-play bandits.

4.4.1 Proof idea in previous work

This subsection revisits the proofs for regret lower bounds for multitask bandit, given in [13] and [44]. We note that, from Yao's minimax principle, it suffices to construct a probabilistic distribution of ℓ_t such that any deterministic algorithm suffers large regret in expectations.

In both of the proofs, the probabilistic distribution of the loss vectors is defined as follows. Set a parameter $\varepsilon > 0$, which is to be optimized later. For $a^* = [a_1^*, \ldots, a_d^*]^\top \in \{0,1\}^d$, a probabilistic distribution D_{a^*} over $\mathbb{R}^d$ is defined so that $\ell \sim D_{a^*}$ satisfies

$$\mathop{\mathbf{E}}_{\ell \sim D_{a^*}}[\ell_i] = \frac{1}{2} - \varepsilon a_i^* \tag{4.4}$$

for each $i \in [d]$. More concretely, [13] define D_{a^*} such that the i-th entry of the vector follows a Bernoulli distribution of parameter $\frac{1}{2} - \varepsilon a_i^*$, independently. Cohen et al. [44] define D_{a^*} such that the i-th entry is equal to $\frac{1}{2} - \varepsilon a_i^* + Z$ where Z follows the normal distribution $N(0, \sigma^2)$. We can confirm that these two definitions satisfy (4.4). The environment picks $a^* \in \mathcal{A}$ uniformly at random before the game begins, and then in round $t = 1, 2, \ldots, T$, generates a loss vector ℓ_t following D_{a^*} i.i.d. Recall that $\mathcal{A}$ is defined by (4.1) here.

We analyze the regret bounds for these loss vectors. Let $S^* = \{i \in [d] \mid a_i^* = 1\}$, and a_t be the action chosen by the player in round t. Let us define N_i to be the number of rounds in $[T]$ in which the player suffers a loss for the i-th entry of loss vectors, i.e., $N_i = |\{t \in [T] \mid a_{ti} = 1\}|$. Then, from (4.4), the regret R_T satisfies

$$\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T \sim D_{a^*}}[R_T] \geq \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T \sim D_{a^*}}\left[\sum_{t=1}^{T} \ell_t^\top a_t - \sum_{t=1}^{T} \ell_t^\top a^*\right] = \varepsilon \left(kT - \sum_{i \in S^*} \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T \sim D_{a^*}}[N_i]\right). \tag{4.5}$$

From (4.5), in order to obtain a lower bound on R_T, it suffices to bound $\sum_{i \in S^*} N_i$. To obtain a bound on N_i, we use the following lemma:

Lemma 4.4.1. *Let D and D' be probability distributions over $[0,1]^d$. We then have*

$$\left| \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T \sim D}[N_i] - \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T \sim D'}[N_i] \right| \leq T \sqrt{\sum_{t=1}^{T} \mathop{\mathbf{E}}_{a_t \sim A_t(D)} \left[\mathop{D_{\mathrm{KL}}}_{\ell \sim D, \ell' \sim D'} (a_t^\top \ell || a_t^\top \ell') \right]} \tag{4.6}$$

for any deterministic algorithm, where $A_t(D)$ represents the probability distribution of outputs from the algorithm in round t for the inputs $\ell_1, \ell_2, \ldots, \ell_{t-1}$ following D independently.

This lemma follows from Pinsker's inequality and the chain rule for the KL divergence. For details, see, e.g., Lemma A.1. in [16].

Lemma 4.4.1 gives a connection between bounds on N_i and upper bounds on KL divergences of specific distributions. To give a bound on N_i by means of Lemma 4.4.1, Audibert et al. [13] and Cohen et al. [44] used specific properties of their distributions. We observe that the key in their arguments is the fact that their distributions of loss vectors satisfy the following condition regarding the KL divergence:

$$\begin{aligned} & a \in \mathcal{A},\ a^*, \hat{a} \in \{0,1\}^d,\ \hat{a}^\top a - a^{*\top} a = 1,\ \ell^* \sim D_{a^*}, \hat{\ell} \sim D_{\hat{a}} \\ \Longrightarrow \quad & D_{\mathrm{KL}}(\ell^{*\top} a || \hat{\ell}^\top a) \leq C_D \varepsilon^2 \text{ for a constant } C_D \text{ depending on } \{D_a\}. \end{aligned} \tag{4.7}$$

Intuitively, the precondition of (4.7) means that the discrepancy w.r.t. the expected loss is at most ε. In fact, $a^{*\top} a$ in (4.7) corresponds to "goodness of action a" for the loss vector $\ell \sim D_{a^*}$ because the expected loss for action a is equal to $k/2 - \varepsilon a^{*\top} a$ from (4.4). Thus $\hat{a}^\top a - a^{*\top} a = 1$ means that the expected loss for D_{a^*} is smaller than that for $D_{\hat{a}}$ by ε.

We can show that, if (4.7) is true, then Lemma 4.4.1 implies that, if a^* follows a uniform distribution over $\mathcal{A}$ defined by (4.1), we have $\mathop{\mathbf{E}}_{a^*, \ell_1,\ldots,\ell_T \sim D_{a^*}} \left[\sum_{i \in S^*} N_i\right] \leq k \left(\frac{T}{2} + T\varepsilon \sqrt{\frac{kT}{d} C_D} \right)$.

Hence, if we set $\varepsilon \leq \sqrt{d/(16C_D kT)}$, we have $\underset{a^*,\ell_1,\ldots,\ell_T \sim D_{a^*}}{\mathbf{E}}\left[\sum_{i \in S^*} N_i\right] \leq 3kT/4$, and consequently, we obtain $\mathbf{E}[R_T] \geq \frac{\varepsilon kT}{4}$ from (4.5). A key observation in this subsection is summarized as follows:

Observation 4.4.1. *Suppose a family* $\{D_{a^*} \mid a^* \in \{0,1\}^d\}$ *of distributions with a parameter* $\varepsilon \leq \sqrt{d/(16C_D kT)}$ *satisfies* (4.4) *and* (4.7)*. If* $\mathcal{A}$ *is given by* (4.1) *with* $n \geq 2$*, then we have a regret lower bound of* $\mathbf{E}[R_T] = \Omega(\varepsilon kT)$*.*

4.4.2 Construction of probabilistic distribution

The goal of this subsection is to construct a family $\{D_{a^*} \mid a^* \in \{0,1\}^d\}$ of distributions such that (4.4) and (4.7) are satisfied with $C_D = O(1/k^2)$. From Observation 4.4.1, such a construction leads to a regret lower bound of $\mathbf{E}[R_T] = \Omega(\sqrt{dk^3T})$ for the multitask bandit problem, proving Theorem 4.2.1.

Our probabilistic distribution of loss vectors is defined as follows. Set a parameter $\varepsilon \in [0, 2^{-16}]$, which is to be optimized later. For $a^* = [a^*_1, \ldots, a^*_d]^\top \in \{0,1\}^d$, let D_{a^*} be a distribution of $\ell = [\ell_1, \ldots, \ell_d]^\top \in [0,1]^d$ generated in the following way:

(i) Draw u_0 from a uniform distribution over $[0,1]$. (4.8)

(ii) Draw b_i from a Bernoulli distribution of parameter $(1/2 + 2\varepsilon a^*_i)$.

(iii) For $i \in [d]$, draw u_i from a uniform distribution over $\begin{cases} [0,1/2] & \text{if } b_i = 1, \\ (1/2, 1] & \text{if } b_i = 0. \end{cases}$

(iv) Let $\ell_i = 1$ if $u_i \geq u_0$, and otherwise, $\ell_i = 0$.

We can confirm that (4.4) holds for this D_{a^*}. In fact, step (iv) means $\mathbf{E}[\ell_i] = \mathrm{Prob}[u_i \geq u_0]$, and since u_0 follows the uniform distribution over $[0,1]$ and $u_i \in [0,1]$, we have $\mathrm{Prob}[u_i \geq u_0] = \mathbf{E}[u_i]$. Moreover, from steps (ii) and (iii), we have $\mathbf{E}[u_i] = \frac{1}{4}\mathrm{Prob}[b_i = 1] + \frac{3}{4}\mathrm{Prob}[b_i = 0] = \frac{1}{4}(\frac{1}{2} + 2\varepsilon a^*_i) + \frac{3}{4}(\frac{1}{2} - 2\varepsilon a^*_i) = \frac{1}{2} - \varepsilon a^*_i$, which means that (4.4) holds.

Let us show that (4.7) is satisfied with $C_D = O(1/k^2)$. Since D_{a^*} is a distribution over $\{0,1\}^d$, $\ell^{*\top} a$ takes values over $\{0,1,\ldots,k\}$ for any $a \in \mathcal{A}$. For $i = 0,1,\ldots,k$, define $P(i) = \mathrm{Prob}[\ell^{*\top} a = i]$ and $P'(i) = \mathrm{Prob}[\hat{\ell}^\top a = i]$. Then, from the definition, the KL divergence can be expressed as follows:

$$\begin{aligned} D_{\mathrm{KL}}(\ell^{*\top} a \| \hat{\ell}^\top a) &= -\sum_{i=0}^{k} P(i) \log \frac{P'(i)}{P(i)} = -\sum_{i=0}^{k} P(i) \log \left(1 + \frac{P'(i) - P(i)}{P(i)}\right) \\ &\leq -\sum_{i=0}^{k} P(i) \left(\frac{P'(i) - P(i)}{P(i)} - 2\left(\frac{P'(i) - P(i)}{P(i)}\right)^2\right) = 2\sum_{i=0}^{k} \frac{(P'(i) - P(i))^2}{P(i)}, \end{aligned}$$

where the inequality comes from the facts that $\log(1+x) \geq x - 2x^2$ for $|x| \leq 1/2$ and $|P'(i) - P(i)|/P(i) \leq 1/2$ holds.[4] Thus, it suffices to bound $(P'(i) - P(i))^2/P(i)$ for deriving an upper bound on the KL divergence. We can then show that $P(i) = \Omega(1/k)$ for all i. Indeed, if $\varepsilon = 0$, then we have $P(i) = 1/(k+1)$; since $\mathrm{Prob}[\ell_{ti} = 1] = \mathrm{Prob}[u_i \geq u_0]$ from the definition (4.8) of

[4]The fact $|P'(i) - P(i)|/P(i) \leq 1/2$ comes from $\varepsilon \leq 2^{-16}$. See the proof of Lemma 4.4.2 for details.

D_a, and since each u_i is a uniform random variable over $[0,1]$ under the condition of $\varepsilon = 0$, we have

$$P(i) = \mathrm{Prob}\left[\sum_{j=1}^{k} \ell_{tj} = i\right] = \mathrm{Prob}\left[\ u_0 \text{ is the } (i+1)\text{-th smallest among } \{u_j\}_{j=0}^{k}\ \right] = \frac{1}{k+1},$$

where the last equality comes from the fact that $u_0, u_1, \ldots, u_k$ are i.i.d. random variables. Even if $\varepsilon > 0$, we show in the proof of Lemma 4.4.2 that, for $\varepsilon \leq 2^{-16}$, $P(i)$ is sufficiently close to $\frac{1}{k+1}$ to have an order of $\Omega(1/k)$. Thus, we have $P(i) = \Omega(1/k)$ for all $i = 1, \ldots, k$ and $\varepsilon \in [0, 2^{-16}]$, and hence $D_{\mathrm{KL}}(\ell^{*\top} a || \hat{\ell}^\top a) = O\left(k \sum_{i=0}^{k} (P'(i) - P(i))^2\right)$. Finally, by proving $|P'(i) - P(i)| = O(\varepsilon/k^2)$, we obtain the following lemma:

Lemma 4.4.2. *Let $a^*, \hat{a} \in \{0,1\}^d$ and $\ell^* \sim D_{a^*}, \hat{\ell} \sim D_{\hat{a}}$. Then, for $\varepsilon \in [0, 2^{-16}]$ and $a \in \{0,1\}^d$ satisfying $\|a\|_1 = k$ and $\hat{a}^\top a - \hat{a}^{*\top} a = 1$, we have*

$$D_{\mathrm{KL}}(\ell^{*\top} a || \hat{\ell}^\top a) = O\left(\frac{\varepsilon^2}{k^2} + \frac{\varepsilon^4}{k^{3/2}}\right). \tag{4.9}$$

Proof. Let $\chi(S) \in \{0,1\}^d$ denote the indicator vector of subset $S \subseteq [d]$. Without loss of generality, we suppose that $a = \chi([k])$, $\hat{a} = \chi([s])$, and $a^* = \chi([s] \cup \{k\})$ for some $s \in \{0, 1, \ldots, k-1\}$. We then have $\hat{\ell}^\top a = \sum_{i=1}^{k} \ell_i$ and $\ell^{*\top} a = \sum_{i=1}^{k} \ell'_i$. Note that $\hat{\ell}^\top a$ and $\ell^{*\top} a$ take values in $\{0, 1, \ldots, k\}$. Let us denote

$$P(i) = \mathrm{Prob}[\hat{\ell}^\top a = i], \quad P'(i) = \mathrm{Prob}[\ell^{*\top} a = i] \tag{4.10}$$

for $i = 0, 1, \ldots, k$. For $j = 1, 2, \ldots, k$, we denote $B_j = \sum_{i=1}^{j} b_i$ and $B'_j = \sum_{i=1}^{j} b'_i$, where b_i and b'_i stand for the values b_i in (4.8) for generating ℓ^* and $\hat{\ell}$, respectively. Let us denote

$$Q_j(i) = \mathrm{Prob}[B_j = i], \quad Q'_j(i) = \mathrm{Prob}[B'_j = i] \tag{4.11}$$

for $i = 0, 1, \ldots, j$. Let us consider conditional probability of $\ell^{*\top} a$ given B_k and u_0 in (4.8). Given B_k and $u_0 \in [0, \frac{1}{2}]$, $\ell^{*\top} a$ follows a uniform distribution over $\{B_k, B_k + 1, \ldots, k\}$. Indeed, we have $\ell^*_j = 1$ for each $j \in I_{\mathrm{p}} := \{j' \in [k] \mid b_{j'} = 1\}$ since we have $u_j \geq \frac{1}{2} \geq u_0$. Since u_j for $j \in I_{\mathrm{n}} := \{j' \in [k] \mid b_{j'} = 0\}$ and u_0 follows a uniform distribution over $[0, \frac{1}{2}]$ independently, $\sum_{j \in I_{\mathrm{n}}} \ell^*_j$, the number of $j \in I_n$ with u_j lager than u_0, follows a uniform distribution over $\{0, 1, \ldots, |I_{\mathrm{n}}|\}$. Since we have $|I_{\mathrm{p}}| = B_k$ and $|I_{\mathrm{n}}| = k - B_k$, it holds that $\ell^{*\top} a = \sum_{j \in I_p} \ell^*_j + \sum_{j \in I_n} \ell^*_j = B_k + \sum_{j \in I_n} \ell^*_j$ follows a uniform distribution over $\{B_k, B_k + 1, \ldots, k\}$, given B_k and $u_0 \in [0, \frac{1}{2}]$. Similarly, given B_k and $u_0 \in [\frac{1}{2}, 1]$, $\ell^{*\top} a$ follows a uniform distribution over $\{0, 1, \ldots, B_k\}$. Hence, $P(i)$ can be expressed as

$$\begin{aligned} P(i) &= Q_k(0)\frac{1}{2} \cdot \frac{1}{k+1} + Q_k(1)\frac{1}{2} \cdot \frac{1}{k} + \cdots + Q_k(i-1)\frac{1}{2} \cdot \frac{1}{k-i+2} + Q_k(i)\frac{1}{2} \cdot \frac{1}{k-i+1} \\ &\quad + Q_k(i)\frac{1}{2} \cdot \frac{1}{i+1} + Q_k(i+1)\frac{1}{2} \cdot \frac{1}{i+2} + \cdots + Q_k(k)\frac{1}{2} \cdot \frac{1}{k+1} \\ &= \frac{1}{2}\sum_{j=0}^{i} \frac{Q_k(j)}{k-j+1} + \frac{1}{2}\sum_{j=i}^{k} \frac{Q_k(j)}{j+1} \end{aligned} \tag{4.12}$$

for each $i = 0, 1, \ldots, k$. Similarly, we have

$$P'(i) = \frac{1}{2}\sum_{j=0}^{i} \frac{Q'_k(j)}{k-j+1} + \frac{1}{2}\sum_{j=i}^{k} \frac{Q'_k(j)}{j+1}. \tag{4.13}$$

Hence, we have

$$P(i) - P'(i) = \frac{1}{2}\sum_{j=0}^{i} \frac{Q_k(j) - Q'_k(j)}{k-j+1} + \frac{1}{2}\sum_{j=i}^{k} \frac{Q_k(j) - Q'_k(j)}{j+1}. \tag{4.14}$$

From the assumption that $\hat{a} = \chi([s])$, and $a^* = \chi([s] \cup \{k\})$ for $s \leq k-1$, we have $Q_{k-1}(j) = Q'_{k-1}(j)$ and $Q_k(j) = \frac{1}{2}Q_{k-1}(j) + \frac{1}{2}Q_{k-1}(j-1)$, $Q'_k(j) = (\frac{1}{2}+2\varepsilon)Q_{k-1}(j) + (\frac{1}{2}-2\varepsilon)Q_{k-1}(j-1)$, and hence, we have $Q_k(j) - Q'_k(j) = 2\varepsilon(Q_{k-1}(j-1) - Q_{k-1}(j))$. By substituting this into (4.14), we obtain

$$\begin{aligned}
P(i) - P'(i) &= \varepsilon \sum_{j=0}^{i} \frac{Q_{k-1}(j-1) - Q_{k-1}(j)}{k-j+1} + \varepsilon \sum_{j=i}^{k} \frac{Q_{k-1}(j-1) - Q_{k-1}(j)}{j+1} \\
&= \varepsilon \left(\sum_{j=0}^{i-1} Q_{k-1}(j) \left(\frac{1}{k-j} - \frac{1}{k-j+1} \right) - \frac{Q_{k-1}(i)}{k-i+1} \right. \\
&\qquad \left. + \frac{Q_{k-1}(i-1)}{i+1} + \sum_{j=i}^{k-1} Q_{k-1}(j) \left(\frac{1}{j+2} - \frac{1}{j+1} \right) \right) \\
&= \varepsilon \left(\frac{Q_{k-1}(i-1)}{i+1} - \frac{Q_{k-1}(i)}{k-i+1} + \sum_{j=0}^{i-1} \frac{Q_{k-1}(j)}{(k-j)(k-j+1)} - \sum_{j=i}^{k-1} \frac{Q_{k-1}(j)}{(j+1)(j+2)} \right).
\end{aligned}$$

The last term $\sum_{j=i}^{k-1} \frac{Q_{k-1}(j)}{(j+1)(j+2)}$ can be bounded as follows:

$$\begin{aligned}
&\sum_{j=i}^{k-1} \frac{Q_{k-1}(j)}{(j+1)(j+2)} \leq \sum_{j=0}^{k-1} \frac{Q_{k-1}(j)}{(j+1)(j+2)} \leq \operatorname{Prob}\left[B_{k-1} < \left\lfloor \frac{k}{4} \right\rfloor \right] + \sum_{j=\lfloor k/4 \rfloor}^{k-1} \frac{Q_{k-1}(j)}{(j+1)(j+2)} \\
&\leq \operatorname{Prob}\left[B_{k-1} - \mathbf{E}[B_{k-1}] \leq -\frac{k}{8} \right] + \sum_{j=\lfloor k/4 \rfloor}^{k-1} \frac{Q_{k-1}(j)}{(k/4)(k/4+1)} \leq \exp(-\frac{k}{32}) + \frac{16}{k^2} \leq \frac{2^{11}}{k^2},
\end{aligned}$$

where the third inequality comes from the fact that $\mathbf{E}[B_{t-1}] \geq \frac{3(k-1)}{8}$ and the fourth inequality comes from Hoeffding's inequality. In a similar way, we can show that $\sum_{j=0}^{i-1} \frac{Q_{k-1}(j)}{(k-j)(k-j+1)} \leq \frac{2^{11}}{k^2}$. Hence, we have

$$|P(i) - P'(i)| \leq \varepsilon \left(\left| \frac{Q_{k-1}(i-1)}{i+1} - \frac{Q_{k-1}(i)}{k-i+1} \right| + \frac{2^{12}}{k^2} \right). \tag{4.15}$$

Next, we show $\left| \frac{Q_{k-1}(i-1)}{i+1} - \frac{Q_{k-1}(i)}{k-i+1} \right| = O(1/k^2 + \varepsilon/k)$ via showing that $R_j(i) := \frac{Q_j(i)}{Q_j(i-1)} \approx \frac{j+1-i}{i}$. Define $\beta = \frac{1+4\varepsilon}{1-4\varepsilon}$. We show it holds for all $j = 1, 2, \ldots, k-1$ and $i = 1, \ldots, j$, that

$$\frac{1}{\beta}\frac{j+1-i}{i} \leq R_j(i) \leq \frac{j+1-i}{i}, \tag{4.16}$$

by induction in j. For $j = 1$, (4.16) clearly holds. Since Q_j corresponds to the probability distribution of $\sum_{i=1}^{j} b_i$ and b_i for $k \leq k-1$ follows a Bernoulli distribution as defined in (4.8) with $a^* = \chi([s])$, R_{j+1} can be expressed as

$$R_{j+1}(i) = \frac{Q_{j+1}(i+1)}{Q_{j+1}(i)} = \frac{\alpha_j Q_j(i+1) + Q_j(i)}{\alpha_j Q_j(i) + Q_j(i-1)} = \frac{1+\alpha_j R_j(i)}{\alpha_j + 1/R_j(i-1)} = \frac{1}{\alpha_j}\frac{1+\alpha_j R_j(i)}{1+1/(\alpha_j R_j(i-1))}$$

for $j = 1, 2, \dots, k-2$, where $\alpha_j = \beta$ for $j \leq s-1$ and $\alpha_j = 1$ otherwise. Assuming (4.16) holds for $j = j'$ and $\alpha_{j'} = \beta$, we obtain

$$R_{j'+1}(i) = \frac{1}{\beta}\frac{1+\beta R_{j'}(i)}{1+1/(\beta R_{j'}(i-1))} \leq \frac{1}{\beta}\frac{1+\beta\frac{j'+1-i}{i}}{1+\frac{1}{\beta}\frac{i-1}{j'+2-i}} = \frac{i+\beta(j'+1-i)}{i-1+\beta(j'+2-i)}\frac{j'+2-i}{i} \leq \frac{j'+2-i}{i}$$
$$R_{j'+1}(i) = \frac{1}{\beta}\frac{1+\beta R_{j'}(i)}{1+1/(\beta R_{j'}(i-1))} \geq \frac{1}{\beta}\frac{1+\frac{j'+1-i}{i}}{1+\frac{i-1}{j'+2-i}} = \frac{1}{\beta}\frac{j'+2-i}{i},$$

which means (4.16) also holds for incremented $j = j'+1$. Similarly in the case of $\alpha_{j'} = 1$, we can show that (4.16) for $j = j'$ implies that (4.16) holds for $j = j'+1$. Hence, (4.16) holds for all $j \in \{1, 2, \dots, k-1\}$ and $i \in \{0, 1, \dots, i-1\}$. As a result, we have

$$\begin{aligned}&\left|\frac{Q_{k-1}(i-1)}{i+1} - \frac{Q_{k-1}(i)}{k-i+1}\right| = Q_{k-1}(i-1)\left|\frac{1}{i+1} - \frac{R_{k-1}(i-1)}{k-i+1}\right| \\ &\leq Q_{k-1}(i-1)\left(\left|\frac{1}{i+1} - \frac{k-i-1}{(i-1)(k-i+1)}\right| + \left|\left(1-\frac{1}{\beta}\right)\frac{k-i-1}{(i-1)(k-i+1)}\right|\right) \\ &\leq Q_{k-1}(i-1)\left(\frac{2k}{(i+1)(i-1)(k-i+1)} + \frac{8\varepsilon}{i-1}\right)\end{aligned}$$

From Hoeffding's inequality, for $i < \lfloor k/4 \rfloor$, $Q_{k-1}(i-1) \leq \exp(-k^2/32) \leq 2^{10}/k^2$. For $i \geq \lfloor k/4 \rfloor$, $\frac{2k}{(i+1)(i-1)(k-i+1)} \leq 2^{10}/k^2$. Hence, the right-most-hand side above is bounded by $\frac{2^{10}}{k^2} + \frac{8\varepsilon Q_{k-1}(i-1)}{i-1}$. From this and (4.15), we have

$$|P(i) - P'(i)| \leq \varepsilon\left(\frac{2^{13}}{k^2} + \frac{8\varepsilon Q_{k-1}(i-1)}{i-1}\right) \tag{4.17}$$

Further, from (4.12), we have

$$P(i) = \frac{1}{2}\sum_{j=0}^{i}\frac{Q_k(j)}{k-j+1} + \frac{1}{2}\sum_{j=i}^{k}\frac{Q_k(j)}{j+1} \geq \frac{1}{2}\sum_{j=0}^{k}\frac{Q_k(j)}{k+1} = \frac{1}{2(k+1)} \tag{4.18}$$

From (4.17) and (4.18), $|P(i) - P'(i)|/P(i) \leq 1/2$ for $\varepsilon \leq 2^{-16}$. The KL divergence between P

and P' is bounded as

$$\begin{aligned}
D_{\mathrm{KL}}(\ell^{*\top} a || \hat{\ell}^\top a) &= -\sum_{i=0}^{k} P(i) \log\left(\frac{P'(i)}{P(i)}\right) = -\sum_{i=0}^{k} P(i) \log\left(1 + \frac{P'(i) - P(i)}{P(i)}\right) \\
&\leq -\sum_{i=0}^{k} P(i) \left(\frac{P'(i) - P(i)}{P(i)} - 2\left(\frac{P'(i) - P(i)}{P(i)}\right)^2\right) \\
&\leq 2\sum_{i=0}^{k} \frac{(P'(i) - P(i))^2}{P(i)} \\
&\leq 4(k+1)\varepsilon^2 \sum_{i=0}^{k} \left(\frac{2^{13}}{k^2} + \frac{8\varepsilon Q_{k-1}(i-1)}{i-1}\right)^2 \\
&\leq 8(k+1)\varepsilon^2 \left(\sum_{i=0}^{k} \left(\frac{2^{13}}{k^2}\right)^2 + \sum_{i=0}^{k} \left(\frac{8\varepsilon Q_{k-1}(i-1)}{i-1}\right)^2\right) \\
&\leq 8(k+1)\varepsilon^2 \left(\frac{2^{26}(k+1)}{k^4} + \frac{2^{16}\varepsilon^2}{k^{5/2}}\right) \leq \frac{2^{51}\varepsilon^2}{k^2} + \frac{2^{16}\varepsilon^4}{k^{3/2}},
\end{aligned}$$

where the first inequality comes from $\log(1+x) \geq x - 2x^2$ for $|x| \leq 1/2$, the second inequality comes from $\sum P(i) = \sum P'(i) = 1$, the third inequality comes from (4.17) and (4.18), the forth inequality comes from a standard inequality of $(x+y)^2 \leq 2(x^2 + y^2)$, and the fifth inequality comes from Hoeffding's inequality and $Q_{k-1}(i-1) \leq 2^{10} k^{-1/2}$. □

4.4.3 Lower Bound for the Multitask Bandit Problem

We obtain an improved lower bound for $\mathcal{A}$ defined as (4.1), by combining Observation 4.4.1 and Lemma 4.4.2.

From Lemma 4.4.2, if $\varepsilon \leq 2^{-16} k^{-\frac{1}{4}}$, there is a global constant C for which (4.7) holds with $C_D = (C/k)^2$. Hence, setting $\varepsilon = \min\{2^{-16} k^{-\frac{1}{4}}, \frac{1}{4C}\sqrt{\frac{dk}{T}}\}$, we have $\mathbf{E}[R_T] = \Omega(\varepsilon k T) = \Omega(\min\{k^{\frac{3}{4}} T, \sqrt{dk^3 T}\})$, which provides the lower bound in Theorem 4.2.1, for $\mathcal{A}$ given by (4.1), i.e., the multitask bandit problem. A key point for shaving off the $\sqrt{\log T}$ factor is that our probabilistic distribution constructed in Section 4.4.2 satisfies (4.7) with $C_D = O(1/k^2)$, while the previous work [44] does not exceed $C_D = O(\log T / k^2)$.

4.4.4 Lower Bound for the Multiple-Play Bandit Problem

For the multiple-play bandit problem, i.e., for $\mathcal{A} = \binom{[d]}{k}$, Observation 4.4.1 does not directly derive a regret lower bound. In this subsection, we extend the observation to multiple-play bandit problems.

Proof of Theorem 4.2.2. Let $U(\Sigma_d)$ denote a uniform distribution over all permutations of $[d]$. For a permutation $\sigma : [d] \to [d]$, let $\sigma([i])$ denote the element of $\{0,1\}^d$ such that the $\sigma(j)$-th component is 1 if $j \in [i]$, and 0 otherwise; i.e., $\sigma([i])$ is the indicator vector of $\{\sigma(j) \mid j \in [i]\}$. If $\sigma \sim U(\Sigma_d)$, then $\sigma([k])$ follows a uniform distribution $U(\mathcal{A})$ over $\mathcal{A} = \binom{[d]}{k}$. We hence have

$\mathop{\mathbf{E}}_{a^*\sim U(\mathcal{A})}\left[\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{a^*}}[R_T]\right] = \mathop{\mathbf{E}}_{\sigma\sim U(\Sigma_d)}\left[\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[R_T]\right]$. Let us define $M(k,i)\in\mathbb{R}$ by

$$M(k,i) = \mathop{\mathbf{E}}_{\sigma\sim U(\Sigma_d)}\left[\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[N_{\sigma(i)}]\right].$$

We then have $M(k,1) = M(k,2) = \cdots = M(k,k)$ and $M(k,k+1) = M(k,k+2) = \cdots = M(k,d)$. From (4.5), the expectation of the regret can be expressed as

$$\mathop{\mathbf{E}}_{\sigma\sim U(\Sigma_d)}\left[\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[R_T]\right] = \varepsilon\left(kT - \sum_{i=1}^{k} M(k,i)\right) = \varepsilon(kT - kM(k,k)) = \varepsilon k(T - M(k,k)). \tag{4.19}$$

Let us evaluate $M(k,k)$ with the difference between $M(k,k)$ and $M(k-1,k)$. From Lemma 4.4.1, we have

$$\left|\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}] - \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[N_{\sigma(k)}]\right| \\ \le T\sqrt{\sum_{t=1}^{T}\mathop{\mathbf{E}}_{a_t\sim A_t(D_{\sigma([k-1])})}\left[\mathop{D_{\mathrm{KL}}}_{\ell\sim D_{\sigma([k-1])},\ell'\sim D_{\sigma([k])}}(a_t^\top\ell\|a_t^\top\ell')\right]}. \tag{4.20}$$

Let us consider $D_{\mathrm{KL}}(a_t^\top\ell\|a_t^\top\ell')$ for $\ell\sim D_{\sigma([k-1])}$, $\ell'\sim D_{\sigma([k])}$, fixed $\sigma\in\Sigma_d$, and fixed $a_t\in\mathcal{A}$. If $a_{t,\sigma(k)}=1$, we have $D_{\mathrm{KL}}(a_t^\top\ell\|a_t^\top\ell') = O(\frac{\varepsilon^2}{k^2}+\frac{\varepsilon^4}{k^{3/2}})$ from Lemma 4.4.2. Otherwise, we have $D_{\mathrm{KL}}(a_t^\top\ell\|a_t^\top\ell')=0$ because the probabilistic distribution of $a_t^\top\ell$ is equal to that of $a_t^\top\ell'$. Hence,

$$\sum_{t=1}^{T}\mathop{\mathbf{E}}_{a_t\sim A_t(D_{\sigma([k-1])})}\left[\mathop{D_{\mathrm{KL}}}_{\ell\sim D_{\sigma([k-1])},\ell'\sim D_{\sigma([k])}}(a_t^\top\ell\|a_t^\top\ell')\right] \\ \le \sum_{t=1}^{T}\mathop{\mathrm{Prob}}_{a_t\sim A_t(D_{\sigma([k-1])})}[a_{t,\sigma(k)}=1]\left(\frac{C\varepsilon}{k}\right)^2 = \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}]\left(\frac{C\varepsilon}{k}\right)^2.$$

From this and (4.20), we have

$$\left|\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}] - \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[N_{\sigma(k)}]\right| \le \frac{C\varepsilon T}{k}\sqrt{\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}]}.$$

Using this inequality, we obtain

$$|M(k-1,k)-M(k,k)| \le \mathop{\mathbf{E}}_{\sigma}\left[\left|\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}] - \mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k])}}[N_{\sigma(k)}]\right|\right] \\ \le \frac{C\varepsilon T}{k}\mathop{\mathbf{E}}_{\sigma}\sqrt{\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}]} \le \frac{C\varepsilon T}{k}\sqrt{\mathop{\mathbf{E}}_{\sigma}\left[\mathop{\mathbf{E}}_{\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}[N_{\sigma(k)}]\right]} = \frac{C\varepsilon T}{k}\sqrt{M(k-1,k)},$$

where the first and the last inequalities come from Jensen's inequality. Hence, $M(k,k)$ is bounded as $M(k,k)\le M(k-1,k)+\frac{C\varepsilon T}{k}\sqrt{M(k-1,k)}$. Similarly, we can also show that $M(k,1)\le M(k-1,1)+\frac{C\varepsilon T}{k}\sqrt{M(k-1,1)}$. Noting that $M(k,1)=M(k,k)$, we obtain $M(k,k)\le$

$\beta + \frac{C\varepsilon T}{k}\sqrt{\beta}$ for $\beta = \min\{M(k-1,k), M(k-1,1)\}$. Since we have $M(k-1,1) = M(k-1,2) = \cdots = M(k-1,k-1)$ and $M(k-1,k) = \cdots = M(k-1,d)$, and $\sum_{i=1}^{d} M(k-1,i) = \underset{\sigma\sim U(\Sigma_d),\ell_1,\ldots,\ell_T\sim D_{\sigma([k-1])}}{\mathbf{E}}\left[\sum_{i=1}^{d} N_i\right] = kT$, we have $\beta \le \frac{kT}{d}$. Hence, we have $M(k,k) \le \frac{kT}{d} + \frac{C\varepsilon T}{k}\sqrt{\frac{kT}{d}} = T\left(\frac{k}{d} + C\sqrt{\frac{T}{dk}}\varepsilon\right)$. By setting $\varepsilon = 2^{-16}\min\left\{k^{-1/4}, \frac{d-k}{2Cd}\sqrt{\frac{dk}{T}}\right\}$, we have $M(k,k) \le T(\frac{k}{d} + \frac{d-k}{2d}) = \frac{T(d+k)}{2d}$. From (4.19), we have $\mathbf{E}[R_T] \ge \varepsilon k(T - M(k,k)) \ge \varepsilon kT\left(1 - \frac{d+k}{2d}\right) = \frac{\varepsilon kT(d-k)}{2d} = 2^{-16}\min\left\{\frac{1}{C}\left(\frac{d-k}{d}\right)^2\sqrt{dk^3T}, \quad \frac{d-k}{2d}k^{\frac{3}{4}}T\right\}$. $\square$

4.4.5 Lower Bound for the Bandit Ranking Problem

For the bandit ranking problem, Cohen et al. [44] have provided lower bounds by considering $\ell_t \sim D_{a^*}$ for $a^* \in \mathcal{A}$, similarly to the multitask bandit problem. Unfortunately, this approach does not work well for the case of full permutations (i.e., with $k = n$), and has left an $\Omega(\sqrt{n})$-gap between the lower and the upper bounds, as mentioned in their conclusion.

We can get rid of this $\Omega(\sqrt{n})$-gap by improving the lower bound, by means of a surprisingly simple approach. In contrast to the probability distribution by Cohen et al. [44] that has k good arms (i such that $a^*_i = 1$), we consider the probability distribution with $m = \lceil k/2 \rceil$ good arms, i.e., we consider $a^* \in \mathcal{A}' \subseteq \{0,1\}^d$ defined by

$$\mathcal{A}' = \left\{ a \in \{0,1\}^d \;\middle|\; \sum_{i=(j-1)n+1}^{jn} a_i = \begin{cases} 1 & (1 \le j \le m) \\ 0 & (m < j \le k) \end{cases}, \; \sum_{i=1}^{k} a_{(i-1)n+j} \le 1 \quad (j \in [n]) \right\}. \tag{4.21}$$

Lemma 4.4.3. *Suppose a family $\{D_{a^*} \mid a^* \in \{0,1\}^d\}$ of distributions with a parameter $\varepsilon \le \sqrt{d/(32C_D kT)}$ satisfies (4.4) and (4.7). Suppose $n \ge 2$ and $1 \le k \le n$. If a^* is chosen from $\mathcal{A}'$ defined by (4.21) and ℓ_t follows D_{a^*} for $t = 1, 2, \ldots, T$, independently, then, for the bandit ranking problem defined by (4.2), any algorithm suffers regret of $\mathbf{E}[R_T] = \Omega(\varepsilon kT)$.*

Proof. We start with introducing notations: There is a one-to-one correspondence between $\mathcal{A}$ defined by (4.2) and the set of all injection σ from $[k]$ to $[n]$. In fact, given an injection $\sigma : [k] \to [n]$, the indicator vector $\chi(S) \in \{0,1\}^d$ of S for $S = \{(i-1)n + j \mid i \in [k], j \in [n], \sigma(i) = j\}$ is an element of $\mathcal{A}$. Conversely, for any $a \in \mathcal{A}$, there is a unique injection $\sigma : [k] \to [n]$ such that $a_{(i-1)n+j} = 1$ if and only if $j = \sigma(i)$. Hence, we can regard each element in $a \in \mathcal{A}$ as an injection from $[k]$ to $[n]$. For outputs a_t from the algorithm, let denote $\sigma_t : [k] \to [n]$ the corresponding injection. Similarly, There is a one-to-one correspondence between $\mathcal{A}'$ and the set of all injection σ from $[m]$ to $[n]$. Let $\sigma^* : [m] \to [n]$ denote the injection corresponding to $a^* \in \mathcal{A}'$. Then we have $a^* = \chi(\{(i-1)n + j \mid i \in [m], j \in [n], \sigma^*(i) = j\}) =: b(\sigma^*)$. Let $\Sigma_{m,n}$ denote the set of all injections from $[m]$ to $[n]$.

Since we have $\underset{\ell_t\sim D_{a^*}}{\mathbf{E}}[\hat{\ell}_t^\top a] = \frac{k}{2} - \varepsilon\sum_{i=1}^{m} a_{(i-1)n+\sigma^*(i)}$ from (4.4), the expectation of the regret can be expressed as

$$\mathbf{E}[R_T] = \sum_{t=1}^{T}\sum_{i=1}^{m} \varepsilon\left(1 - \mathbf{E}[a_{t,(i-1)n+\sigma^*(i)}]\right) = \varepsilon\left(mT - \sum_{i=1}^{m}\mathbf{E}\left[N_{(i-1)n+\sigma^*(i)}\right]\right), \tag{4.22}$$

where the expectation is taken w.r.t. $\ell_t \sim D_{a^*} = D_{b(\sigma^*)}$ for $t \in [T]$. We consider bounding $\mathbf{E}\left[N_{(i-1)n+\sigma^*(i)}\right]$ by means of Lemma 4.4.1. For $\sigma^* : [k] \to [n]$, define $b'(\sigma^*) := \chi(\{(i-1)n + j \mid$

$i \in [m-1], j \in [n], \sigma^*(i) = j\})$. From Lemma 4.4.1, we have

$$\begin{aligned}
&\left| \mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] - \mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \right| \\
&\leq \mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \left| \mathop{\mathbf{E}}_{\ell_t \sim D_{b(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] - \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \right| \\
&\leq T \mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \sqrt{\sum_{t=1}^{T} \mathop{\mathbf{E}}_{a_t \sim A_t(D_{b'(\sigma^*)})} \left[\mathop{D_{\mathrm{KL}}}_{\ell \sim D_{b'(\sigma^*)}, \ell' \sim D_{b(\sigma^*)}} (a_t^\top \ell || a_t^\top \ell') \right]} \\
&\leq T \sqrt{\mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{a_t \sim A_t(D_{b'(\sigma^*)})} \left[\mathop{D_{\mathrm{KL}}}_{\ell \sim D_{b'(\sigma^*)}, \ell' \sim D_{b(\sigma^*)}} (a_t^\top \ell || a_t^\top \ell') \right]}, \qquad (4.23)
\end{aligned}$$

where the first and the third inequalities follows from Jensen's inequality, and the second inequality follows from Lemma 4.4.1. Let us consider $D_{\mathrm{KL}}(a_t^\top \ell || a_t^\top \ell')$ for $\ell \sim D_{b'(\sigma^*)}$, $\ell' \sim D_{b(\sigma^*)}$, fixed $\sigma^* \in \Sigma_{m,n}$, and fixed $a_t \in \mathcal{A}$. If $a_{t,(m-1)n+\sigma(m)} = 1$, from the assumption of (4.7), we have $D_{\mathrm{KL}}(a_t^\top \ell || a_t^\top \ell') \leq C_D \varepsilon^2$. Otherwise, we have $D_{\mathrm{KL}}(a_t^\top \ell || a_t^\top \ell') = 0$ because the probabilistic distribution of $a_t^\top \ell$ is equal to that of $a_t^\top \ell'$. Hence, We have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathop{\mathbf{E}}_{a_t \sim A_t(D_{b'(\sigma^*)})} \left[\mathop{D_{\mathrm{KL}}}_{\ell \sim D_{b'(\sigma^*)}, \ell' \sim D_{b(\sigma^*)}} (a_t^\top \ell || a_t^\top \ell') \right] \\
&\leq \sum_{t=1}^{T} \mathop{\mathrm{Prob}}_{a_t \sim A_t(D_{b'(\sigma^*)})} [a_{t,(m-1)n+\sigma(m)} = 1] C_D \varepsilon^2 = \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] C_D \varepsilon^2 \qquad (4.24)
\end{aligned}$$

Define $S \in \mathbb{R}$ by

$$S := \mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \qquad (4.25)$$

Combining the above two inequalities (4.23) and (4.24), we have

$$\mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \leq S + T\sqrt{C_D \varepsilon^2 S}. \qquad (4.26)$$

Then we evaluate S defined by (4.25). Let $\sigma^*|_{[m-1]}$ denote the restriction of $\sigma^* : [m] \to [n]$ to $[m-1]$, i.e., $\sigma^*|_{[m-1]} : [m-1] \to [n]$ is defined by $\sigma^*|_{[m-1]}(i) = \sigma^*(i)$ for $i \in [m-1]$. Let $R'(\sigma^*)$ denote the range of $\sigma^*|_{[m-1]}$. From the definition of b', $b'(\sigma^*)$ does not depend on σ^*, but is determined by $\sigma^*|_{[m-1]}$. If σ^* follows an uniform distribution over $\Sigma_{m,n}$, the posterior probability of $\sigma^*(m)$ given $\sigma^*|_{[m-1]}$ is an uniform distribution over $[n] \setminus R'(\sigma^*)$. Hence, S can be evaluated as follows:

$$\begin{aligned}
S &= \mathop{\mathbf{E}}_{\sigma^*|_{[m-1]}} \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} \mathop{\mathbf{E}}_{\sigma^*(m) \sim U([n] \setminus R'(\sigma^*))} [N_{(m-1)n+\sigma(m)}] \\
&= \mathop{\mathbf{E}}_{\sigma^*|_{[m-1]}} \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} \left[\frac{1}{|[n] \setminus R'(\sigma^*)|} \sum_{j \in [n] \setminus R'(\sigma^*)} N_{(m-1)n+j} \right] \\
&= \mathop{\mathbf{E}}_{\sigma^*|_{[m-1]}} \mathop{\mathbf{E}}_{\ell_t \sim D_{b'(\sigma^*)}} \left[\frac{1}{|[n] \setminus R'(\sigma^*)|} \sum_{t=1}^{T} \sum_{j \in [n] \setminus R'(\sigma^*)} a_{t,(m-1)n+j} \right] \leq \frac{T}{n-m+1}, \qquad (4.27)
\end{aligned}$$

where the last inequality comes from $|R'(\sigma*)| = m - 1$ and that $a_t \in \mathcal{A}$ defined by (4.2). Combining (4.26) and (4.27), we have

$$\mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \le \frac{T}{n-m+1} + T\sqrt{\frac{C_D \varepsilon^2 T}{n-m+1}}.$$

Since we assume $k \le n$ and $n \ge 2$, we have $n - m + 1 = n - \lceil k/2 \rceil + 1 \ge n - \lceil n/2 \rceil + 1 \ge \max\{2, n/2\}$. Hence, by setting $\varepsilon = \sqrt{\frac{n}{32 C_D T}}$, we obtain

$$\mathop{\mathbf{E}}_{\sigma^* \sim U(\Sigma_{m,n})} \mathop{\mathbf{E}}_{\ell_t \sim D_{b(\sigma^*)}} [N_{(m-1)n+\sigma(m)}] \le \frac{T}{2} + T\sqrt{\frac{2C_D \varepsilon^2 T}{n}} = \frac{T}{2} + \frac{T}{4} = \frac{3T}{4}.$$

For each $i \in [m]$ besides m, we can show $\mathbf{E}[N_{(i-1)n+\sigma(i)}] \le \frac{3T}{4}$ in a similar way. Then, from this and (4.22), we have

$$\mathbf{E}[R_T] \ge \varepsilon \left(mT - \sum_{i=1}^{m} \frac{3T}{4} \right) = \frac{\varepsilon m T}{4} = \frac{m}{4}\sqrt{\frac{nT}{32 C_D}} \ge \frac{k}{8}\sqrt{\frac{nT}{32 C_D}} = \frac{1}{8}\sqrt{\frac{dkT}{32 C_D}}.$$

where the inequality comes from $m = \lceil k/2 \rceil \ge k/2$ and the last inequality domes from $d = kn$. □

The lower bound in Lemma 4.4.3 is valid even if $k = n$, while the previous work [44] considering $a^* \in \mathcal{A}$ applies only to the case of $n \ge 2k$. Intuitively, this difference can be explained as follows: the regret depends on the number of good arms ($i \in [d]$ such that $a^*_i = 1$) in chosen arms ($i \in [d]$ such that $a_{ti} = 1$). If a^* and a_t are chosen from $\mathcal{A}$ with $k = n$, and if the chosen arms (defined by a_t) includes $k - 1$ good arms, then the chosen arms automatically includes the whole $[k]$ good arms, because a^* and a_t express edge sets of perfect matchings of the complete bipartite graph $K_{k,k}$. This means that, in this setting, the probability of choosing some good arms strongly affects that of choosing other good arms, which makes the analysis difficult. On the other hand, such an effect can be reduced if a^* is chosen from $\mathcal{A}'$, i.e., a^* has only $m = \lceil k/2 \rceil$ good arms.

The lower bound in Theorem 4.2.1 for the bandit ranking problem, i.e., $\mathcal{A}$ given by (4.2), can be derived in the same way as in Section 4.4.3. This accomplishes the proof of Theorem 4.2.1.

4.5 Upper Bounds

In this section, we give the proof of Theorems 4.2.3. We consider a generalization of the bandit combinatorial optimization called *stochastic linear bandit* with finite number of arms. In this problem, a player is given a finite decision set $\mathcal{A}$ before the game starts. In each round $t \in [T]$, the player chooses action $a_t \in \mathcal{A}$. After that, observe loss $L_t = \ell^{*\top} a_t + \eta_t$, where η_t is conditionally α-subgaussian given $a_1, L_1, a_2, L_2, \ldots, a_{t-1}, L_{t-1}$ and a_t, i.e., $\mathbf{E}[\exp(\lambda \eta_t) \mid \mathcal{F}_t] \le \exp(\alpha^2 \lambda^2 / 2)$ almost surely, for $\mathcal{F}_t = \sigma(a_1, L_1, \ldots, a_{t-1}, L_{t-1}, a_t)$, which is the σ-algebra generated by $\{a_1, L_1, \ldots, a_{t-1}, L_{t-1}, a_t\}$. We suppose that the suboptimality gap $\max_{a,b \in \mathcal{A}} \ell^{*\top}(a - b)$ is at most L. For this problem, we define the regret R'_T as follows:

$$R'_T = \max_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell^{*\top}(a_t - a) = \sum_{t=1}^{T} \ell^{*\top} a_t - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell^{*\top} a = \sum_{t=1}^{T} \ell^{*\top}(a_t - a^*), \tag{4.28}$$

where we define $a^* \in \arg\min_{a \in \mathcal{A}} \left\{ \ell^{*\top} a \right\}$.

4.5.1 Algorithm for Stochastic Linear Bandit

We analyze R'_T for output of Algorithm 7, which is modified from Algorithm 12 in Section 22 of the preprint book by Lattimore and Szepesvári [118]. The differences between Algorithm 7 here and Algorithm 12 in [118] are:

- They deal with only the case in which the noise η_t has a bounded variance, i.e., $\alpha = 1$. To deal with the case for a general α, we modify the definition (4.31) of T_k in their algorithm.
- They assume that the suboptimality gap $\max_{a,b \in \mathcal{A}} \{\ell^{*\top}(a-b)\}$ is bounded by 1. To cope well with changing suboptimality gaps, we modify the definition of ε_t in their algorithm.
- They basically consider maximization problems, while we consider minimization (This results in no essential differences).

Algorithm 7 is controlled by parameters $\varepsilon_1 > 0$ and $\delta > 0$. The algorithm divides rounds into *phases*: the k-th phase consists of T_k rounds, where T_k will be defined later. In each phase, a subset $\mathcal{A}_k$ of action set $\mathcal{A}$ is maintained. The algorithm chooses actions from $\mathcal{A}_k$ in the k-th phase, and $\mathcal{A}_k$ does not change over all rounds in this phase. At the beginning of each phase, the algorithm constructs a probabilistic measure π_k over $\mathcal{A}_k$ satisfying the following:

$$\max_{b \in \mathcal{A}_k} \left\{ b^\top \left(\sum_{a \in \mathcal{A}_k} \pi_k(a) a a^\top \right)^{-1} b \right\} \leq d, \quad \|\pi_k\|_0 \leq \frac{d(d+1)}{2} + 1. \tag{4.29}$$

Such a measure always exists for all k. Indeed, as Kiefer and Wolfowitz [107] showed, if $\mathrm{span}(\mathcal{A}) = \mathbb{R}^d$, a maximizer π^* of $\det(\sum_{a \in \mathcal{A}} \pi(a) a a^\top)$ satisfies

$$\max_{b \in \mathcal{A}} \left\{ b^\top \left(\sum_{a \in \mathcal{A}} \pi^*(a) a a^\top \right)^{-1} b \right\} = d. \tag{4.30}$$

Even if $\mathrm{span}(\mathcal{A})$ does not equal $\mathbb{R}^d$, equivalently if $d' = \dim(\mathrm{span}(\mathcal{A}))$ is smaller than d, we can obtain π^* for which the left-hand side of (4.30) is equal to d', by maximizing the determinant of $\sum_{a \in \mathcal{A}} \pi(a)(Ba)(Ba)^\top$ for an appropriate matrix $B \in \mathbb{R}^{d' \times d}$. Hence, the left inequality of (4.29) can be satisfied. Further, Carathéodory's theorem implies that for arbitrary $\pi \in \Delta^{\mathcal{A}} := \{\pi : \mathcal{A} \to \mathbb{R}_{\geq 0} \mid \sum_{a \in \mathcal{A}} \pi(a) = 1\}$, there exists $\pi' \in \Delta^{\mathcal{A}}$ such that

$$\sum_{a \in \mathcal{A}} \pi(a) a a^\top = \sum_{a \in \mathcal{A}} \pi'(a) a a^\top, \quad \|\pi'\|_0 \leq \dim(\mathrm{span}\{a a^\top \mid a \in \mathcal{A}\}) + 1.$$

The dimensionality of $\mathrm{span}\{a a^\top \mid a \in \mathcal{A}\}$ is at most $d(d+1)/2$, which is the dimensionality of the linear space of all symmetric matrices of size d. Hence, the right inequality of (4.29) can be satisfied.

The k-th phase consists of T_k rounds from the $(t_k + 1)$-th round to the t_{k+1}-th round, in which the algorithm chooses action $a \in \mathcal{A}_k$ in exactly $T_k(a)$ rounds for each $a \in \mathcal{A}_k$. Here, $T_k(a)$ $(a \in \mathcal{A}_k)$, T_k, and t_k are defined as

$$T_k(a) = \left\lceil \frac{2d\alpha^2 \pi_k(a)}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} \right\rceil, \quad T_k = \sum_{a \in \mathcal{A}_k} T_k(a), \quad t_k = \sum_{j=1}^{k-1} T_j, \tag{4.31}$$

Algorithm 7 Algorithm for stochastic linear bandits with finite arms

```
Require: $\mathcal{A} \subseteq \mathbb{R}^d, \alpha, \delta, \varepsilon_1$
1: Set $\mathcal{A}_1 = \mathcal{A}, t_1 = 0$.
2: for $k = 1, \ldots$ do
3:     Let $\pi_k \in \Delta^{\mathcal{A}_k}$ be a probabilistic measure over $\mathcal{A}_k$ such that (4.29) is satisfied.
4:     Define $T_k(a)$, $T_k$ and $t_{k+1}$ by (4.31)
5:     Choose action $a \in \mathcal{A}_k$ exactly $T_k(a)$ times, from the $(t_k + 1)$-th round to the $t_{k+1}$-th
       round.
6:     Calculate empirical estimate $\hat{\ell}_k$ of $\ell^*$ by (4.32).
7:     Eliminate arms with a high estimated loss on the basis of (4.33).
8: end for
```

where $\varepsilon_k = 2^{-k+1}\varepsilon_1$. At the end of the k-th phase, the algorithm calculates a least squares estimator $\hat{\ell}_k$ of ℓ^* by

$$\hat{\ell}_k = V_k^{-1} \sum_{t=t_k+1}^{t_k+T_k} r_t a_t \quad \text{with} \quad V_k = \sum_{a \in \mathcal{A}_k} T_k(a) a a^\top = \sum_{t=t_k+1}^{t_k+T_k} a_t a_t^\top. \tag{4.32}$$

Moreover, $\mathcal{A}_{k+1}$ is defined by eliminating actions that are not promising, as follows:

$$\mathcal{A}_{k+1} = \left\{ a \in \mathcal{A}_k \;\middle|\; \min_{b \in \mathcal{A}_k} \hat{\ell}_k^\top (b - a) \geq -2\varepsilon_k \right\}. \tag{4.33}$$

4.5.2 Regret Bound for Stochastic Linear Bandit

The outputs of Algorithm 7 satisfy the regret upper bound in Theorem 4.2.4.

Proof of Theorem 4.2.4. To derive an upper bound of the regret defined by (4.28), we first consider confidence bound for $\hat{\ell}_k^\top a$. From the standard analysis of confidence bounds for least squares estimators (see, e.g., [118]), for all $b \in \mathbb{R}^d$, we have

$$\begin{aligned} &\operatorname{Prob}\left[|(\hat{\ell}_k - \ell^*)^\top b| \geq \alpha \sqrt{2 b^\top V_k^{-1} b \log \frac{2|\mathcal{A}|k(k+1)}{\delta}} \;\middle|\; \mathcal{F}_{t_{k-1}} \right] \\ &\leq \exp\left(-\log \frac{2|\mathcal{A}|k(k+1)}{\delta} \right) = \frac{\delta}{|\mathcal{A}|k(k+1)}. \end{aligned} \tag{4.34}$$

From the definitions of V_k, T_k and π_k, it holds for all $b \in \mathcal{A}_k$ that

$$\begin{aligned} b^\top V_k^{-1} b &= b^\top \left(\sum_{a \in \mathcal{A}_k} T_k(a) a a^\top \right)^{-1} b \\ &\leq \left(\frac{2d\alpha^2}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} \right)^{-1} b^\top \left(\sum_{a \in \mathcal{A}_k} \pi_k(a) a a^\top \right)^{-1} b \leq \left(\frac{2\alpha^2}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} \right)^{-1}, \end{aligned}$$

where the first equality comes from (4.32), and the first and the second inequalities come from (4.31) and (4.29), respectively. Combining the above and (4.34), we obtain

$$\operatorname{Prob}[|(\ell^* - \hat{\ell}_k)^\top b| \geq \varepsilon_k | \mathcal{F}_{t_{k-1}}] \leq \frac{\delta}{|\mathcal{A}|k(k+1)}$$

for all k and $b \in \mathcal{A}_k$. Hence, we obtain

$$\mathrm{Prob}\left[\exists k,\ \exists b \in A_k,\ |(\ell^* - \hat{\ell}_k)^\top b| > \varepsilon_k\right] \leq \sum_{k=1}^{\infty} \frac{|\mathcal{A}_k|\delta}{|\mathcal{A}|k(k+1)} \leq \delta \sum_{k=1}^{\infty} \frac{1}{k(k+1)} = \delta.$$

In the discussion below, we assume that

$$|(\ell^* - \hat{\ell}_k)^\top a| \leq \varepsilon_k \text{ for all } k \in \{1, 2, \ldots\} \text{ and } a \in \mathcal{A}_k. \tag{4.35}$$

Then, for all $k = 1, 2, \ldots$, we have $a^* \in \mathcal{A}_k$ because $a^* \in \mathcal{A}_1 = \mathcal{A}$ and $\hat{\ell}_k^\top(b - a^*) \geq \ell^{*\top}(b - a^*) - 2\varepsilon_k \geq -2\varepsilon$ for all k. Further, we have

$$\ell^{*\top}(a - a^*) \leq 8\varepsilon_k \text{ for all } k \in \{2, 3, 4, \ldots\} \text{ and } a \in \mathcal{A}_k. \tag{4.36}$$

Indeed, if $\ell^{*\top}(a - a^*) > 8\varepsilon_k$, we can see that $a \notin \mathcal{A}_k$ in both cases of (i) $a \notin \mathcal{A}_{k-1}$ and $a \in \mathcal{A}_{k-1}$: (i)since $\mathcal{A}_k \subseteq \mathcal{A}_{k-1}$ from the definition (4.33) of $\mathcal{A}_k$, $a \notin \mathcal{A}_{k-1}$ implies $a \notin \mathcal{A}_k$; (ii)assuming $a, a^* \in \mathcal{A}_{k-1}$, $\ell^{*\top}(a - a^*) > 8\varepsilon_k = 4\varepsilon_{k-1}$ and (4.35), we obtain $\hat{\ell}_{k-1}^\top(b - a) \geq \ell^{*\top}(b - a) - 2\varepsilon_{k-1} \geq \ell^{*\top}(a^* - a) - 2\varepsilon_{k-1} > 2\varepsilon_{k-1}$ for all $b \in \mathcal{A}_{k-1}$, which implies $a \notin \mathcal{A}_k$ from the definition (4.33) of $\mathcal{A}_k$.

Define $k(t)$ to be $k \in \{1, 2, \ldots\}$ such that $t_k < t \leq t_{k+1}$. Since $t_k < t \leq t_{k+1}$ means $a_t \in \mathcal{A}_k$, we have $a_t \in \mathcal{A}_{k(t)}$ for all t. Hence, from (4.36), we have $\ell^{*\top}(a_t - a^*) \leq 8\varepsilon_{k(t)}$ for $t > T_1$. Therefore, for $T \geq T_1$, we have

$$\begin{aligned}\sum_{t=1}^{T} \ell^{*\top}(a_t - a^*) &\leq LT_1 + \sum_{t=t_2}^{T} \ell^{*\top}(a_t - a^*) \leq LT_1 + 8\sum_{t=t_2}^{T} \varepsilon_{k(t)} \\ &\leq LT_1 + 8\sum_{t=t_2}^{t_{k(T)+1}} \varepsilon_{k(t)} \leq LT_1 + 8\sum_{k=2}^{k(T)} T_k \varepsilon_k. \end{aligned}\tag{4.37}$$

Let us evaluate T_k and $t_k = \sum_{j=1}^{k-1} T_j$. From the definition (4.31) of T_k, we have

$$\frac{2d\alpha^2}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} \leq T_k < \frac{2d\alpha^2}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} + \frac{d(d+1)}{2} + 1, \tag{4.38}$$

since $T_k(a) - \frac{2d\alpha^2 \pi_k(a)}{\varepsilon_k^2} \log \frac{2|\mathcal{A}|k(k+1)}{\delta} \in [0, 1)$ for at most $d(d+1)/2 + 1$ actions $a \in \mathcal{A}$ and $T_k(a) = 0$ for the other actions. From the left inequality of (4.38), for $T > T_1$, we have

$$T > t_{k(T)} \geq T_{k(T)-1} \geq \frac{2d\alpha^2}{\varepsilon_{k(T)-1}^2} \log \frac{2|\mathcal{A}|k(T)(k(T)-1)}{\delta} \geq \frac{d\alpha^2 2^{2k(T)}}{8\varepsilon_1^2} \log \frac{2|\mathcal{A}|k(T)}{\delta},$$

which implies that

$$2^{k(T)} \leq \frac{\varepsilon_1}{\alpha}\sqrt{\frac{8T}{d}} \left(\log \frac{2|\mathcal{A}|k(T)}{\delta}\right)^{-1/2}, \quad k(T) \leq \log_2\left(\frac{\varepsilon_1}{\alpha}\sqrt{\frac{8T}{d}}\right) \leq \log T. \tag{4.39}$$

From this inequality and (4.36), we have

$$\begin{aligned}
\sum_{m=2}^{k(T)} T_m \varepsilon_m &\le \sum_{m=2}^{k(T)} \left(\frac{2d\alpha^2}{\varepsilon_m} \log \frac{2|\mathcal{A}|m(m+1)}{\delta} + \frac{\varepsilon_m (d+2)^2}{2} \right) \\
&\le \sum_{m=2}^{k(T)} \left(\frac{2d\alpha^2}{\varepsilon_1 2^{-m+1}} \log \frac{2|\mathcal{A}|k(T)(k(T)+1)}{\delta} + \frac{\varepsilon_1 2^{-m+1}(d+2)^2}{2} \right) \\
&\le \frac{2d\alpha^2 2^{k(T)}}{\varepsilon_1} \log \frac{2|\mathcal{A}|k(T)(k(T)+1)}{\delta} + 2\varepsilon_1 d^2 \\
&\le 2\alpha\sqrt{8dT} \left(\log \frac{2|\mathcal{A}|k(T)}{\delta} \right)^{-1/2} \cdot 2 \log \frac{2|\mathcal{A}|k(T)}{\delta} + 2\varepsilon_1 d^2 \\
&\le 4\alpha \sqrt{8dT \log \frac{2|\mathcal{A}|k(T)}{\delta}} + 2\varepsilon_1 d^2 \le 16\alpha \sqrt{dT \log \frac{|\mathcal{A}|k(T)}{\delta}} + 2\varepsilon_1 d^2,
\end{aligned}$$

where the first inequality comes from the right inequality of (4.38), the fourth inequality comes from the left inequality of (4.38) and the fact that $2|\mathcal{A}|k(T)(k(T)+1) \le (2|\mathcal{A}|k(T))^2$. From the above inequality and (4.37), we have

$$\begin{aligned}
\sum_{t=1}^{T} \ell^{*\top}(a_t - a^*) &\le LT_1 + 128\alpha \sqrt{dT \log \frac{|\mathcal{A}| \log T}{\delta}} + 16\varepsilon_1 d^2 \\
&\le 128\alpha \sqrt{dT \log \frac{|\mathcal{A}| \log T}{\delta}} + \frac{4dL\alpha^2}{\varepsilon_1^2} \log \frac{|\mathcal{A}|}{\delta} + (L + 16\varepsilon_1) d^2.
\end{aligned}$$

□

4.5.3 Regret Bound for Stochastic Combinatorial Bandit

In this subsection, we show Theorem 4.2.3 by means of Theorem 4.2.4.

Proof of Theorem 4.2.4. Suppose ℓ_t follows an arm-wise independent distribution D^*, i.i.d. Then, the bandit combinatorial optimization for $\{\ell_t\}$ will be a special case of the stochastic linear bandits with $\ell^* = \mathop{\mathbf{E}}_{\ell \sim D^*}[\ell]$ and $\eta_t = (\ell_t - \eta^*)^\top a_t$. In this problem, the suboptimality gap $L = \max_{a,b \in \mathcal{A}} \ell^{*\top}(a - b)$ is at most k because $\ell^* \in [0,1]$ and $\mathcal{A} \subseteq \{a \in \{0,1\}^d \mid \|a\|_0 = k\}$. Further, η_t is $\sqrt{k}/2$-subgaussian from Hoeffding's Lemma, since $\ell_{t1}, \ell_{t2}, \ldots, \ell_{td}$ are independent and $[0,1]$-valued random variables. Hence, applying Algorithm 7 with $L = k$, $\alpha = \sqrt{k}/2$ and $\delta = \delta'/2$, we have

$$R'_T = \sum_{t=1}^{T} \ell^{*\top}(a_t - a^*) \le 64 \sqrt{dkT \log \frac{2|\mathcal{A}| \log T}{\delta'}} + \frac{dk^2}{\varepsilon_1^2} \log \frac{2|\mathcal{A}|}{\delta'} + (k + 16\varepsilon_1) d^2 \tag{4.40}$$

with probability $1 - \delta'/2$. Moreover, we have $R_T - R'_T = O(\sqrt{kT \log(|\mathcal{A}|/\delta')})$ with probability $1 - \delta'/2$. In fact, we have $\sum_{t=1}^{T} \ell_t^\top (a_t - a) - R'_T \le \sum_{t=1}^{T} \eta_t - \sum_{t=1}^{T} (\ell_t - \ell^*)^\top a$ for all $a \in \mathcal{A}$, and from the Azuma-Hoeffding inequality, we have

$$\mathrm{Prob}\left[\sum_{t=1}^{T} \eta_t \ge \sqrt{\frac{kT}{2} \log \frac{4|\mathcal{A}|}{\delta'}} \right] \le \frac{\delta'}{4|\mathcal{A}|}, \quad \mathrm{Prob}\left[\sum_{t=1}^{T} (\ell^* - \ell_t)^\top a \ge \sqrt{\frac{kT}{2} \log \frac{4|\mathcal{A}|}{\delta'}} \right] \le \frac{\delta'}{4|\mathcal{A}|}.$$

From the second inequality, the probability that there exists $a \in \mathcal{A}$ such that $\sum_{t=1}^{T}(\ell^* - \ell_t)^\top a \geq \sqrt{\frac{kT}{2}\log\frac{4|\mathcal{A}|}{\delta'}}$ is at most $\delta'/4$. Combining this and the above first inequality, we obtain $R_T - R'_T \leq \sqrt{2kT\log\frac{4|\mathcal{A}|}{\delta'}}$ with probability $1-\delta'/2$. From this and (4.40), with probability $1-\delta'$, we have $R_T \leq R'_T + \sqrt{2kT\log\frac{4|\mathcal{A}|}{\delta'}} \leq 66\sqrt{dkT\log\frac{2|\mathcal{A}|\log T}{\delta'}} + \frac{dk^2}{\varepsilon_1^2}\log\frac{2|\mathcal{A}|}{\delta'} + (k+16\varepsilon_1)d^2 \leq 66\sqrt{dk^2T\log\frac{2\mathrm{e}d\log T}{k\delta'}} + \frac{dk^3}{\varepsilon_1^2}\log\frac{2\mathrm{e}d}{k\delta'} + (k+16\varepsilon_1)d^2$. By setting $\varepsilon_1 = \Theta(k)$ and $\delta' = \Theta(\sqrt{d}T)$, we have $R_T = O(\sqrt{dk^2T\log(\mathrm{e}d\log T/k\delta')} + d^2k)$ with probability $1-\delta'$, and $R_T = O(kT)$ with probability δ'. Hence, we have $\mathbf{E}[R_T] = O(\sqrt{dk^2T\log(\mathrm{e}d\log T/k\delta')} + d^2k + \delta'kT) = O(\sqrt{dk^2T\log T\log(\mathrm{e}d/k)})$ for $T = \Omega(d^3)$. □

4.6 Conclusion

This chapter has considered regret bounds of bandit combinatorial optimization. We have proven regret lower bounds that are improved upon the existing study [44] by a factor of $\sqrt{\log T}$. Our lower bounds apply to three practically important examples of bandit combinatorial optimization, and are valid under parameter constraints milder than in existing studies. In particular, our bound for the bandit ranking answers to an open problem posed in [44]. To shave off $\sqrt{\log T}$ factor, we have introduced a novel class of distributions, which is potentially used to improve regret lower bounds even in other problems. We have also shown that correlation among losses is essential to having a large regret, by proving a smaller regret bound under the assumption of independent losses.

For bandit combinatorial optimization, this work has decreased the gap between the upper and the lower bounds to $O(\log(\mathrm{e}d/k))$. We leave it as an open question to improve this to a constant factor alone.

Chapter 5

Submodular Function Minimization with Noisy Evaluation Oracle

This chapter considers submodular function minimization with *noisy evaluation oracles* that return the function value of a submodular objective with zero-mean additive noise. For this problem, we provide an algorithm that returns an $O(n^{3/2}/\sqrt{T})$-additive approximate solution in expectation, where n and T stand for the size of the problem and the number of oracle calls, respectively. There is no room for reducing this error bound by a factor smaller than $O(1/\sqrt{n})$. Indeed, we show that any algorithm will suffer additive errors of $\Omega(n/\sqrt{T})$ in the worst case. Further, we consider an extended problem setting with *multiple-point feedback* in which we can get the feedback of k function values with each oracle call. Under the additional assumption that each noisy oracle is submodular and that $2 \leq k = O(1)$, we provide an algorithm with an $O(n/\sqrt{T})$-additive error bound as well as a worst-case analysis including a lower bound of $\Omega(n/\sqrt{T})$, which together imply that the algorithm achieves an optimal error bound up to a constant.

5.1 Introduction

Submodular function minimization (SFM) is an important problem that appears in a wide range of research areas, including image segmentation [97, 109], learning with structured regularization [19], and pricing optimization [87, 88]. The goal in this problem is to find a minimizer of a *submodular function*, a function $f : 2^{[n]} \to \mathbb{R}$ defined on the subsets of a given finite set $[n] := \{1, 2, \ldots, n\}$ and satisfying the following inequality:

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y). \tag{5.1}$$

This condition is equivalent to the *diminishing marginal returns* property (see, e.g., [64]): for every $X \subseteq Y \subseteq [n]$ and $i \in [n] \setminus Y$, $f(X \cup \{i\}) - f(X) \geq f(Y \cup \{i\}) - f(Y)$.

Existing studies on SFM assume access to an *evaluation oracle* for f that returns the value $f(X)$ for any X in the feasible region. Under this assumption, a number of efficient algorithms have been discovered, in which the number of oracle calls as well as other computational time is bounded by a polynomial in n. The first polynomial-time algorithm was given by Grötschel, Lovász, and Schrijver [70] and was based on the ellipsoid method. Combinatorial strongly polynomial-time algorithms have been independently proposed by Iwata, Fleischer, and Fujishige [94] and by Schrijver [151]. The current best computational time is of $O(n^3 \log^2 n \cdot \mathrm{EO} + n^4 \log^{O(1)} n)$ by Lee et al. [121], where EO denotes the time taken by the evaluation oracle to answer a single query. For approximate optimization, Chakrabarty et al. [40] have recently proposed an algorithm that finds an ε-additive approximate solution in $\tilde{O}(n^{5/3} \cdot \mathrm{EO}/\varepsilon^2)$ time.

In some applications, however, evaluation oracles are not always available, and only *noisy* function values are observable. For example, in the pricing optimization problem, let us consider selling n types of products, where the value of the objective function $f(X)$ corresponds to the expected gross profit, and the variable $X \subseteq [n]$ corresponds to the set of discounted products. In this scenario, Ito and Fujimaki [87] have shown that $-f(X)$ is a submodular function under certain assumptions, which means that the problem of maximizing the gross profit $f(X)$ is an example of SFM. In a realistic situation, however, we are not given an explicit form of f, and the only thing we can do is to observe the sales of products while changing prices. The observed gross profit does not coincide with its expectation $f(X)$, but changes randomly due to the inherent randomness of purchasing behavior or some temporary events. This means that exact values of $f(X)$ are unavailable, and, consequently, existing works do not directly apply to this situation.

To deal with such problems, we introduce SFM with *noisy evaluation oracles* that return a random value with expectation $f(X)$. In other words, the noisy evaluation oracle $\hat{f}$ returns $\hat{f}(X) = f(X) + \xi$, where ξ is a zero-mean noise that may or may not depend on X. We assume access to T independent noisy evaluation oracles $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_T$ with bounded ranges. We start with the *single-point feedback* setting and then study the more general *multiple-point feedback* (or *k-point feedback*) setting: In the former setting, for each $t \in [T]$, we choose one query X_t to feed $\hat{f}_t$, and get feedback of $\hat{f}_t(X_t)$. In the latter setting, we are given a positive integer k, and for each t, choose k queries to feed $\hat{f}_t$ and observe k real values of feedback. Such a situation with multiple-point feedback can be assumed in some applications. For example, in the case of pricing optimization for E-commerce, we can get multiple-point feedback by employing the A/B-testing framework, i.e., by showing different prices to randomly divided groups of customers. Note that each $\hat{f}_t$ is not necessarily submodular even if its expectation is submodular.

Our contribution is two-fold, positive results (algorithms, Theorem 5.4.1) and negative results (worst-case analyses, Theorem 5.4.2): We propose algorithms that return $O(1/\sqrt{T})$-additive approximate solutions, and we show that arbitrary algorithms suffer additive errors of $\Omega(1/\sqrt{T})$ in the worst case. The results are summarized in Table 5.1 with positive results in $O(\cdot)$ notation and negative ones in $\Omega'(\cdot)$ notation.

As shown in Table 5.1, for the single-point feedback setting, we propose an algorithm that finds an $O(n^{3/2}/\sqrt{T})$-additive approximate solution. Moreover, there is no room for reducing this additive error bound by a smaller factor than $O(1/\sqrt{n})$. Indeed, our Theorem 5.4.2 implies that arbitrary algorithms, including those requiring exponential time and space, suffer at least $\Omega(n/\sqrt{T})$ additive errors. For the k-point feedback setting, both the lower and the upper bounds are decreased by $1/\sqrt{k}$ factors, without additional assumptions. Under the assumption that each $\hat{f}_t$ is submodular (Assumption 5.3.1), however, the situation changes: Our proposed algorithm achieves $O(n/\sqrt{kT})$-additive error, which is $O(1/\sqrt{n})$-times smaller than without Assumption 5.3.1. We also show the lower bound of $\Omega(n/\sqrt{2^k T} + \sqrt{n}/\sqrt{kT})$, which implies that, if $k = O(1)$ or $k = \Omega(n)$, then our algorithm is *optimal* up to constant factors, i.e., no algorithms achieve additive errors of a smaller order.

Table 5.1: Additive error bounds for submodular minimization with noisy evaluation oracle.

	Assume $\hat{f}_t$: submodular	Do not assume $\hat{f}_t$: submodular
single-point feedback	[78][1]: $O\left(\frac{n}{T^{1/3}}\right)$ (if $T = \Omega(n^3)$) **[This work]**: $O\left(\frac{n^{3/2}}{\sqrt{T}}\right)$ and $\Omega'\left(\frac{n}{\sqrt{T}}\right)$ ($\Omega'(\cdot) := \Omega(\min\{1, \cdot\})$)	
k-point feedback ($2 \leq k \leq n$)	**[This work]**: $O\left(\frac{n}{\sqrt{kT}}\right)$ and $\Omega'\left(\frac{n}{\sqrt{2^k T}} + \frac{\sqrt{n}}{\sqrt{T}}\right)$	**[This work]**: $O\left(\frac{n^{3/2}}{\sqrt{kT}}\right)$ and $\Omega'\left(\frac{n}{\sqrt{kT}}\right)$
$(n+1)$-point feedback	[78]: $O\left(\frac{\sqrt{n}}{\sqrt{T}}\right)$ and $\Omega'\left(\frac{\sqrt{n}}{\sqrt{T}}\right)$	**[This work]**: $O\left(\frac{n}{\sqrt{T}}\right)$ and $\Omega'\left(\frac{\sqrt{n}}{\sqrt{T}}\right)$

To construct the algorithms, we combine a convex relaxation technique based on the *Lovász extension* and *stochastic gradient descent* (SGD) method. The Lovász extension for a submodular function is a convex function of which minimizers lead to solutions for SFM. Thanks to this, we can reduce SFM to a convex optimization problem. In this study, we seek a minimizer of the Lovász extension by means of SGD, in which we need to construct unbiased estimators of subgradients. The performance of the SGD depends strongly on the variance of subgradient estimators. We present ways for constructing subgradient estimators, and it turns out that Assumption 5.3.1 enables us to obtain estimators with smaller variances. The combination of Lovász extension and SGD has been already introduced in the work on *bandit submodular minimization* by Hazan and Kale [78]. Our work, however, considers different problem settings, including multiple-point feedback, and presents tighter and more detailed analyses. Details in the difference are given in Section 5.2.

A key technique for our lower bounds comes from the proof of regret lower bounds for bandit problems by Auer et al. [16]. Their proof consists of two steps: they first construct a probabilistic distribution of inputs for which it is hard to detect a *good arm* offering a large reward, and then show that any algorithm actually chooses the good arm only infrequently. We follow a line similar to these two steps to prove Theorem 5.4.2, in which a number of technical issues arise. In the case of multiple-point feedback, in particular, we need to assess the KL divergence carefully for the observed signals from evaluation oracles.

5.2 Related Work

Bandit submodular minimization (BSM) by Hazan and Kale [78] is strongly related to our model. BSM is described as follows: in each iteration $t \in [T]$, a decision maker chooses $X_t \subseteq [n]$ and observe $f_t(X_t)$, where each $f_t : 2^{[n]} \to [-1, 1]$ is a submodular function. In contrast to our model, no stochastic models for f_t are assumed, and the performance of the decision maker is measured by the *regret* defined as $\mathrm{Regret}_T := \sum_{t=1}^{T} f_t(X_t) - \min_{X \subseteq [n]} \sum_{t=1}^{T} f_t(X)$. This BSM problem can be regarded as a generalization of our problem with single-point feedback under Assumption 5.3.1. Indeed, given a BSM algorithm achieving $\mathrm{Regret}_T \leq b(n, T)$ for some function b, one can construct an SFM algorithm that returns $b(n, T)/T$-additive approximate solutions (see, e.g., [77]). Since a BSM algorithm with an $O(nT^{2/3})$ regret bound has been proposed

[1]This work applies to more general problem settings than ours, *bandit submodular minimization* and *online submodular minimization*. See Section 5.2 for details.

in [78], an $O(n/T^{1/3})$-additive approximate algorithm immediately follows, as in Table 5.1. In BSM, however, it has been left as an open problem whether or not one can achieve $O(n^{O(1)}\sqrt{T})$ regret bounds.

With respect to SFM with an *exact* evaluation oracle, there is a large body of literature [19, 39, 93, 140, 166, 49], in addition to the works mentioned in Section 5.1. The Fujishige-Wolfe algorithm [64], based on Wolfe's minimum norm point algorithm [166] and the connection between minimum norm points and the SFM shown in [63], is known to have the best empirical performance in many cases [18, 65]. Chakrabarty et al. [39] have shown that the Fujishige-Wolfe algorithm finds an ε-additive approximate solution with a running time of $O(n^2(\mathrm{EO}+n)/\varepsilon^2)$. The same runtime bound can be achieved by a gradient descent approach presented by Bach [19].

For submodular function *maximization* with noisy evaluation oracles, there have been many studies . Hassani et al. [75] provided a nearly 1/2-approximate algorithm for monotone submodular maximization. Singla et al. [155] considered a similar problem with applications to crowdsourcing. Karimi et al. [103] considered maximizing weighted coverage functions, a special case of submodular functions, under matroid constraints, and presented an efficient nearly $(1-1/\mathrm{e})$-approximate algorithm. Hassidim and Singer [76] provided a nearly $(1-1/\mathrm{e})$-approximate algorithm for monotone submodular maximization with cardinality constraints. Mokhtari et al. [133] showed that a stochastic continuous greedy method works well for monotone submodular function maximization subject to a convex body constraint. For *minimization* problems with similar assumptions, in contrast to maximization problems, only a little literature can be found. Blais et al. [22] considered approximate submodular minimization with an *approximate oracle* model, and presented a polynomial time algorithm with a high-probability error bound. Their model is more general than ours while their algorithm requires a larger computational cost than ours to achieve a similar error bound. Halabi and Jegelka [73] dealt with minimization of *weakly DR-submodular functions*, which is a class of approximately submodular functions, and provided algorithms with reasonable approximation ratios.

Zero-order or *derivative-free convex optimization* [6, 95, 153], optimization problems with evaluation oracle for convex objectives without access to gradients, is also related to our model because Lovász extensions are convex. For general convex objectives, Agarwal et al. [6], Belloni et al. [20] and Bubeck et al. [34] have proposed algorithms that return $\tilde{O}(1/\sqrt{T})$-additive approximate solutions, ignoring factors of polynomials in $\log T$ and $n^{O(1)}$, where n stands for the dimension of the feasible region. Though the error bounds in these results include factors larger than $O(n^3)$, it has been reported [18, 80] that dependence w.r.t. n can be improved under such additional assumptions as the smoothness and the strong convexity of the objectives. These improved results, however, do not apply to our problems because Lovász extensions are neither smooth nor strongly convex. Multiple-point feedback has been considered in zero-order convex optimization, and some algorithms have been reported to achieve optimal performance in such problem settings [5, 54, 154]. In terms of the lower bound on the additive error, Jamieson et al. [95] and Shamir [153] have shown lower bounds of $\Omega(1/\sqrt{T})$ or $\Omega(1/T)$ for various classes of convex objectives, which, however, do not directly apply to our model.

5.3 Problem Setting

Let n be a positive integer, and let $[n] = \{1, 2, \ldots, n\}$ stand for the finite set consisting of positive integers at most n. Let $L \subseteq 2^{[n]}$ be a distributive lattice, i.e., we assume that $X, Y \in L$ implies $X \cap Y, X \cup Y \in L$. Let $f : L \to [-1, 1]$ be a submodular function that we aim to minimize. In our problem setting, we are not given access to exact values of f, but given *noisy* evaluation oracles $\{\hat{f}_t\}_{t=1}^T$ of f, where $\hat{f}_t$ are random functions from L to $[-1, 1]$ that satisfy $\mathbf{E}[\hat{f}_t(X)] = f(X)$ for all $t = 1, 2, \ldots, T$ and $X \in L$. We also assume that $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_T$ are independent.

Our goal is to construct algorithms for solving the following problem: First, the algorithm is given the decision set L and the number T of available oracle calls. For $t = 1, 2, \ldots, T$, the algorithm chooses $X_t \in L$ and observes $\hat{f}_t(X_t)$. The chosen query X_t can depend on previous observations $\{\hat{f}_j(X_j)\}_{j=1}^{t-1}$. After T rounds of observation, the algorithm outputs $\hat{X} \in L$. We call this problem a *single-point feedback setting.* In an alternative problem setting, a *multi-point* or *k-point feedback setting*, we are given a parameter $k \geq 2$ in addition to T and L. In the k-point feedback setting, the algorithm can choose k queries $X_t^{(1)}, X_t^{(2)}, \ldots, X_t^{(k)} \in L$, and, after that, it observes the values $\hat{f}_t(X_t^{(1)}), \hat{f}_t(X_t^{(2)}), \ldots, \hat{f}_t(X_t^{(k)})$ from the evaluation oracle in each round $t \in T$. In both settings, the performance of the algorithm is evaluated in terms of the additive error E_T defined as $E_T = f(\hat{X}) - \min_{X \in L} f(X)$.

A part of our results relies on the following assumption. Note that the following is assumed only when it is explicitly mentioned.

Assumption 5.3.1. Assume that each $\hat{f}_t : L \to [-1, 1]$ is submodular and that $k \geq 2$.

5.4 Our Contribution

Our contribution is two-fold: positive results (Theorem 5.4.1) and negative results (Theorem 5.4.2).

Theorem 5.4.1. *Suppose $1 \leq k \leq n + 1$. For the problem with k-point feedback, there is an algorithm that returns $\hat{X}$ such that*

$$\mathbf{E}[E_T] = \mathbf{E}[f(\hat{X})] - \min\nolimits_{X \in L} f(X) = O(n^{3/2}/\sqrt{kT}). \tag{5.2}$$

If Assumption 5.3.1 holds, there is an algorithm that returns $\hat{X}$ such that

$$\mathbf{E}[E_T] = \mathbf{E}[f(\hat{X})] - \min\nolimits_{X \in L} f(X) = O(n/\sqrt{kT}). \tag{5.3}$$

The expectation is taken w.r.t. the randomness of oracles $\hat{f}_t$ and the algorithm's internal randomness. In both algorithms, the running time is bounded by $O((k\mathrm{EO} + n \log n)T)$, where EO *stands for the time taken by an evaluation oracle to answer a single query.*

The proof of this theorem is given in Section 5.5.4. If we can choose the number T of oracle calls arbitrarily, we are then able to compute ε-additive approximate solution (in expectation) for arbitrary $\varepsilon > 0$, by means of the algorithm with the error bound (5.2). The computational time for it is of $O(\frac{n^3}{\varepsilon^2}(\mathrm{EO} + \frac{n}{k} \log n))$. Indeed, to find an ε-additive approximate solution, it suffices to set T so that $\varepsilon = \Theta(\frac{n^{3/2}}{\sqrt{kT}})$, which is equivalent to $T = \Theta(\frac{n^3}{k\varepsilon^2})$. The computational time is then bounded as $O((k\mathrm{EO} + n \log n)T) = O(\frac{n^3}{\varepsilon^2}(\mathrm{EO} + \frac{n}{k} \log n))$. Similarly, if Assumption 5.3.1

holds and the algorithm achieving (5.3) is used, an ε-additive approximate solution can be found in $O(\frac{n^2}{\varepsilon^2}(\mathrm{EO} + \frac{n}{k}\log n))$ time.

The following theorem gives an insight regarding how tight the above error bounds in Theorem 5.4.1 are.

Theorem 5.4.2. *There is a probability distribution of instances for which any algorithm suffers errors of*

$$\mathbf{E}[E_T] = \mathbf{E}[f(\hat{X}) - \min_{X \in L} f(X)] = \Omega'(n/\sqrt{kT}), \tag{5.4}$$

where we denote $\Omega'(\cdot) := \Omega(\min\{1, \cdot\})$*. In addition, there is a probability distribution of instances satisfying Assumption 5.3.1 for which any algorithm suffers errors of*

$$\mathbf{E}[E_T] = \mathbf{E}[f(\hat{X}) - \min_{X \in L} f(X)] = \Omega'(n/\sqrt{2^k T} + \sqrt{n/T}). \tag{5.5}$$

The expectation is taken w.r.t. the randomness of the instance f *and oracles* $\hat{f}_t$*, and the algorithm's internal randomness.*

The proof of this theorem is given in Section 5.6.2. From (5.4) in this theorem, we can see that at least $\Omega(\frac{n^2}{\varepsilon^2}\mathrm{EO})$ computational time is required to find an ε-additive approximate solution. This can be shown by an argument similar to that after Theorem 5.4.1. For the problem with exact evaluation oracles, on the other hand, Chakrabarty et al. [39] have proposed an algorithm running in $O(\frac{n^{5/3}}{\varepsilon^2}\mathrm{EO})$. By comparing these two results, we can see that SFM with noisy oracle is essentially harder than SFM with exact oracle.

5.5 Algorithm

5.5.1 Preliminary

Lovász extension of submodular function For a $[0,1]$-valued vector $x = (x_1, \ldots, x_n)^\top \in [0,1]^d$ and a real value $u \in [0,1]$, define $H_x(u) \subseteq [n]$ to be the set of indices i for which $x_i \geq u$, i.e., $H_x(u) = \{i \in [n] \mid x_i \geq u\}$. For a distributive lattice L, define a convex hull $\tilde{L} \subseteq [0,1]^n$ of L as follows: $\tilde{L} = \{x \subseteq [0,1]^n \mid H_x(u) \in L \text{ for all } u \in [0,1]\}$. Given a function $f : L \to \mathbb{R}$, we define the Lovász extension $\tilde{f} : \tilde{L} \to \mathbb{R}$ of f as

$$\tilde{f}(x) = \int_0^1 f(H_x(u))\mathrm{d}u. \tag{5.6}$$

From the definition, we have $\tilde{f}(\chi_X) = f(X)$ for all $X \in L$, i.e., $\tilde{f}$ is an extension of f.[2] The following theorem provides a connection between submodular functions and convex functions:

Theorem 5.5.1 ([126])**.** *A function* $f : L \to \mathbb{R}$ *is submodular if and only if* $\tilde{f}$ *is convex. For a submodular function* $f : L \to \mathbb{R}$*, we have* $\min_{X \in L} f(X) = \min_{x \in \tilde{L}} \tilde{f}(x)$

For a proof of this theorem, see, e.g., [64, 126].

For $x \in [0,1]^n$, let $\sigma : [n] \to [n]$ be a permutation over $[n]$ such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \cdots \geq x_{\sigma(n)}$. For any permutation σ over $[n]$, define $S_\sigma(i) = \{\sigma(j) \mid j \leq i\}$. The Lovász extension

[2] $\chi_X \in \{0,1\}^n$ denotes the indicator vector of X, i.e., $(\chi_X)_i = 1$ for $i \in X$ and $(\chi_X)_i = 0$ for $i \in [n] \setminus X$.

defined by (5.6) can then be rewritten as

$$\tilde{f}(x) = f([0]) + \sum_{i=1}^{n}(f(S_\sigma(i)) - f(S_\sigma(i-1)))x_{\sigma(i)} \tag{5.7}$$

$$= f([0])(1 - x_{\sigma(1)}) + \sum_{i=1}^{n-1} f(S_\sigma(i))(x_{\sigma(i)} - x_{\sigma(i+1)}) + f([n])x_{\sigma(n)}. \tag{5.8}$$

Similar expression can be found, e.g., Lemma 6.19 in the book [64].

Subgradient of Lovász extension From the above two expressions (5.7) and (5.8) of the Lovász extension, we obtain two alternative ways to express its subgradient. For a permutation σ over $[n]$ and $i \in \{0, 1, \dots, n\}$, define $\psi_\sigma(i) \in \{-1, 0, 1\}^n$ as

$$\psi_\sigma(0) = -\chi_{\sigma(1)}, \quad \psi_\sigma(n) = \chi_{\sigma(n)}, \quad \psi_\sigma(i) = \chi_{\sigma(i)} - \chi_{\sigma(i+1)} \quad (i = 1, 2, \dots, n-1). \tag{5.9}$$

A subgradient of $\tilde{f}$ at x can then be expressed by $g(\sigma_x)$ defined as

$$g(\sigma) := \sum_{i=1}^{n}(f(S_\sigma(i)) - f(S_\sigma(i-1)))\chi_{\sigma(i)} \tag{5.10}$$

$$= -f([0])\chi_{\sigma(1)} + \sum_{i=1}^{n-1} f(S_\sigma(i))(\chi_{\sigma(i)} - \chi_{\sigma(i+1)}) + f([n])\chi_{\sigma(n)} = \sum_{i=0}^{n} f(S_\sigma(i))\psi_\sigma(i), \tag{5.11}$$

where (5.10) and (5.11) come from (5.7) and (5.8), respectively.

5.5.2 Stochastic Gradient Descent Method

Our algorithm is based on the *stochastic gradient descent* method for $\tilde{f} : \tilde{L} \to [0, 1]$. To start with, we initialize $x_1 = \frac{1}{2} \cdot \mathbf{1} \in \tilde{L}$. For $t = 1, 2, \dots, T$, we update x_t by iteratively calling the oracle $\hat{f}_t$ to obtain x_{t+1}. In each update, we construct an *unbiased estimator* $\hat{g}_t$ of a subgradient of $\tilde{f}$ at x_t (a more concrete construction will be given later), and x_{t+1} is given by

$$x_{t+1} = P_{\tilde{L}}(x_t - \eta \hat{g}_t), \tag{5.12}$$

where $P_{\tilde{L}} : \mathbb{R}^n \to \tilde{L}$ stands for a Euclidean projection to $\tilde{L}$, i.e., $P_{\tilde{L}}(x) \in \arg\min_{y \in \tilde{L}} \|y - x\|_2$, and $\eta > 0$ is a parameter that we can change arbitrarily. We then compute $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$ and draw u from a uniform distribution over $[0, 1]$, and output $\hat{X} = H_{\bar{x}}(u)$. From (5.6), we have $\mathbf{E}[f(\hat{X})] = \mathbf{E}[\tilde{f}(\bar{x})]$. To analyze the performance of our algorithm, we use the following theorem:

Theorem 5.5.2. *Let $D \in \mathbb{R}^n$ be a compact convex set containing* 0*. For a convex function $\tilde{f} : D \to \mathbb{R}$, let $x_1, \dots, x_T$ be defined by $x_1 = 0$ and $x_{t+1} = P_D(x_t - \eta \hat{g}_t)$, where $\mathbf{E}[\hat{g}_t | x_t]$ is a subgradient of $\tilde{f}$ at x_t for each t. Then, $\bar{x} := \frac{1}{T}\sum_{t=1}^{T} x_t$ satisfies*

$$\mathbf{E}[\tilde{f}(\bar{x})] - \min_{x^* \in D} \tilde{f}(x^*) \leq \frac{1}{T}\left(\frac{\max_{x \in D}\|x\|_2^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbf{E}[\|\hat{g}_t\|_2^2]\right). \tag{5.13}$$

Proof. Since $\tilde{f}$ is convex, from Jensen's inequality, we have

$$\tilde{f}(\bar{x}) = \tilde{f}\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right) \leq \frac{1}{T}\sum_{t=1}^{T} \tilde{f}(x_t). \tag{5.14}$$

Denote $g_t = \mathbf{E}[\hat{g}_t | x_t]$. since g_t is a subgradient of the convex function $\tilde{f}$ at x_t, we have

$$\tilde{f}(x_t) - \tilde{f}(x^*) \leq g_t^\top (x_t - x^*), \tag{5.15}$$

for all $t \in [T]$ and $x^* \in D$. By combining (5.14) and (5.15), we obtain

$$\mathbf{E}[\tilde{f}(\bar{x})] - \tilde{f}(x^*) \leq \frac{1}{T}\mathbf{E}\left[\sum_{t=1}^{T} g_t^\top (x_t - x^*)\right] \leq \frac{1}{T}\mathbf{E}\left[\sum_{t=1}^{T} \hat{g}_t^\top (x_t - x^*)\right]. \tag{5.16}$$

Since we have $\|x_t - \eta \hat{g}_t - x^*\|_2^2 = \|x_t - x^*\|_2^2 - 2\eta \hat{g}_t^\top (x_t - x^*) + \eta^2 \|\hat{g}_t\|_2^2$, the value of $\hat{g}_t^\top (x_t - x^*)$ can be bounded as follows:

$$\hat{g}_t^\top (x_t - x^*) = \frac{1}{2\eta}(\|x_t - x^*\|_2^2 - \|x_t - \eta \hat{g}_t - x^*\|_2^2) + \frac{\eta}{2}\|\hat{g}_t\|_2^2. \tag{5.17}$$

From the Pythagorean theorem (see, e.g., Theorem 2.1 in [77]), since $x^* \in D$, we have $\|x_t - \eta \hat{g}_t - x^*\|_2 \geq \|P_D(x_t - \eta \hat{g}_t) - x^*\|_2 = \|x_{t+1} - x^*\|_2$. From this and (5.17), we have

$$\hat{g}_t^\top (x_t - x^*) \leq \frac{1}{2\eta}(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2) + \frac{\eta}{2}\|\hat{g}_t\|_2^2.$$

By taking summation of this for $t = 1, 2, \ldots, T$, we obtain

$$\begin{aligned}
\sum_{t=1}^{T} \hat{g}_t^\top (x_t - x^*) &\leq \frac{1}{2\eta}\sum_{t=1}^{T}(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|\hat{g}_t\|_2^2 \\
&= \frac{1}{2\eta}(\|x_1 - x^*\|_2^2 - \|x_{T+1} - x^*\|_2^2) + \frac{\eta}{2}\sum_{t=1}^{T}\|\hat{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta}\max_{x \in D}\|x\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\hat{g}_t\|_2^2,
\end{aligned}$$

where the last inequality follows from $x_1 = 0$, $x^* \in D$ and $\|x_{T+1} - x^*\|_2^2 \geq 0$. From this and (5.16), we have

$$\mathbf{E}[\tilde{f}(\bar{x})] - \tilde{f}(x^*) \leq \frac{1}{T}\left(\frac{1}{2\eta}\max_{x \in D}\|x\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\mathbf{E}[\|\hat{g}_t\|_2^2]\right).$$

Since this holds for arbitrary $x^* \in D$, we have (5.13). □

A similar analysis can be found in, e.g., Lemma 11 of [78]. When setting $D = \tilde{L} - \frac{1}{2}\cdot \mathbf{1}$, we have $\max_{x \in D}\|x\|_2^2 \leq \frac{n}{4}$. From this, Theorems 5.5.1 and 5.5.2, if $\hat{g}_t$ is bounded as $\mathbf{E}[\|\hat{g}_t\|_2^2] \leq G^2$ for all t, we then have $\mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*) \leq \frac{1}{T}\left(\frac{n}{8\eta} + \frac{\eta}{2}G^2 T\right)$. The performance of the algorithm here depends on G, an upper bound on the expected norm of unbiased estimator $\hat{g}_t$. We evaluate the magnitude of G for specific examples of $\hat{g}_t$, in the following subsection.

5.5.3 Unbiased Estimators of Subgradients

In this subsection, we present two different ways to construct unbiased estimators for a subgradient of $\tilde{f}$ that are based on (5.11) and (5.10), respectively. The latter is available for the case of multiple-point feedback, i.e., $k \geq 2$, and produces a smaller error bound under Assumption 5.3.1. Without such an assumption, the former gives a better error bound. Given $x_t = (x_{t1}, x_{t2}, \dots, x_{tn})^\top \in [0,1]^n$, let $\sigma : [n] \to [n]$ be a permutation over $[n]$ for which $x_{t\sigma(1)} \geq x_{t\sigma(2)} \geq \cdots \geq x_{t\sigma(n)}$.

An estimator based on the expression (5.11) Suppose $k \in [n+1]$. Consider choosing queries $\{X_t^{(j)}\}_{j=1}^k$ randomly as follows: Choose a subset $I_t = \{i_t^{(j)}\}_{j=1}^k \subseteq \{0, 1, \dots, n\}$ of size k, uniformly at random, i.e., I_t follows a uniform distribution over the subset family $\{I \subseteq \{0, 1, \dots, n\} \mid |I| = k\}$. Then let $X_t^{(j)} = S_\sigma(i_t^{(j)}) = \{\sigma(j) \mid j \leq i_t^{(j)}\}$ and observe $\hat{f}_t(X_t^{(j)})$ for $j \in [k]$. Define $\hat{g}_t$ as

$$\hat{g}_t = \tfrac{n+1}{k} \textstyle\sum_{j=1}^k \hat{f}_t(X_t^{(j)}) \psi_\sigma(i_t^{(j)}) = \tfrac{n+1}{k} \textstyle\sum_{i \in I_t} \hat{f}_t(S_\sigma(i)) \psi_\sigma(i), \tag{5.18}$$

where $\psi_\sigma(i)$ is defined in (5.9). Note that $\hat{g}_t$ relies on x_t since σ depends on x_t. Then, $\hat{g}_t$ is an unbiased estimator of a subgradient and satisfies $\mathbf{E}[\|\hat{g}_t\|_2^2] = O(n^2/k)$:

Lemma 5.5.1. *Suppose that $\hat{g}_t$ is given by* (5.18)*. We then have*

$$\mathbf{E}[\hat{g}_t | x_t] \in \partial \tilde{f}(x_t), \quad \mathbf{E}[\|\hat{g}_t\|_2^2] \leq 2(n+1)(n+k)/k. \tag{5.19}$$

Proof. From the definition (5.18) of $\hat{g}_t$, its expectation may be expressed as

$$\begin{aligned} \mathbf{E}[\hat{g}_t | x_t] &= \frac{n+1}{k} \mathbf{E}\left[\sum_{i \in I_t} \hat{f}_t(S_\sigma(i)) \psi_\sigma(i) \right] = \frac{n+1}{k} \sum_{i=0}^n \mathrm{Prob}[i \in I_t] \, \mathbf{E}\left[\hat{f}_t(S_\sigma(i)) \right] \psi_\sigma(i) \\ &= \frac{n+1}{k} \sum_{i=0}^n \mathrm{Prob}[i \in I_t] f_t(S_\sigma(i)) \psi_\sigma(i), \end{aligned} \tag{5.20}$$

where the last equality comes from the assumption of $\mathbf{E}[\hat{f}_t(X)] = f(X)$. Since I_t is chosen uniformly at random from all subsets of $\{0, 1, \dots, n\}$ having size k, for each $i \in \{0, 1, \dots, n\}$, the probability that $i \in I_t$ is $\binom{n}{k-1} / \binom{n+1}{k} = \frac{k}{n+1}$. Substituting this into (5.20), we obtain $\mathbf{E}[\hat{g}_t | x_t] = \sum_{i=0}^n f_t(S_\sigma(i)) \psi_\sigma(i)$. This vector is equal to $g(\sigma)$ given in (5.11), which is a subgradient of $\tilde{f}$ at x_t. We next evaluate the expectation of $\|\hat{g}_t\|_2^2$. From the definition (5.18) of $\hat{g}_t$, we have

$$\begin{aligned} \mathbf{E}[\|\hat{g}_t\|_2^2] &= \frac{(n+1)^2}{k^2} \mathbf{E}\left[\sum_{i \in I_t} \sum_{j \in I_t} \hat{f}_t(S_\sigma(i)) \hat{f}_t(S_\sigma(j)) \psi_\sigma(i)^\top \psi_\sigma(j) \right] \\ &= \frac{(n+1)^2}{k^2} \sum_{i=0}^n \sum_{j=0}^n \mathrm{Prob}[i, j \in I_t] \, \mathbf{E}\left[\hat{f}_t(S_\sigma(i)) \hat{f}_t(S_\sigma(j)) \right] \psi_\sigma(i)^\top \psi_\sigma(j) \\ &\leq \frac{(n+1)^2}{k^2} \sum_{i=0}^n \sum_{j=0}^n | \mathrm{Prob}[i, j \in I_t] \psi_\sigma(i)^\top \psi_\sigma(j) |, \end{aligned} \tag{5.21}$$

where the last inequality follows from the assumption that $\hat{f}_t(X) \in [-1, 1]$. From the definition of I_t, we have

$$\mathrm{Prob}[i, j \in I_t] = \begin{cases} \binom{n}{k-1} / \binom{n+1}{k} = \frac{k}{n+1} & (i = j) \\ \binom{n-1}{k-2} / \binom{n+1}{k} = \frac{k(k-1)}{n(n+1)} & (i \neq j) \end{cases}. \tag{5.22}$$

Further, from the definition (5.9) of $\psi_\sigma(i)$, we have

$$\psi_\sigma(i)^\top \psi_\sigma(j) = \begin{cases} 2 & (i = j) \\ -1 & (|i-j| = 1) \\ 0 & (|i-j| \geq 2) \end{cases} . \tag{5.23}$$

Combining (5.21), (5.22) and (5.23), we have

$$\begin{aligned}
\mathbf{E}[\|\hat{g}_t\|_2^2] &\leq \frac{(n+1)^2}{k^2}\left(\sum_{i=0}^{n} \mathrm{Prob}[i \in I_t]\cdot 2 + \sum_{i=0}^{n-1} \mathrm{Prob}[i, i+1 \in I_t] + \sum_{i=1}^{n} \mathrm{Prob}[i, i-1 \in I_t]\right) \\
&= \frac{(n+1)^2}{k^2}\left(\sum_{i=0}^{n} \frac{2k}{n+1} + \sum_{i=0}^{n-1} \frac{k(k-1)}{n(n+1)} + \sum_{i=1}^{n} \frac{k(k-1)}{n(n+1)}\right) \\
&= \frac{(n+1)^2}{k^2}\left(2k + \frac{2k(k-1)}{n+1}\right) = \frac{2(n+1)(n+k)}{k},
\end{aligned}$$

which proves (5.19). □

An estimator based on the expression (5.10) Suppose $2 \leq k \leq n+1$ holds, and let l denote $l = \lfloor k/2 \rfloor \geq 1$. Consider choosing queries $\{X_t^{(j)}\}_{j=1}^k$ randomly as follows: Choose a subset $J_t \subseteq \{1, \ldots, n\}$ of size l, uniformly at random. Then, set queries $\{X_t^{(j)}\}_{j=1}^k$ so that $\bigcup_{i \in J_t}\{S_\sigma(i), S_\sigma(i-1)\} \subseteq \{X_t^{(j)}\}_{j=1}^k$, and observe $\hat{f}_t(S_\sigma(i))$ and $\hat{f}_t(S_\sigma(i-1))$ for $i \in J_t$. Define $\hat{g}_t$ as

$$\hat{g}_t = \tfrac{n}{l} \textstyle\sum_{i \in J_t} (\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1))\chi_{\sigma(i)}. \tag{5.24}$$

Then, $\hat{g}_t$ is an unbiased estimator of a subgradient and satisfies $\mathbf{E}[\|\hat{g}_t\|_2^2] = O(n^2/k)$, and if $\hat{f}_t$ is a submodular function, then $\mathbf{E}[\|\hat{g}_t\|_2^2] = O(n/k)$ holds.

Lemma 5.5.2. *Suppose that $\hat{g}_t$ is given by* (5.24)*. We then have*

$$\mathbf{E}[\hat{g}_t | x_t] \in \partial \tilde{f}(x_t), \quad \mathbf{E}[\|\hat{g}_t\|_2^2] \leq 4n^2/l \leq 12n^2/k. \tag{5.25}$$

In addition, if $\hat{f}_t$ is a submodular function, we then have

$$\mathbf{E}[\|\hat{g}_t\|_2^2] \leq 16n/l \leq 48n/k. \tag{5.26}$$

Proof. From the definition (5.24) of $\hat{g}_t$, its expectation may be expressed as

$$\begin{aligned}
\mathbf{E}[\hat{g}_t | x_t] &= \frac{n}{l}\mathbf{E}\left[\sum_{i \in J_t} (\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1))\chi_{\sigma(i)}\right] \\
&= \frac{n}{l}\sum_{i=1}^{n} \mathrm{Prob}[i \in J_t]\,\mathbf{E}\left[\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1)\right]\chi_{\sigma(i)} \\
&= \sum_{i=1}^{n} (f_t(S_\sigma(i)) - f_t(S_\sigma(i-1))\chi_{\sigma(i)},
\end{aligned}$$

where the last equality follows from the fact that $\mathrm{Prob}[i \in J_t] = \binom{n-1}{l-1}/\binom{n}{l} = \frac{l}{n}$ holds for all $i \in [n]$ and the assumption of $\mathbf{E}[\hat{f}_t(X)] = f(X)$. This vector is equal to the $g(\sigma)$ given in

Algorithm 8 An algorithm for submoudular function minimization with noisy evaluation oracle

```
Require: The size $n \geq 1$ of the problem, the number $T \geq 1$ of oracle calls, the number
    $k \in [n+1]$ of feedback values per oracle call, and the learning rate $\eta > 0$.
 1: Set $x_1 = \frac{1}{2} \cdot \mathbf{1}$.
 2: for $t = 1, 2, \ldots, T$ do
 3:   Let $\sigma : [n] \to [n]$ be a permutation corresponding to $x_t$, i.e., $x_{t\sigma(1)} \geq \cdots \geq x_{t\sigma(n)}$.
 4:   if Assumption 5.3.1 holds then
 5:     Choose a subset $J_t \subseteq [n]$ of size $l = \lfloor k/2 \rfloor$, uniformly at random.
 6:     Call the evaluation oracle $\hat{f}_t$ to observe $\hat{f}_t(S_\sigma(i))$ and $\hat{f}_t(S_\sigma(i-1))$ for $i \in J_t$.
 7:     Construct an unbiased estimator $\hat{g}_t$ of a subgradient of $\tilde{f}$ at $x_t$, as (5.24).
 8:   else
 9:     Choose a subset $I_t \subseteq \{0, 1, \ldots, n\}$ of size $k$, uniformly at random.
10:     Call the evaluation oracle $\hat{f}_t$ to observe $\hat{f}_t(S_\sigma(i))$ for $i \in I_t$.
11:     Construct an unbiased estimator $\hat{g}_t$ of a subgradient of $\tilde{f}$ at $x_t$, as (5.18).
12:   end if
13:   Compute $x_{t+1}$ from $x_t$ and $\hat{g}_t$ on the basis of (5.12).
14: end for
15: Set $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$.
16: Draw $u$ from a uniform distribution over $[0,1]$, and output $\hat{X} = H_{\bar{x}}(u) = \{i \in [n] \mid \bar{x}_i \geq u\}$.
```

(5.10), which is a subgradient of $\tilde{f}$ at x_t. We next evaluate the expectation of $\|\hat{g}_t\|_2^2$. From the definition (5.24) of $\hat{g}_t$, we have

$$\begin{aligned}
\mathbf{E}[\|\hat{g}_t\|_2^2] &= \frac{n^2}{l^2}\,\mathbf{E}\left[\sum_{i \in J_t}\sum_{j \in J_t}(\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1)))(\hat{f}_t(S_\sigma(j)) - \hat{f}_t(S_\sigma(j-1)))\chi_{\sigma(i)}^\top \chi_{\sigma(j)}\right] \\
&= \frac{n^2}{l^2}\sum_{i=1}^{n}\mathrm{Prob}[i \in J_t]\,\mathbf{E}\left[(\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1)))^2\right] \\
&= \frac{n}{l}\sum_{i=1}^{n}\mathbf{E}\left[(\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1)))^2\right] \leq \frac{4n^2}{l},
\end{aligned} \tag{5.27}$$

where the second equality comes from that $\chi_i^\top \chi_j = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$, the third equality comes from that $\mathrm{Prob}[i \in J_t] = \frac{l}{n}$, and the inequality follows from the assumption that $\hat{f}_t(X) \in [-1, 1]$. From this and the fact that $l = \lfloor k/2 \rfloor \geq k/3$, (5.25) follows. If $\hat{f}_t$ is a submodular function, from Lemma 8 in [78], or Lemma 1 in [96], we have $\sum_{i=1}^{n}(\hat{f}_t(S_\sigma(i)) - \hat{f}_t(S_\sigma(i-1)))^2 \leq 16$. From this and (5.27), we obtain (5.26). □

A key factor in the advantage of the estimator defined by (5.24) is that the vector $(f_t(S_\sigma(i)) - f_t(S_\sigma(i-1)))_{i=1}^{n} \in \mathbb{R}^n$ has a smaller norm than $(f_t(S_\sigma(i)))_{i=0}^{n} \in \mathbb{R}^{n+1}$, which is implied by Lemma 8 in [78] or Lemma 1 in [96].

5.5.4 Proof of Error Upper Bound

By combining SGD described in Section 5.5.2 and unbiased estimators defined by (5.18) or (5.24), we obtain Algorithm 8. Let us evaluate the additive errors for this algorithm. Note that we have $\mathbf{E}[\tilde{f}(\bar{x})] - \min_{x* \in \tilde{L}} \tilde{f}(x^*) = \mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*)$ from (5.6) and Theorem 5.5.1.

Suppose $\hat{X}$ is produced by Algorithm 8 in which Steps 9–11 are chosen. From Theorem 5.5.2 and Lemma 5.5.1, we have $\mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*) \leq \frac{1}{T}\left(\frac{n}{8\eta} + \frac{\eta T(n+1)(n+k)}{k}\right)$. The right-hand side is minimized when η is chosen as $\eta = \sqrt{\frac{kn}{8T(n+1)(n+k)}}$. We then have $\mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*) \leq \sqrt{\frac{n(n+1)(n+k)}{2kT}} = O(\frac{n^{3/2}}{kT})$, which proves (5.2).

Suppose that Assumption 5.3.1 holds and that $\hat{X}$ is produced by Algorithm 8, where Steps 5–7 are chosen. From Theorem 5.5.2 and Lemma 5.5.1, we have $\mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*) \leq \frac{1}{T}\left(\frac{n}{8\eta} + \frac{24\eta T n}{k}\right)$. The right-hand side is minimized when η is chosen as $\eta = \sqrt{\frac{k}{192T}}$. We then have $\mathbf{E}[f(\hat{X})] - \min_{X^* \in L} f(X^*) \leq \frac{\sqrt{12}n}{\sqrt{kT}} = O(\frac{n}{\sqrt{kT}})$, which proves (5.3).

The computational time of Algorithm 8 can be evaluated as follows: Step 3 can be conducted by a sorting algorithm, which takes $O(n \log n)$ time. Step 5 can be done by generating uniform random numbers over $[m]$ for $m = n, n-1, \ldots, n-k+1$, which takes $O(k \log n)$ times. Step 6 requires $O(k\mathrm{EO})$ time computation. Step 7 can be computed with $O(n)$ arithmetic operations. Steps 9–11 are similar to Steps 5–7. Step 13 takes $O(n)$ time since $x_t - \eta \hat{g}_t$ can be computed with $O(n)$ arithmetic operations and since $P_{\tilde{L}}(x)$ has an explicit form. Hence, Steps 2–14 require $O((n \log n + k \log n + k\mathrm{EO} + n + n) \cdot T) = O((k\mathrm{EO} + n \log n)T)$ time. The other steps do not require time greater than this. Therefore, the overall time complexity is of $O((k\mathrm{EO} + n \log n)T)$.

5.6 Lower Bound

5.6.1 Construction of Hard Instance

Define $h_i : 2^{[n]} \to \{-1, 1\}$ as $h_i(X) := \begin{cases} -1 & (i \in X) \\ 1 & (i \notin X) \end{cases}$. for $i \in [n]$. Fix a subset $S^* \subseteq [n]$ and a positive real value $\varepsilon \in [0,1]$. Consider the following procedure that produces a function $\hat{f} : 2^{[n]} \to \{-1, 1\}$: (1) Choose $i \in [n]$ uniformly at random, and set $s = 1$ with probability $\frac{1-\varepsilon}{2}$, $s = -1$ with probability $\frac{1+\varepsilon}{2}$. (2) Define $\hat{f} : 2^{[n]} \to \{-1, 1\}$ by $\hat{f}(X) = s \cdot h_i(S^* \triangle X) = s \cdot h_i(S^*) h_i(X)$, where $S^* \triangle X$ stands for the symmetric difference between S^* and X, i.e., $S^* \triangle X = (S^* \setminus X) \cup (X \setminus S^*)$. Let $F(S^*, \varepsilon)$ denote the distribution of functions generated by the above procedure. A similar construction can be found in [52], which is for a lower bound of bandit linear optimization.

In addition, define $F'(S^*, \varepsilon)$ similarly, so that all function values of $f \sim F'(S^*, \varepsilon)$ are *stochastically independent*: Choose $i_X \in [n]$ and s_X with the probability defined as the above, independently for all $X \subseteq [n]$, and define $\hat{f}(X) = s_X \cdot h_{i_X}(S^*) h_i(X)$. Let $F'(S^*, \varepsilon)$ denote the distribution of functions generated by this procedure. Note that each $\hat{f}$ generated from $F(S^*, \varepsilon)$ is a modular function and that this does not always hold for $F'(S^*, \varepsilon)$. If $D_{S^*} = F(S^*, \varepsilon)$ or if $D_{S^*} = F'(S^*, \varepsilon)$, the expectation of $\hat{f} \sim D_{S^*}$ is a submodular function expressed as

$$f_{S^*,\varepsilon}(X) := \mathop{\mathbf{E}}_{\hat{f} \sim D_{S^*}}[\hat{f}(X)] = -\frac{\varepsilon}{n} \textstyle\sum_{i=1}^{n} h_i(S^*) h_i(X) = \frac{\varepsilon}{n}(2|S^* \triangle X| - n), \tag{5.28}$$

where the second equality comes from $\mathbf{E}[s] = \mathbf{E}[s_X] = \frac{1-\varepsilon}{2} - \frac{1+\varepsilon}{2} = -\varepsilon$.

5.6.2 Proof of Error Lower Bound

To prove Theorem 5.4.2, we start with bounding the additive error from below by means of KL divergences. Fix $X^{(1)}, X^{(2)}, \ldots, X^{(k)} \subseteq [n]$ arbitrarily. For a class $\{D_{S^*} \mid S^* \subseteq [n]\}$ of distributions over $\{\hat{f} : 2^{[n]} \to \{-1,1\}\}$, let P_{S^*} denote the distribution of $y(\hat{f}) = (\hat{f}(X^{(1)}), \hat{f}(X^{(2)}) \ldots, \hat{f}(X^{(k)}))^\top \in \mathbb{R}^k$ for $\hat{f} \sim D_{S^*}$. We then have the following:

Lemma 5.6.1. *Suppose that a class of distributions $\{D_{S^*} \mid S^* \subseteq [d]\}$ satisfies (5.28) for all $S^* \subseteq [d]$. In addition, suppose that the following holds for arbitrary $S^*, X^{(1)}, X^{(2)}, \ldots, X^{(k)} \subseteq [n]$:*

$$\textstyle\sum_{i=1}^n D_{\mathrm{KL}}(P_{S^*} || P_{S^* \triangle \{i\}}) \leq \frac{n}{2T}. \tag{5.29}$$

If S^* is chosen uniformly at random from $2^{[n]}$, and $\hat{f}_t$ follows D_{S^*} i.i.d. for $t = 1, 2, \ldots, T$, then any algorithm suffers an additive error of $\mathbf{E}[E_T] = \mathbf{E}\left[f_{S^*,\varepsilon}(\hat{X}) - \min_{S \in 2^{[n]}} f_{S^*,\varepsilon}(S)\right] \geq \frac{\varepsilon}{2}$, where the expectation is taken w.r.t. S^*, $\hat{f}_t$, and the internal randomness of algorithms.

Proof. Since any randomized algorithms can be regarded as a convex combination of deterministic algorithms, it suffices to consider only deterministic algorithms. Fix a deterministic algorithm and let $\{(X_t^{(1)}, \ldots, X_t^{(k)})\}_{t=1}^T$ denote the queries generated by it. Denote $y_t = (\hat{f}_t(X_t^{(1)}), \ldots, \hat{f}_t(X_t^{(k)}))^\top \in \{-1,1\}^k$, the input to the algorithm. From (5.28), we have

$$f_{S^*,\varepsilon}(\hat{X}) = -\frac{\varepsilon}{n} \sum_{i=1}^n h_i(S^* \triangle \hat{X}). \tag{5.30}$$

To evaluate the above value, we fix $i \in [n]$, and focus on $\mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^*}}[h_i(S^* \triangle \hat{X})]\right]$. Since $S^* \triangle \{i\}$ follows a uniform distribution over $2^{[n]}$, the same distribution as of S^*, we have

$$\mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^*}}[h_i(S^* \triangle \hat{X})]\right] = \mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^* \triangle \{i\}}}[h_i((S^* \triangle \{i\}) \triangle \hat{X})]\right] = -\mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^* \triangle \{i\}}}[h_i(S^* \triangle \hat{X})]\right]$$

where the second equality comes from the definition of h_i. Hence, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^*}}[h_i(S^* \triangle \hat{X})]\right] &= \frac{1}{2}\mathop{\mathbf{E}}_{S^*}\left[\mathop{\mathbf{E}}_{f_t \sim D_{S^*}}[h_i(S^* \triangle \hat{X})] - \mathop{\mathbf{E}}_{f_t \sim D_{S^* \triangle \{i\}}}[h_i(S^* \triangle \hat{X})]\right] \\
&= \frac{1}{2}\mathop{\mathbf{E}}_{S^*}\left[h_i(S^*)\left(\mathop{\mathbf{E}}_{f_t \sim D_{S^*}}[h_i(\hat{X})] - \mathop{\mathbf{E}}_{f_t \sim D_{S^* \triangle \{i\}}}[h_i(\hat{X})]\right)\right],
\end{aligned} \tag{5.31}$$

where the second equality comes from $h_i(S \triangle S') = h_i(S) h_i(S')$. Since $\hat{X}$ is determined by $Y^T = (y_1, \ldots, y_T) \in \{-1,1\}^{k \times T}$, there is a function $\phi_i : \{-1,1\}^{k \times T} \to \{-1,1\}$ such that $h_i(\hat{X}) = \phi_i(Y^T)$. Let Q and Q_i' denote the probability distributions of Y^T for $f_t \sim F(S^*, \varepsilon)$

and $f_t \sim D(S^* \triangle \{i\}, \varepsilon)$, respectively. We then have

$$\begin{aligned}
&\left| \mathop{\mathbf{E}}_{f_t \sim D_{S^*}} [h_i(\hat{X})] - \mathop{\mathbf{E}}_{f_t \sim D_{S^* \triangle \{i\}}} [h_i(\hat{X})] \right| \\
&= \left| \sum_{y \in \{-1,1\}^T} Q(y)\phi_i(y) - \sum_{y \in \{-1,1\}^T} Q'_i(y)\phi_i(y) \right| \\
&= \left| \sum_{y \in \{-1,1\}^T} (Q(y) - Q'_i(y))\phi_i(y) \right| \\
&\leq \sum_{y \in \{-1,1\}^T} |Q(y) - Q'_i(y)| = \|Q - Q'_i\|_1 \qquad (5.32)
\end{aligned}$$

where $\|Q - Q'_i\|_1$ stands for the total variation distance between Q and Q'_i. From Pinsker's inequality (see, e.g., Theorem 12.6.1 in [48]), the total variation distance can be bounded by means of the KL divergence as

$$\|Q - Q'_i\|_1 \leq \sqrt{2D_{\mathrm{KL}}(Q||Q'_i)}. \qquad (5.33)$$

Combining equations (5.30) – (5.33) and $|h_i(S^*)| = 1$, we have

$$-\mathop{\mathbf{E}}_{S^*, f_t \sim D_{S^*}} [f_{S^*,\varepsilon}(\hat{X})] \leq \frac{\varepsilon}{2n} \mathop{\mathbf{E}}_{S^*} \left[\sum_{i=1}^n \sqrt{2D_{\mathrm{KL}}(Q||Q'_i)} \right] \leq \frac{\varepsilon}{2} \mathop{\mathbf{E}}_{S^*} \left[\sqrt{\frac{2}{n} \sum_{i=1}^n D_{\mathrm{KL}}(Q||Q'_i)} \right], \qquad (5.34)$$

where the second inequality follows from the Jensen's inequality and the concavity of $\sqrt{x}$. Denote $Y^t = (y_1, \ldots, y_t)$. From the chain rule of KL divergence (see, e.g., Theorem 2.5.3 in [48]), we have

$$\sum_{i=1}^n D_{\mathrm{KL}}(Q||Q'_i) = \sum_{t=1}^T \mathop{\mathbf{E}}_{Y^{t-1} \sim Q} \left[\sum_{i=1}^n \mathop{D_{\mathrm{KL}}}_{y_t \sim Q|_{Y^{t-1}},\, y'_t \sim Q'_i|_{Y^{t-1}}} (y_t || y'_t) \right]. \qquad (5.35)$$

When Y^{t-1} is fixed, $X_t^{(1)}, \ldots, X_t^{(k)}$ are also fixed since we have fixed a deterministic algorithm. Hence, from the assumption of (5.29), we have

$$\sum_{i=1}^n \mathop{D_{\mathrm{KL}}}_{y_t \sim Q|_{Y^{t-1}},\, y'_t \sim Q'_i|_{Y^{t-1}}} (y_t || y'_t) \leq \frac{n}{2T}.$$

By combining this, (5.34), and (5.35), we have

$$-\mathop{\mathbf{E}}_{S^*, f_t \sim D_{S^*}} [f_{S^*,\varepsilon}(\hat{X})] \leq \frac{\varepsilon}{2}.$$

We also here have

$$\min_{S \in 2^{[n]}} f_{S^*,\varepsilon}(S) = f_{S^*,\varepsilon}(S^*) = -\varepsilon.$$

From the above two inequalities, the expected additive error is bounded as

$$\mathbf{E}\left[f_{S^*,\varepsilon}(\hat{X}) - \min_{S \in 2^{[n]}} f_{S^*,\varepsilon}(S) \right] \geq -\frac{\varepsilon}{2} + \varepsilon = \frac{\varepsilon}{2},$$

which accomplishes the proof. □

Intuitively, the condition (5.29) means that the distribution of the observed values y does not change much even if the optimal solution S^* is perturbed. Consequently, under the condition (5.29), it is hard for any algorithm to detect S^*. Sufficient conditions for (5.29) are given in the following two lemmas:

Lemma 5.6.2. *Suppose that* $\{P_{S^*}\}$ *is defined by* $D_{S^*} = F'(S^*, \varepsilon)$ *for* $0 \leq \varepsilon \leq \min\{\frac{1}{6}, \frac{n}{\sqrt{8kT}}\}$*. Then* (5.29) *holds for arbitrary* $S^*, X^{(1)}, \ldots, X^{(k)} \subseteq [n]$.

Proof. Suppose that $X^{(1)}, \ldots, X^{(k)}$ are distinct. Then, since $\hat{f}(X^{(j)})$ with $\hat{f} \sim F'(S^*, \varepsilon)$ are stochastically independent for $j = 1, \ldots, k$, we have

$$D_{\mathrm{KL}}(P_{S^*} || P_{S^* \triangle \{i\}}) = \sum_{j=1}^{k} \underset{\hat{f} \sim F'(S^*, \varepsilon), \hat{f}' \sim F'(S^* \triangle \{i\}, \varepsilon)}{D_{\mathrm{KL}}} (\hat{f}(X^{(j)}) || \hat{f}'(X^{(j)})). \tag{5.36}$$

From the definition of $F'(S^*, \varepsilon)$, $\hat{f}(X^{(j)})$ follows Bernoulli distributions of parameters $\theta := \frac{1}{2} + \frac{\varepsilon}{2n}(2|S^* \triangle X_t| - n)$ and $\theta' := \frac{1}{2} + \frac{\varepsilon}{2n}(2|(S^* \triangle \{i\}) \triangle X_t| - n)$ for $\hat{f} \sim F(S^*, \varepsilon)$ and $\hat{f} \sim F(S^* \triangle \{i\}, \varepsilon)$, respectively. Since $\theta, \theta' \in [\frac{1}{3}, \frac{2}{3}]$ and $|\theta - \theta'| = \frac{\varepsilon}{n} \leq \frac{1}{6}$, we have

$$\begin{aligned}
&\underset{\hat{f} \sim F'(S^*, \varepsilon), \hat{f}' \sim F'(S^* \triangle \{i\}, \varepsilon)}{D_{\mathrm{KL}}} (\hat{f}(X^{(j)}) || \hat{f}'(X^{(j)})) = -\theta \log \frac{\theta'}{\theta} - (1 - \theta) \log \frac{1 - \theta'}{1 - \theta} \\
&\leq -\theta \left(\frac{\theta' - \theta}{\theta} - \frac{4}{5} \left(\frac{\theta' - \theta}{\theta} \right)^2 \right) - (1 - \theta) \left(\frac{\theta - \theta'}{1 - \theta} - \frac{4}{5} \left(\frac{\theta - \theta'}{1 - \theta} \right)^2 \right) \\
&= \frac{4}{5}(\theta' - \theta)^2 \left(\frac{1}{\theta} + \frac{1}{1 - \theta} \right) \leq \frac{4\varepsilon^2}{n^2} \leq \frac{1}{2kT},
\end{aligned} \tag{5.37}$$

where the first inequality follows from $-\log(1 + x) \leq -x + \frac{4}{5}x^2$ for $|x| \leq \frac{1}{2}$ and the last inequality follows from the assumption of $\varepsilon \leq \frac{n}{\sqrt{8kT}}$. By combining (5.36) and (5.37), we obtain $D_{\mathrm{KL}}(P_{S^*} || P_{S^* \triangle \{i\}}) \leq \frac{1}{2T}$ for all $i \in [n]$, which implies that (5.29) holds. For the case that $X^{(1)}, \ldots, X^{(k)}$ are not distinct, i.e., when $\{X^{(1)}, \ldots, X^{(k)}\}$ consists of $k' < k$ elements, we can show $D_{\mathrm{KL}}(P_{S^*} || P_{S^* \triangle \{i\}}) \leq \frac{k'}{2kT}$ in the same way, from which (5.29) follows. □

Lemma 5.6.3. *Suppose that* $\{P_{S^*}\}$ *is given by* $D_{S^*} = F(S^*, \varepsilon)$ *for a nonnegative* ε *such that* $\varepsilon \leq \min\{\frac{1}{6}, n\sqrt{\frac{5}{24T \min\{2^k, 2n\}}}\}$*. Then* (5.29) *holds for arbitrary* $S^*, X^{(1)}, \ldots, X^{(k)} \subseteq [n]$.

Proof. From the definition of $F(S^*, \varepsilon)$, $\hat{f} \sim F(S^*, \varepsilon)$ can be expressed as $\hat{f}(X) = V \cdot h_I(S^* \triangle X)$, where I and V follows distributions over $[n]$ and $\{-1, 1\}$, respectively. Hence, if S^* and $X^{(1)}, \ldots, X^{(k)}$ are fixed, $y(\hat{f}) := (\hat{f}(X^{(1)}), \ldots, \hat{f}(X^{(k)})) \in \{-1, 1\}^k$ changes depending only on I and V. Therefore, there exists a function $\lambda : [n] \times \{-1, 1\} \to \{-1, 1\}^k$ such that $y(\hat{f}) = \lambda(I, V)$. Denote $Z = \mathrm{range}(\lambda) = \{\lambda(i, s) \in \{-1, 1\}^k \mid i \in [n], s \in \{-1, 1\}\}$. Then we have

$$|Z| \leq \min\{|\{-1, 1\}^k|, |[n] \times \{-1, 1\}|\} = \min\{2^k, 2n\}. \tag{5.38}$$

From the definition of $F'(S^*, \varepsilon)$, for any fixed $z \in Z$ and $i \in [n]$, we have

$$\left| \underset{\hat{f} \sim F(S^*, \varepsilon)}{\mathrm{Prob}} [y(\hat{f}) = z] - \underset{\hat{f} \sim F(S^* \triangle \{i\}, \varepsilon)}{\mathrm{Prob}} [y(\hat{f}) = z] \right| = \begin{cases} \frac{\varepsilon}{n} & (z \in \{\lambda(i, -1), \lambda(i, 1)\}) \\ 0 & (z \notin \{\lambda(i, -1), \lambda(i, 1)\}) \end{cases}. \tag{5.39}$$

From this and the definition of the KL divergence, we have

$$
\begin{aligned}
D_{\mathrm{KL}}(P_{S^*} \| P_{S^* \triangle \{i\}}) &= - \sum_{z \in Z} \operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z] \log \frac{\operatorname*{Prob}_{\hat{f} \sim F(S^* \triangle \{i\}, \varepsilon)} [y(\hat{f}) = z]}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z]} \\
&\leq \frac{4}{5} \sum_{z \in Z} \frac{\left(\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y_t = y] - \operatorname*{Prob}_{\hat{f} \sim F(S^* \triangle \{i\}, \varepsilon)} [y(\hat{f}) = z] \right)^2}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z]} \\
&= \frac{4\varepsilon^2}{5n^2} \left(\frac{1}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = \lambda(i, -1)]} + \frac{1}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = \lambda(i, 1)]} \right),
\end{aligned}
$$

where the inequality follows from a calculation used in (5.37), and the last equality follows from (5.39). By taking a sum of the above for $i \in [n]$, we obtain the following:

$$
\begin{aligned}
\sum_{i=1}^{n} D_{\mathrm{KL}}(P_{S^*} \| P_{S^* \triangle \{i\}}) &\leq \frac{4\varepsilon^2}{5n^2} \sum_{i=1}^{n} \left(\frac{1}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = \lambda(i, -1)]} + \frac{1}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = \lambda(i, 1)]} \right) \\
&= \frac{4\varepsilon^2}{5n^2} \sum_{z \in Z} \left(\frac{|\{i \in [n] \mid \lambda(i, 1) = z\}|}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z]} + \frac{|\{i \in [n] \mid \lambda(i, -1) = z\}|}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z]} \right) \\
&= \frac{4\varepsilon^2}{5n^2} \sum_{z \in Z} \frac{|\{(i, s) \in [n] \times \{-1, 1\} \mid \lambda(i, s) = z\}|}{\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z]}.
\end{aligned}
$$

Since I follows a uniform distribution over $[n]$ and V follows a Bernoulli distribution with the parameter $\frac{1-\varepsilon}{2}$, we have $\mathrm{Prob}[I = i, V = s] \geq \frac{1-\varepsilon}{2n} \geq \frac{1}{3n}$ for all $i \in [n]$ and $s \in \{-1, 1\}$. Hence, we have

$$
\operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [y(\hat{f}) = z] = \operatorname*{Prob}_{\hat{f} \sim F(S^*, \varepsilon)} [\lambda(I, V) = z] \geq \frac{|\{(i, s) \in [n] \times \{-1, 1\} \mid \lambda(i, s) = z\}|}{3n}.
$$

Combining the above two equations, (5.38), and the assumption of $\varepsilon \leq n\sqrt{\frac{5}{24T \min\{2^k, 2n\}}}$, we obtain

$$
\sum_{i=1}^{n} \operatorname*{D_{\mathrm{KL}}}_{y_t \sim P|_{Y^{t-1}},\, y'_t \sim P'_i|_{Y^{t-1}}} (y_t \| y'_t) \leq \frac{4\varepsilon^2}{5n^2} \sum_{y \in Z} 3n = \frac{12\varepsilon^2 |Z|}{5n} \leq \frac{n}{2T}.
$$

□

Theorem 5.4.2 can be proven by combining Lemmas 5.6.1, 5.6.2, and 5.6.3. From Lemmas 5.6.1 and 5.6.2, if S^* is chosen uniformly at random from $2^{[n]}$ and if $\hat{f}_t$ follows $F'(S^*, \varepsilon)$ with $\varepsilon = \min\{\frac{1}{6}, \frac{n}{\sqrt{8kT}}\}$, i.i.d. for $t \in [T]$, we then have $\mathbf{E}[E_T] \geq \frac{\varepsilon}{2} = \min\{\frac{1}{12}, \frac{n}{\sqrt{32kT}}\} = \Omega'(\frac{n}{\sqrt{kT}})$, which proves (5.4). If $k \geq 2$ and if S^* is chosen uniformly at random from $2^{[n]}$, and $\hat{f}_t$ follows $F(S^*, \varepsilon)$ with $\varepsilon = \min\{\frac{1}{6}, n\sqrt{\frac{5}{24T \min\{2^k, 2n\}}}\}$, i.i.d. for $t \in [T]$, then Assumption 5.3.1 is satisfied since $\hat{f}_t \sim F(S^*, \varepsilon)$ is a submodular. Further, from Lemmas 5.6.1 and 5.6.3, we have $\mathbf{E}[E_T] \geq \frac{\varepsilon}{2} = \min\{\frac{1}{12}, n\sqrt{\frac{5}{96T \min\{2^k, 2n\}}}\} = \Omega'(\max\{\frac{n}{\sqrt{T2^k}}, \sqrt{\frac{n}{T}}\}) = \Omega'(\frac{n}{\sqrt{T2^k}} + \sqrt{\frac{n}{T}})$, which proves (5.5).

5.7 Conclusion

We have introduced submodular function minimization with noisy evaluation oracle, and have provided algorithms and lower bounds, which together implies that the proposed algorithms achieve nearly optimal additive errors, modulo $O(\sqrt{n})$ factors. For the special cases of k-point feedback settings, in which $2 \leq k = O(1)$ and each noisy evaluation oracle itself is a submodular function, we have provided a tight error bound. For the other cases, we leave it as an open question to find tight bounds.

Chapter 6

Price Optimization via Submodular Function Minimization

This chapter deals with price optimization, which is to find the best pricing strategy that maximizes revenue or profit, on the basis of demand forecasting models. Though recent advances in regression technologies have made it possible to reveal price-demand relationship of a large number of products, most existing price optimization methods, such as mixed integer programming formulation, cannot handle tens or hundreds of products because of their high computational costs. To cope with this problem, this chapter proposes a novel approach based on network flow algorithms. We reveal a connection between supermodularity of the revenue and cross elasticity of demand. On the basis of this connection, we propose an efficient algorithm that employs network flow algorithms. The proposed algorithm can handle hundreds or thousands of products, and returns an exact optimal solution under an assumption regarding cross elasticity of demand. Even if the assumption does not hold, the proposed algorithm can efficiently find approximate solutions as good as other state-of-the-art methods, as empirical results show.

6.1 Introduction

Price optimization is a central research topic with respect to revenue management in marketing science [130, 141]. The goal is to find the best price strategy (a set of prices for multiple products) that maximizes revenue or profit. There is a lot of literature regarding price optimization [21, 36, 114, 136, 141, 147], and significant success has been achieved in industries such as online retail [56], fast-fashion [36], hotels [114, 120], and airlines [130]. One key component in price optimization is demand modeling, which reveals relationships between price and demand. Though traditional studies have focused more on a single price-demand relationship, such as price elasticity of demand [114, 120] and the law of diminishing marginal utility [130], multi-product relationships such as cross price elasticity of demand [128] have recently received increased attention [36, 136]. Recent advances in regression technologies (non-linear, sparse, etc.) make demand modeling over tens or even hundreds of products possible, and data oriented demand modeling has become more and more important.

Given demand models of multiple products, the role of optimization is to find the best price strategy. Most existing studies for multi-product price optimization employ mixed-integer programming [36, 114, 120] due to the discrete nature of individual prices, but their methods cannot be applied to large scale problems with tens or hundreds of products since their computational costs exponentially increases over increasing numbers of products. Though restricting demand models might make optimization problems tractable [36, 56], such approaches cannot capture complicated price-demand relationships and often result in poor performance. Ito and Fujimaki [88] have recently proposed a *prescriptive price optimization* framework to efficiently solve multi-product price optimization with non-linear demand models. In this prescriptive price optimization, the problem is transformed into a sort of binary quadratic programming problem, and they have proposed an efficient relaxation method based on semi-definite programming (SDP). Although their approach has significantly improved computational efficiency over that of mixed-integer approaches, the computational complexity of their SDP formulation requires $O(M^6)$ in theory, where M is the number of products, and it is not sufficiently scalable for large scale problems with hundreds of products, as our empirical evaluation show in Section 6.5.

The goal of this chapter is to develop an efficient algorithm for large scale multi-product price optimization problems that can handle hundreds of products as well as flexible demand models. Our main technical contributions are two-fold. First, we reveal the connection between submodularity of the revenue and cross elasticity of demand. More specifically, we show that the gross profit function of the prescriptive price optimization is supermodular (i.e., the maximization of the gross profit function is equivalent to the submodular minimization) under the assumption regarding cross elasticity of demand that there are no pairs of complementary goods (we refer to this property as a *substitute-goods property*).[1] On the basis of the submodularity, we propose a practical, efficient algorithm that employs network flow algorithms for minimum cut problems and returns exact solutions for problems with the substitute-goods property. Further, even in cases in which the property does not hold, it can efficiently find approximate solutions by iteratively improving submodular lower bounds. Our empirical results show that the proposed algorithm can successfully handle hundreds of products and derive solutions as good as other state-of-the-art methods, while its computational cost is much cheaper, regardless of whether the substitute-goods property holds or not.

6.2 Literature Review

Our price optimization problems are reduced to binary quadratic problems such as (6.4). It is well known that submodular binary quadratic programming problems can be reduced to minimum cut problems [111], and hence it can be solved by maximum flow algorithms. Also, for unconstrained non-submodular binary quadratic programming problems, there is a lot of literature regarding optimization algorithm using minimum cut, especially in the context of Markov random fields inference or energy minimization in computer vision [25, 26, 27, 69, 110, 158]. Above all, QPBO method [25, 110] and its extensions such as QPBOI method [145] are known to be state-of-the-art methods in terms of scalability and theoretical properties. These QPBO/QPBOI and our method are similar in that they all employ network flow algorithms and derive not only partial/approximate solutions but also lower bounds of the exact optimal

[1]"Complementary goods" and "substitute goods" are terms in economics. A good example of complementary goods might be wine and cheese, i.e., if we discount wine, the sales of cheese will increase. An example of substitute goods might be products of different brands in the same product category. If we discount one product, sales of the other products will decrease.

(minimum) value. Our methods, however, differs from QPBO and its extensions in network structures, accuracy and scalability, as is shown in Section 6.5.

6.3 Submodularity in Cross Elasticity of Demand

Suppose we have M products and a product index is denoted by $i \in \{1, \dots, M\}$. In prescriptive price optimization [88], for a price strategy $p = [p_1, \dots, p_M]^\top$, where p_i is the price of the i-th product, and for external variables $r = [r_1, \dots, r_D]^\top$ such as weather, temperature and days of the week, the sales quantity (demand) for the i-th product is modeled by the following regression formula:

$$q_i(p, r) = \sum_{j=1}^{M} f_{ij}(p_j) + \sum_{t=1}^{D} g_{it}(r_t), \tag{6.1}$$

where f_{ii} expresses the effect of price elasticity of demand, f_{ij} $(i \neq j)$ reflects the effect of cross elasticity, and g_{it} represent how the t-th external variable affect the sales quantity. Note that f_{ij} for all (i, j) can be arbitrary functions, and Eq. (6.1) covers various regression (demand) models, such as linear regression, additive models [156], linear regression models with univariate basis functions, etc. This chapter assumes that the regression models are given using existing methods and focuses its discussion on optimization.

Given $q_i(p)$ for all i and a cost vector $c = [c_1, \dots, c_M]^\top$, and fixed external variables r, the *gross profit* can be represented as

$$\ell(p) = \sum_{i=1}^{M} (p_i - c_i) q_i(p) = \sum_{i=1}^{M} (p_i - c_i) \left(\sum_{j=1}^{M} f_{ij}(p_j) + \sum_{t=1}^{D} g_{it}(r_t) \right). \tag{6.2}$$

The goal of price optimization is to find p maximizing $\ell(p)$. In practice, p_m is often chosen from the finite set $\mathcal{P}_i = \{P_{i1}, \dots, P_{iK}\} \subseteq \mathbb{R}$ of K price candidates, where P_{iK} might be a list price and P_{ik} $(k < K)$ might be discounted prices such as 10%-off, 5%-off, 3%-off. Then, the problem of maximizing the gross profit can be formulated as the following combinatorial optimization problem:

$$\text{Maximize} \quad \ell(p) \quad \text{subject to} \quad p_i \in \mathcal{P}_i. \tag{6.3}$$

It is trivial to show that (6.3) is NP-hard in general.

Let us formally define the "substitute-goods property" as follows.

Definition 6.3.1 (Substitute-Goods Property). The demand model defined by (6.1) of the i-th product is said to satisfy the *substitute-goods property* if f_{ij} is monotone non-decreasing for all $j \neq i$.

The concept of substitute-goods property is practical and important because retailers often deal with substitute goods. Suppose the situation that a retailer decides a price strategy of different brand in the same products category. For example, supermarkets sell milk of different brands and car dealerships sell various types of cars. These products are usually substitute goods. This kind of cross elasticity effect is one of advanced topics in revenue management and is practically important [114, 120, 136]. Our key observation is the connection between the substitute-goods property in marketing science and the supermodularity of the gross profit function, which is formally described in the following proposition.

Proposition 6.3.1. *The gross profit function $\ell : \mathcal{P}_1 \times \cdots \times \mathcal{P}_M \to \mathbb{R}$ is supermodular*[2] *if demand models defined by* (6.1) *for all products satisfies the substitute-goods property.*

Proof. Since the sum of supermodular function is supermodular[64], it suffices to show that ℓ_{ij} and ℓ_i are supermodular for all i and j, under the assumption of the substitute goods property. First, ℓ_i are supermodular because arbitrary univariate functions are supermoludar by the definition. Next, let us show the supermodularity of ℓ_{ij}. Let p_i, p'_i and p_j, p'_j be arbitrary elements of $\mathcal{P}_i$ and $\mathcal{P}_j$, respectively. Denote $\underline{p}_i = \min\{p_i, p'_i\}$, $\overline{p}_i = \max\{p_i, p'_i\}$, $\underline{p}_j = \min\{p_j, p'_j\}$, and $\overline{p}_i = \max\{p_j, p'_j\}$. Without loss of generality, assume $p_i \le p'_i$. If $p_j \le p'_j$, then we have $\ell_{ij}(p_i, p_j) + \ell_{ij}(p'_i, p'_j) - \ell_{ij}(\overline{p}_i, \overline{p}_j) - \ell_{ij}(\underline{p}_i, \underline{p}_j) = 0$. Otherwise, i.e, if $p_j > p'_j$, then we have

$$\begin{aligned}
&\ell_{ij}(p_i, p_j) + \ell_{ij}(p'_i, p'_j) - \ell_{ij}(\overline{p}_i, \overline{p}_j) - \ell_{ij}(\underline{p}_i, \underline{p}_j) \\
&= \ell_{ij}(p_i, p_j) + \ell_{ij}(p'_i, p'_j) - \ell_{ij}(p'_i, p_j) - \ell_{ij}(p_i, p'_j) \\
&= (p_i - p'_i)(f_{ij}(p_j) - f_{ij}(p'_j)) \le 0,
\end{aligned}$$

where the last inequality comes from the substitute goods property. These inequality implies the supermodularity of ℓ_{ij}. □

The above proposition implies that, under the assumption of the substitute-goods property, problem (6.3) can be solved precisely using submodular minimization algorithms, where time complexity is a polynomial in M and K. This fact, however, does not necessarily imply that there exists a practical, efficient algorithm for problem (6.3). Indeed, general submodular minimization algorithms are slow in practice even though their time complexities are polynomial. Further, actual models do not always satisfy the substitute-goods property. We propose solutions to these problems in the next section.

6.4 Submodularity-based Algorithm for Revenue Maximization

6.4.1 Binary Quadratic Programming Formulation

This section shows that problem (6.3) can be reduced to the following binary quadratic programming problem (notations are explained in the latter part of this section):

$$\begin{array}{ll}
\text{Minimize} & x^\top A x + b^\top x \\
\text{subject to} & x = [x_1, \ldots, x_n]^\top \in \{0,1\}^n, \\
& x_u \le x_v \quad ((u,v) \in C),
\end{array} \tag{6.4}$$

Each variable p_i takes P_{ik} if and only if the binary vector $x_i = [x_{i1}, \ldots, x_{i,K-1}]^\top \in \{0,1\}^{(K-1)}$ satisfies:

$$x_i = c_k := [\underbrace{1, \ldots, 1}_{k-1}, \underbrace{0, \ldots, 0}_{K-k}]^\top \quad (k = 1, \ldots, K). \tag{6.5}$$

Also we define $x = [x_1^\top, \ldots, x_M^\top]^\top \in \{0,1\}^{(K-1)M}$ and redefine the indices of the entries of x as $x = [x_1, x_2, \ldots, x_{(K-1)M}]$, i.e. $x_{i,k} = x_{i(K-1)+k}$ for notational simplicity.

[2]We say that a function $f : \mathcal{D}_1 \times \cdots \times \mathcal{D}_n \to \mathbb{R}$ $(\mathcal{D}_j \subseteq \mathbb{R})$ is *submodular* if $f(x) + f(y) \le f(x \vee y) + f(x \wedge y)$ for all x, y, where $x \vee y$ and $x \wedge y$ denote the coordinate-wise maximum and minimum, respectively. We say a function f is *supermodular* if $-f$ is submodular.

Defining $\ell_{ij} : \mathcal{P}_i \times \mathcal{P}_j \to \mathbb{R}$ by $\ell_{ij}(p_i, p_j) = (p_i - c_i) f_{ij}(p_j)$ for $i \neq j$ and $\ell_i : \mathcal{P}_i \to \mathbb{R}$ by $\ell_i(p_i) = (p_i - c_i)(f_{ii}(p_i) + \sum_{t=1}^{D} g_{it}(r_t))$, we can express ℓ as

$$\ell(p) = \sum_{1 \le i,j \le M, i \neq j} \ell_{ij}(p_i, p_j) + \sum_{i=1}^{M} \ell_i(p_i). \tag{6.6}$$

Using x_i, we can construct matrices $A_{ij} \in \mathbb{R}^{(K-1)\times(K-1)}$ for which it holds that

$$\ell_{ij}(p_i, p_j) = -x_i^\top A_{ij} x_j + \text{const}. \tag{6.7}$$

Indeed, matrices $A_{ij} = [a^{ij}_{uv}]_{1 \le u,v \le K-1} \in \mathbb{R}^{(K-1)\times(K-1)}$ defined by

$$a^{ij}_{uv} = -\ell_{ij}(P_{i,u+1}, P_{j,v+1}) + \ell_{ij}(P_{i,u}, P_{j,v+1}) + \ell_{ij}(P_{i,u+1}, P_{j,v}) - \ell_{ij}(P_{i,u}, P_{j,v}) \tag{6.8}$$

satisfy (6.7). In a similar way, we can construct $b_i \in \mathbb{R}^{K-1}$ such that $\ell_i(p_i) = -b_i^\top x_i + \text{const}$. Accordingly, the objective function ℓ of problem (6.3) satisfies $\ell(p) = -(x^\top A x + b^\top x) + \text{const}$, where we define $A = [A_{ij}]_{1 \le i,j \le M} \in \mathbb{R}^{(K-1)M \times (K-1)M}$ and $b = [b_i]_{1 \le i \le M} \in \mathbb{R}^{(K-1)M}$. The conditions $x_i \in \{c_1, \ldots, c_K\} \quad (i = 1, \ldots, M)$ can be expressed as $x_u \le x_v \quad ((u,v) \in C)$, where we define $C := \{((K-1)(i-1)+k+1, (K-1)(i-1)+k) \mid 1 \le i \le M, 1 \le k \le K-2\}$. Consequently, problem (6.3) is reduced to problem (6.4). Although [88] also gives another BQP formulation for the problem (6.3) and relaxes it to a semi-definite programming problem, our construction of the BQP problem can be solved much more efficiently, as is explained in the next section.

6.4.2 Minimum Cut for Problems with Substitute Goods Property

As is easily seen from (6.8), if the problem satisfies the substitute-goods property, matrix A has only non-positive entries. It is well known that unconstrained binary quadratic programming problems such as (6.4) with non-positive $A \in \mathbb{R}^{n \times n}$ and $C = \emptyset$ can be efficiently solved[3] by algorithms for minimum cut [46]. Indeed, we can construct a positive weighted directed graph, $G = (V = \{s, t, 1, 2, \ldots, n\}, E \subseteq V \times V, w : E \to \mathbb{R}_{>0} \cup \{\infty\})$[4] for which

$$x^\top A x + b^\top x = c_G(\{s\} \cup \{u \mid x_u = 1\}) + \text{const} \tag{6.9}$$

holds for all $x \in \{0,1\}^n$, where c_G is the cut function of graph G[5]. Hence, once we can compute a minimum s-t cut U that is a vertex set $U \subseteq V$ minimizing $c_G(U)$ subject to $s \in U$ and $t \notin U$, we can construct an optimal solution $x = [x_1, \ldots, x_n]^\top$ to the problem (6.4) by setting

$$x_u = \begin{cases} 1 & (u \in U) \\ 0 & (u \notin U) \end{cases} \quad (u = 1, \ldots, n). \tag{6.10}$$

For constrained problems such as (6.4) with $C \neq \emptyset$, the constraint $x_u \le x_v$ is equivalent to $x_u = 1 \implies x_v = 1$. This condition can be, in the minimum cut problem, expressed as $u \in U \implies v \in U$. By adding a directed edge (u, v) with weight ∞, we can forbid the minimum cut to violate the constraints. In fact, if both $u \in U$ and $v \notin U$ hold, the value of the cut function is ∞, and hence such a U cannot be a minimum cut. We summarize this in Algorithm 9.

[3]The computational cost of the minimum cut depends on the choice of algorithms. For example, if we use Dinic's method, the time complexity is $O(n^3 \log n) = O((KM)^3 \log(KM))$.

[4]s, t are auxiliary vertices different from $1, \ldots, n$ corresponding to source, sink in maximum flow problems.

[5]For details about the construction of G, see, e.g., [27, 111].

Algorithm 9 s-t cut for price optimization with the substitute-goods property

```
Require: Problem instance $(A, b, C)$ of (6.4), where all entries of $A$ are non-positive.
Ensure: An optimal solution $x^*$ to (6.4).
1: Construct a weighted directed graph $G = (V, E, w)$ satisfying (6.9).
2: Add edges $C$ with weight $\infty$ to $G$, i.e., set $E \leftarrow E \cup C$ and $w(u, v) \leftarrow \infty$ for all $(u, v) \in C$.
3: Compute a minimum $s$-$t$ cut $U^*$ of $G$, define $x^*$ by (6.10) and return $x^*$.
```

6.4.3 Submodular Relaxation for General Problems

For problems without the substitute-goods property, we first decompose the matrix A into A^+ and A^- so that $A^+ + A^- = A$, where $A^+ = [a^+_{uv}]$ and $A^- = [a^-_{uv}] \in \mathbb{R}^{n \times n}$ are given by

$$a^+_{uv} = \begin{cases} a_{uv} & (a_{uv} \geq 0) \\ 0 & (a_{uv} < 0) \end{cases}, \quad a^-_{uv} = \begin{cases} 0 & (a_{uv} \geq 0) \\ a_{uv} & (a_{uv} < 0) \end{cases} \quad (u, v \in N). \tag{6.11}$$

This leads to a decomposition of the objective function of Problem (6.4) into supermodular and submodular terms:

$$x^\top A x + b^\top x = x^\top A^+ x + x^\top A^- x + b^\top x, \tag{6.12}$$

where $x^\top A^+ x$ is supermodular and $x^\top A^- x + b^\top x$ is submodular. Our approach is to replace the supermodular term $x^\top A^+ x$ by a linear function to construct a submodular function approximating $x^\top A x + b^\top x$, that can be minimized by Algorithm 9. Similar approaches can be found in the literature, e.g. [69, 158], but ours has a significant point of difference; our method constructs approximate functions bounding objectives *from below*, which provides information about the degree of accuracy.

Consider an affine function $h(x)$ such that $h(x) \leq x^\top A^+ x$ for all $x \in \{0, 1\}^n$. Such an h can be constructed as follows. Since

$$\gamma_{uv}(x_u + x_v - 1) \leq x_u x_v \quad (x_u, x_v \in \{0, 1\}) \tag{6.13}$$

holds for all $\gamma_{uv} \in [0, 1]$, an arbitrary matrix $\Gamma \in [0, 1]^{n \times n}$ satisfies

$$x^\top A^+ x \geq x^\top (A^+ \circ \Gamma) 1 + 1^\top (A^+ \circ \Gamma) x - 1^\top (A^+ \circ \Gamma) 1 =: h_\Gamma(x), \tag{6.14}$$

where $A^+ \circ \Gamma$ denotes the Hadamard product, i.e., $(A^+ \circ \Gamma)_{uv} = a^+_{uv} \cdot \gamma_{uv}$. From inequality (6.14), the optimal value of the following problem,

$$\begin{array}{ll} \text{Minimize} & x^\top A^- x + b^\top x + h_\Gamma(x) \\ \text{subject to} & x = [x_1, \ldots, x_n]^\top \in \{0, 1\}^n, \\ & x_u \leq x_v \quad ((u, v) \in C), \end{array} \tag{6.15}$$

is a lower bound for that of problem (6.4). Since A^- has non-positive entries and $b^\top x + h_\Gamma(x)$ is affine, we can solve (6.15) using Algorithm 9 to obtain an approximate solution for (6.4) and a lower bound for the optimal value of (6.4).

Proximal gradient method with sequential submodular relaxation An essential problem in submodular relaxation is how to choose $\Gamma \in [0,1]^{n\times n}$ and to optimize x given Γ. Let $\psi(\Gamma)$ denote the optimal value of (6.15), i.e., define $\psi(\Gamma)$ by $\psi(\Gamma) = \min_{x\in R} x^\top A^- x + b^\top x + h_\Gamma(x)$, where R is the feasible region of (6.15). Then, for simultaneous optimization of x and Γ, we consider the following problem:

$$\text{Maximize } \psi(\Gamma) \quad \text{subject to } \Gamma \in [0,1]^{n\times n}, \tag{6.16}$$

which can be rewritten as follows:[6]

$$\text{Minimize} \quad -\psi(\Gamma) + \Omega(\Gamma) \quad \text{subject to } \Gamma \in \mathbb{R}^{n\times n}, \tag{6.17}$$

where we define $\Omega : \mathbb{R}^{n\times n} \to \mathbb{R} \cup \{\infty\}$ by

$$\Omega(\Gamma) = \begin{cases} 0 & (\Gamma \in [0,1]^{n\times n}) \\ \infty & (\Gamma \notin [0,1]^{n\times n}) \end{cases}. \tag{6.18}$$

Then, $-\psi(\Gamma)$ is convex and (6.17) can be solved using a proximal gradient method.

Let $\Gamma_t \in \mathbb{R}^{n\times n}$ denote the solution on the t-th step. Let x_t be the optimal solution of (6.15) with $\Gamma = \Gamma_t$, i.e.,

$$x_t \in \arg\min_{x\in R}\{x^\top A^- x + b^\top x + h_{\Gamma_t}(x)\}. \tag{6.19}$$

The partial derivative of $-h_\Gamma(x)$ w.r.t. Γ at (Γ_t, x_t), denoted by S_t, is then a subgradient of $-\psi(\Gamma)$ at Γ_t, which can be computed as follows:

$$S_t = A^+ \circ (11^\top - x_t 1^\top - 1 x_t^\top) \tag{6.20}$$

By using S_t and a decreasing sequence $\{\eta_t\}$ of positive real numbers, we can express the update scheme for the proximal gradient method as follows:

$$\Gamma_{t+1} \in \arg\min_{\Gamma\in\mathbb{R}^{n\times n}}\{S_t \cdot \Gamma + \frac{1}{2\eta_t}\|\Gamma - \Gamma_t\|^2 + \Omega(\Gamma)\}, \tag{6.21}$$

We can compute Γ_{t+1} satisfying (6.21) by

$$\Gamma_{t+1} = \mathrm{Proj}_{[0,1]^{n\times n}}(\Gamma_t - \eta_t S_t), \tag{6.22}$$

where $\mathrm{Proj}_{[0,1]}(X)$ is defined by

$$(\mathrm{Proj}_{[0,1]}(X))_{uv} = \begin{cases} 0 & ((X)_{uv} < 0) \\ 1 & ((X)_{uv} > 1) \\ (X)_{uv} & (\text{otherwise}) \end{cases}. \tag{6.23}$$

The proposed algorithm can be summarized as Algorithm 10.

The choice of $\{\eta_t\}$ has a major impact on the rate of the convergence of the algorithm. From a convergence analysis of the proximal gradient method, when we set $\eta_t = \Theta(1/\sqrt{t})$, it is guaranteed that ψ_t converge to the optimal value ψ_* of (6.16) and $|\psi_t - \psi_*| = O(1/\sqrt{t})$. Because $\psi(\Gamma)$ is non-smooth and not strongly concave, there is no better guarantee of convergence rate, to the best of our knowledge. In practice, however, we can observe the convergence in ~ 10 steps iteration.

[6]Problem (6.16) can be also solved using the ellipsoid method, which guarantees polynomial time-complexity in the input size. However, it is known that the order of its polynomial is large and that the performance of the algorithm can be poor in practice, especially for large size problems. To try to achieve more practical performance, this chapter proposes a proximal gradient algorithm.

Algorithm 10 An iterative relaxation algorithm for (6.4)

```
Require: Problem instance $(A, b, C)$ of (6.4).
Ensure: An approximate solution $\hat{x}$ to (6.4) satisfying (6.25), a lower bound $\psi$ of optimal value
    of (6.4).
1: Set $\Gamma_1 = 11^\top/2$, $t = 1$, min_value $= \infty$, $\psi = -\infty$.
2: while Not converged do
3:   Compute $x_t$ satisfying (6.19) by using Algorithm 9, and compute
       $$\text{value}_t = x_t^\top A x_t + b^\top x_t, \quad \psi_t = x_t^\top A^- x_t + b^\top x_t + h_{\Gamma_t}(x_t), \quad \psi = \max\{\psi, \psi_t\}$$
4:   if value$_t$ < max_value then
5:     Update value and $\hat{x}$ by
         $$\text{min\_value} = \text{value}_t, \quad \hat{x} = x_t. \tag{6.24}$$
6:   end if
7:   Compute $\Gamma_{t+1}$ by (6.22) and (6.23).
8: end while
9: Return $\hat{x}$, min_value and $\psi$.
```

Initialization of Γ Let $\tilde{x}_\Gamma$ denote an optimal solution to (6.15). We employ $\Gamma_1 = {}^1\!/_2 11^\top$ for the initialization of Γ because $(x_u + x_u - 1)/2$ is the tightest lower bound of $x_u x_v$ in the max-norm sense, i.e., $h(x_u, x_v) = (x_u + x_v - 1)/2$ is the unique minimizer of $\max_{x_u, x_v \in \{0,1\}}\{|x_u x_v - h(x_u, x_v)|\}$, subject to the constraints that $h(x_u, x_v)$ is affine and bounded from above by $x_u x_v$. In this case, $\tilde{x}_\Gamma$ is an approximate solution satisfying the following performance guarantee.

Proposition 6.4.1. *If $\Gamma = 11^\top/2$, then $\tilde{x}_\Gamma$ satisfies*

$$\tilde{x}_\Gamma^\top A \tilde{x}_\Gamma + b^\top \tilde{x}_\Gamma \le x_*^\top A x_* + b^\top x_* + \frac{1}{2} 1^\top A^+ 1, \tag{6.25}$$

where x_* is an optimal solution to (6.4).

Proof. From (6.14), we have

$$x_*^\top A x_* + b^\top x_* \ge x_*^\top A^- x_* + b^\top x_* + h_\Gamma(x_*). \tag{6.26}$$

Further, it holds that

$$x_*^\top A^- x_* + b^\top x_* + h_\Gamma(x_*) \ge \tilde{x}_\Gamma^\top A^- \tilde{x}_\Gamma + b^\top \tilde{x}_\Gamma + h_\Gamma(\tilde{x}_\Gamma) \tag{6.27}$$

since $\tilde{x}_\Gamma$ is an optimal solution to (6.15). From the definition (6.14) of h_Γ, $h_\Gamma(\tilde{x}_\Gamma)$ with $\Gamma = 11^\top/2$ satisfies

$$\begin{aligned} h_\Gamma(\tilde{x}_\Gamma) &= \tilde{x}^\top (A^+ \circ \Gamma) 1 + 1^\top (A^+ \circ \Gamma) \tilde{x} - 1^\top (A^+ \circ \Gamma) 1 \\ &= \frac{1}{2} \tilde{x}^\top A^+ 1 + \frac{1}{2} 1^\top A^+ \tilde{x} - \frac{1}{2} 1^\top A^+ 1 \\ &= \tilde{x}_\Gamma^\top A^+ \tilde{x}_\Gamma - \frac{1}{2} \tilde{x}_\Gamma^\top A^+ \tilde{x}_\Gamma - \frac{1}{2} (1 - \tilde{x}_\Gamma)^\top A^+ (1 - \tilde{x}_\Gamma) \\ &\ge \tilde{x}_\Gamma^\top A^+ \tilde{x}_\Gamma - \frac{1}{2} 1^\top A^+ 1. \end{aligned} \tag{6.28}$$

Table 6.1: Ranges of parameters in regression models. (i) is supermodular, (ii) is supermodular + submodular, and (iii) is submodular.

	β_{ij} $(i \neq j)$	β_{ii}	α_i
(i)	$[0, 2]$	$[-2M, -M]$	$[M, 3M]$
(ii)	$[-25, 25]$	$[-2M, 0]$	$[M, 3M]$
(iii)	$[-2, 0]$	$[M-3, M-1]$	$[1, 3]$

Table 6.2: Results on real retail data. (a) is computational time, (b) is estimated gross profit, (c) is upper bound.

	actual	proposed	QPBO
(a)	-	36[s]	964[s]
(b)	1403700	1883252	1245568
(c)	-	1897393	1894555

From (6.26), (6.27) and (6.28), we obtain

$$\begin{aligned} x_*^\top A^- x_* + b^\top x_* + h_\Gamma(x_*) &\geq \tilde{x}_\Gamma^\top A^- \tilde{x}_\Gamma + b^\top \tilde{x}_\Gamma + \tilde{x}_\Gamma^\top A^+ \tilde{x}_\Gamma - \frac{1}{2} 1^\top A^+ 1. \\ &= \tilde{x}_\Gamma^\top A \tilde{x}_\Gamma + b^\top \tilde{x}_\Gamma - \frac{1}{2} 1^\top A^+ 1. \end{aligned}$$

□

6.5 Experiment

6.5.1 Simulations

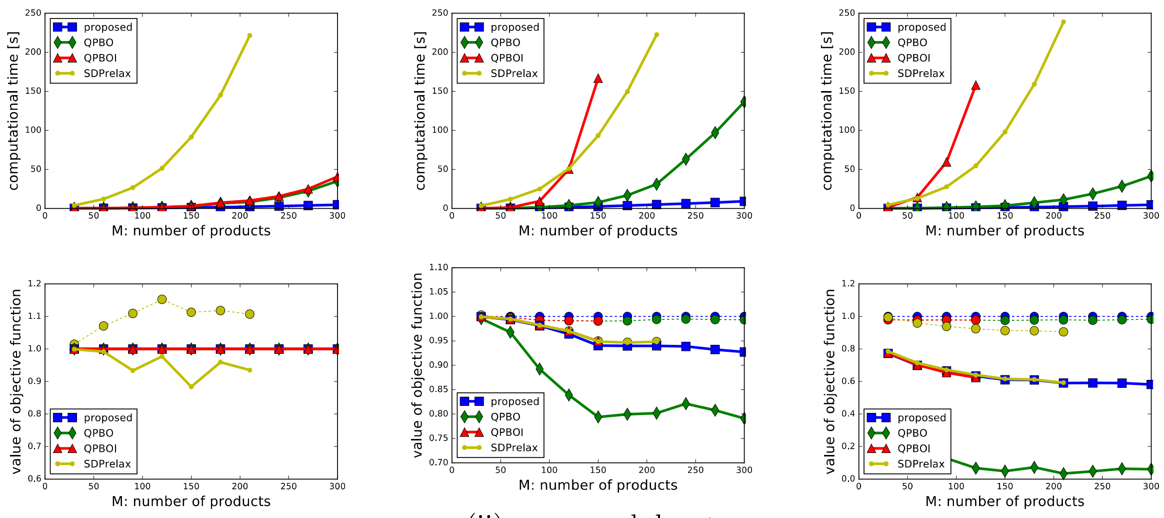
This section investigates behavior of Algorithm 10 on the basis of the simulation model used in [88], and we compare the proposed method with state-of-the-art methods: the SDP relaxation method [88] and the QPBO and QBPOI methods [110]. We use SDPA 7.3.8 to solve SDP problems[7] and use the implementation of QPBO and QPBOI written by Kolmogolov.[8] QPBO methods computes partial labeling, i.e., there might remain unlabeled variables, and we set unlabeled variables to 0 in our experiments. For computing a minimum *s*-*t* cut, we use Dinic's algorithm [46]. All experiments were conducted in a machine equipped with Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz, 768GB RAM. We limited all processes to a single CPU core.

Revenue simulation model [88] The sales quantity q_i of the i-th product was generated from the regression model $q_i = \alpha_i + \sum_{j=1}^{M} \beta_{ij} p_j$, where $\{\alpha_i\}$ and $\{\beta_{ij}\}$ were generated by uniform distributions. We considered three types of uniform distributions to investigate the effect of submodularity, as shown in Table 6.1, which correspond to three different situations: (i) all pairs of products are substitute goods, i.e., the gross profit function is supermodular, (ii) half pairs are substitute goods and the others are complementary goods, i.e., the gross profit function contains submodular terms and supermodular terms, and (iii) all pairs are complementary goods, i.e., the gross profit function is submodular. Price candidates $\mathcal{P}_i$ and cost c_i for each product are fixed to $\mathcal{P}_i = \{0.6, 0.7, \ldots, 1.0\}$ and $c_i = 0$, respectively.

Scalability and accuracy comparison We evaluated four methods in terms of computational time (sec) and optimization accuracy (i.e. optimal values calculated by four methods). In addition to calculating approximate optimal solutions and values, all four algorithms derive

[7]http://sdpa.sourceforge.net/

[8]http://pub.ist.ac.at/~vnk/software.html



(i) supermodular (ii) supermodular + submodular (iii) submodular

Figure 6.1: Comparisons of proposed, QPBO, QPBOI, and SDPrelax methods on revenue simulation data. The horizontal axis represents the number M of products. The vertical axes represent computational time (top) and optimal values of four methods (6.3) (bottom). For the bottom, circle markers with dashed line represent the computed upper bounds of the optimal values, and optimal values and upper bounds are normalized so that upper bounds with the proposed method are equal to 1.

upper bounds of *exact* optimal value, which provide information about how accurate the calculated solution.[9] Fig. 6.1 shows the results with $M = 30, 60, \dots, 300$ for situations (i),(ii) and (iii). The plotted values are arithmetic means of 5 random problem instances. We can observe that proposed, QPBO and QPBOI methods derived exact solutions in the case (i), which can be confirmed from the computed upper bounds coinciding with the values of objective function. For situations (ii) and (iii), on the other hand, the upper bound and the objective value did not coincide and the solutions by QPBO were worse than the others. The solutions by QPBOI and SDPrelax are as good as the proposed methods, but their computational costs are significantly higher especially for the situations (ii) and (iii). For all situations, the proposed method successfully derived solutions as good as the best of the four methods did, and its computational cost was the lowest.

6.5.2 Real-World Retail Data

Data and settings We applied the proposed method to actual retail data from a middle-size supermarket located in Tokyo [164].[10] We selected 50 regularly-sold beer products. The data range is approximately three years from 2012/01 to 2014/12, and we used the first 35 months (1065 samples) for training regression models and simulated the best price strategy for the next 20 days. Therefore, the problem here was to determine 1000 prices (50 products $\times$ 20 days).

[9]For example, the coincidence of the upper bound and the calculated optimal value implies that the algorithm computed the exact optimal solution.

[10]The Data has been provided by KSP-SP Co., LTD, http://www.ksp-sp.com.

For forecasting the sales quantity $q_i^{(d)}$ of the i-product on the d-th day, we use prices features $\{p_j^{(d')}\}_{1\leq j\leq 50, d-19\leq d'\leq d}$ of 50 products for the 20 days before the d-th day. In addition to these 1000 linear price features, we employed "day of the week" and "month" features (both binary), as well as temperature forecasting features (continuous), as external features. The price candidates $\{P_{ik}^{(d)}\}_{k=1}^{5}$ were generated by splitting equally the range $[P_{i1}, P_{i5}]$, where P_{i1} and P_{i5} are the highest and lowest prices of the i-th product in the historical data. We assumed that the cost $c_i^{(d)}$ was $0.3P_{i5}$ (30% of the list prices). Our objective was to obtain a price strategy for 50-products over the 20 days, from the 1066-th to 1085-th, which involves 1000-dimensional variables, in order to maximize the sum of the gross profit for the 20 days. We estimated parameters in regression models, using the ridge regression method. The estimated model contained 310293 pairs with the substitute-goods property and 189207 pairs with complementary goods property.

The results are summarized in Table 6.2, where "actual" means the gross profit computed on the basis of the historical data regarding sales quantities and prices over the 20 days, from the 1066-th to 1085-th, and costs $c_i^{(d)} = 0.3P_{i5}$. Thus, the target is to find a strategy that expectedly achieves better gross profit than "actual". We have omitted results for QPBOI and SDPrelax here because they did not terminate after running over 8 hours. We observe that the proposed method successfully derived a price strategy over 1000 products, which can be expected to increase gross profit significantly in spite of its cheap computational cost, in contrast to QPBO, which failed with more expensive computation. Although Table 6.2 shows results using a single CPU core for fair comparison, the algorithm can be easily parallelized that can finish optimization in a few seconds. This makes it possible to dynamically change prices in real time or enables price managers to flexibly explore a better price strategy (changing a price range, target products, domain constraints, etc.)

6.6 Conclusion

In this chapter we dealt with price optimization based on large-scale demand forecasting models. We have shown that the gross profit function is supermodular under the assumption of the substitute-goods property. On the basis of this supermodularity, we have proposed an efficient algorithm that employs network flow algorithms and that returns exact solutions for problems with the substitute-goods property. Even in case in which the property does not hold, the proposed algorithm can efficiently find approximate solutions. Our empirical results have shown that the proposed algorithm can handle hundreds/thousands products with much cheaper computational cost than other existing methods.

Chapter 7

Optimal Algorithm for Online Convex Optimization with Bandit Feedback

We consider non-stochastic bandit convex optimization with strongly-convex and smooth loss functions. For this problem, Hazan and Levy have proposed an algorithm with a regret bound of $\tilde{O}(d^{3/2}\sqrt{T})$ given access to an $O(d)$-self-concordant barrier over the feasible region, where d and T stand for the dimensionality of the feasible region and the number of rounds, respectively. However, there are no known efficient ways for constructing self-concordant barriers for general convex sets, and a $\tilde{O}(\sqrt{d})$ gap has remained between the upper and lower bounds, as the known regret lower bound is $\Omega(d\sqrt{T})$. Our study resolves these two issues by introducing an algorithm that achieves an optimal regret bound of $\tilde{O}(d\sqrt{T})$ under a mild assumption, without self-concordant barriers. More precisely, the algorithm requires only a membership oracle for the feasible region, and it achieves an optimal regret bound of $\tilde{O}(d\sqrt{T})$ under the assumption that the optimal solution is an interior of the feasible region. Even without this assumption, our algorithm achieves $\tilde{O}(d^{3/2}\sqrt{T})$-regret.

7.1 Introduction

Bandit convex optimization (BCO) is a framework for online decision-making with limited feedback. In this framework, a player is given a convex *feasible region* $\mathcal{K}$ and the number T of *rounds.* In each round $t = 1, 2, \ldots, T$, the player chooses an *action* $a_t \in \mathcal{K}$, and the environment independently chooses convex *loss function* $f_t : \mathcal{K} \to [-1, 1]$. Not all information about the loss function is revealed to the player then, but only the *bandit feedback* is available, i.e., the player can observe $f_t(x_t)$ alone. The goal of the player is to minimize cumulative loss $\sum_{t=1}^{T} f_t(x_t)$, and performance is evaluated in terms of the *regret* $R_T(x^*)$ defined by

$$R_T(x^*) = \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \tag{7.1}$$

for $x^* \in \mathcal{K}$.

This chapter focuses on a *non-stochastic* or *adversarial* setting. In this setting, we do not assume any generative model for the loss functions f_t, and f_t can change arbitrarily. An alter-

Table 7.1: Regret bound for bandit convex optimization with strongly-convex and smooth objective functions.

Reference	Regret bound	Notes
Flaxman et al. [58],	$\tilde{O}(d^{2/3}T^{2/3})$	No additional assumptions.
Agarwal et al. [5]	$\tilde{O}(d\sqrt{T})$	Assume that the optimization is unconstrained, i.e., $\mathcal{K} = \mathbb{R}^d$.
Hazan and Levy [80]	$\tilde{O}(d\sqrt{\nu T})$	Require a ν-self-concordant barrier for $\mathcal{K}$. Parameter ν is at least d.
Corollary 7.5.1 [This work]	$\tilde{O}(d\sqrt{T})$	Assume that the optimal solution is an interior of $\mathcal{K}$.
Corollary 7.5.2 [This work]	$\tilde{O}(d^{3/2}\sqrt{T})$	No additional assumptions.
Shamir [153]	$\Omega(d\sqrt{T})$	A lower bound that implies $O(d\sqrt{T})$-bounds are minimax optimal.

native setting, a *stochastic* setting in which f_t independently follows an unknown probabilistic distribution, can be regarded as a special case of the non-stochastic setting. Indeed, algorithms for the non-stochastic setting work even for this stochastic setting, and regret upper bounds for the former setting apply even to the latter.

Work on non-stochastic BCO was initiated by Flaxman et al. [58] and Kleinberg [108], in which algorithms with regret bounds of $O(T^{3/4})$ were proposed. Note that there has been a lower bound of $\Omega(\sqrt{T})$, i.e., it is known that no algorithm can achieve a better regret bound than $O(\sqrt{T})$. A gap of $O(T^{1/4})$ between the upper and the lower bounds remained for a long time, until Bubeck et al. [33], Bubeck and Eldan [31] and Bubeck et al. [34] proposed algorithms with $\tilde{O}(\sqrt{T})$-regret bounds, where $\tilde{O}(\cdot)$ notation ignores factors of poly-logarithmic terms. There has still been a large gap, however, w.r.t. the dimension d of the feasible region $\mathcal{K}$; the best known upper and lower bounds are of $\tilde{O}(d^{9.5}\sqrt{T})$ and $\Omega(d\sqrt{T})$, respectively. Bubeck et al. [34] have conjectured that the optimal regret bound is of $\tilde{\Theta}(d^{3/2}\sqrt{T})$, but there have been no significant improvements since their study.

This chapter focuses on an important special case of BCO in which the loss functions are strongly-convex and smooth. Work on such special cases is summarized in Table 7.1. Agarwal et al. [5] showed that a modified version of the algorithm by [58] can achieve a regret of $\tilde{O}(d\sqrt{T})$ for unconstrained problems, i.e., for problems with $\mathcal{K} = \mathbb{R}^d$. This result can be said to be minimax optimal because Shamir [153] proved a lower bound of $\Omega(d\sqrt{T})$ that holds even for strongly-convex and smooth losses. On the other hand, for general constrained problems, the minimax optimal rate remains to be determined.

An important sign of progress in strongly-convex and smooth BCO has been shown by Hazan and Levy [80]. They proposed an algorithm that can be applied to constrained problems and has a better regret bound. However, this algorithm does not directly apply to general problems because it requires a *ν-self-concordant barrier*[1] for the feasible region $\mathcal{K}$, where a self-concordant barrier is a convex function with certain properties and $\nu > 0$ is a parameter of it. For some special cases of convex sets, we have explicit forms of self-concordant barriers; for example, if $\mathcal{K}$ can be expressed by m linear inequalities, one has an m-self-concordant barrier for $\mathcal{K}$. For a general convex set, however, there are no known efficient ways for constructing a self-concordant barrier. Further, even if we were to have a ν-self-concordant barrier, the regret bound would be $O(d\sqrt{\nu T})$, which implies that there would still be a $\sqrt{\nu}$ gap between the upper and the lower bounds. Because ν is in general at least d for a compact convex set $\mathcal{K}$ (see, e.g., [137]), there is a gap of $\Omega(\sqrt{d})$ from the lower bound of $O(d\sqrt{T})$, and if we were to have only self-concordant

[1]Self-concordant barriers are special cases of convex functions that were introduced in order to develop interior-point methods for convex optimization. For details on self-concordant barriers, see, e.g., [137].

barriers with a large ν, e.g., if $\mathcal{K}$ were expressed by $m(\gg d)$ linear inequalities, the gap would be even worse.

Our contribution is to overcome the above issues by developing a novel algorithm with the following two strengths. (i) Under a mild assumption, our algorithm achieves $\tilde{O}(d\sqrt{T})$-regret, which is minimax optimal up to logarithmic factors. This represents the first tight bound for bandit convex optimization that applies even to constrained problems. More precisely, under the assumption that the optimal solution is an r-interior,[2] our algorithm enjoys a regret bound of $\tilde{O}(d\sqrt{T} + d^2/r^2)$, as given in Corollary 7.5.1. Also, even without the assumption of interiors, the algorithm has a regret bound of $\tilde{O}(d^{3/2}\sqrt{T})$, which is, at least, not worse than existing algorithms. (ii) Our algorithm does not require self-concordant barriers. Indeed, we only assume that we have access to a membership oracle for the feasible region. This means that our algorithm works well even if $\mathcal{K}$ is expressed by an exponentially large number of linear inequalities, or if we are not given explicit forms of $\mathcal{K}$.

A key ingredient in our algorithm is the *multiplicative weight update* (MWU) method [11], in which we update probabilistic distributions over $\mathcal{K}$ on the basis of estimators of objective functions. To estimate objective functions from bandit feedback, we use techniques of *smoothing* and *ellipsoidal sampling* [58]. Our analyses for regret bounds rely on theories for *log-concave distributions* [127], which is a class of continuous distributions that includes normal distributions, exponential distributions, and distributions in our algorithm. Further, this algorithm can be implemented so that it runs in polynomial time, thanks to efficient algorithms for sampling from log-concave distributions [127, 135].

7.2 Related Work

For *bandit linear optimization*, an important special case of BCO in which objectives are linear functions, there have been many signs of progress. Bubeck et al. [32] and Cesa-Bianchi and Lugosi [38] provided algorithms that achieve regret of $\tilde{O}(d\sqrt{T})$. These algorithms can be applied to *combinatorial bandits*, bandit linear optimization problems in which the feasible region $\mathcal{K}$ is a discrete finite set. The regret bounds of $\tilde{O}(d\sqrt{T})$ can be said to be non-improvable because Dani et al. [52] showed a regret lower bound of $\Omega(d\sqrt{T})$. However, computationally efficient implementation achieving $\tilde{O}(d\sqrt{T})$-regret for a general $\mathcal{K}$, including discrete sets, remains a possibility. Hazan and Karnin [79] have proposed a computationally efficient algorithm that achieves $\tilde{O}(d\sqrt{T})$-regret if $\mathcal{K}$ is a convex set.

Online convex optimization [152, 77, 37] is a variant of BCO in which a player can get feedback on complete information about objective functions, rather than bandit feedback. For general convex objectives, it has been known that online gradient descent methods [169] can achieve $O(\sqrt{dT})$ regret, and this bound is minimax optimal. For a special case called *exp-concave functions*, which involves a milder assumption than strong-convexity, Hazan et al. [81] provided efficient algorithms with a regret bound of $O(d \log T)$, which is minimax optimal as well because there is a lower bound of $\Omega(d \log T)$ [139].

It has been shown that one can achieve better regret for BCO if *multi-point feedback* is available, i.e., if the player can observe the values of objective functions on $k \geq 2$ different points in each round [5, 138, 54]. For general convex functions, it is known that one can achieve

[2]The definition of r-interior is given in Section 7.3 and Figure 7.1.

$O(d^2\sqrt{T})$ regret in a two-point feedback setting. Further, Agarwal et al. [5] have shown that, under the assumption of strong-convexity and two-point feedback, one can achieve $O(d^2 \log T)$ regret.

After the study by Hazan and Levy [80], many works regarding BCO have followed. Hu et al. [84] have considered a more general problem setting in which *biased noisy gradient* is available. Mohri and Yang [132] provided a BCO algorithm that does not require a priori assumptions of strong convexity or smoothness. These two studies capture more general scenarios than ours, but their regret bounds achieved in our setting are not superior to those by Hazan and Levy [80]. Kumagai [115] considered a *dueling bandit* problem with strongly-convex and smooth costs, in which an algorithm based on self-concordant functions was proposed. Chen et al. [41] focused on computationally efficient methods for BCO, and provided a projection-free algorithm that achieves sublinear regret for general convex losses.

Our proposed algorithm is based on the multiplicative weight update (MWU) method [11]. Algorithms similar to MWU can be found in the literature in the early 1950s in the context of game theory [29, 28, 144], and MWU has been independently rediscovered in other fields including computational geometry, and machine learning. Our approach is to use continuous MWU, multiplicative weight update over continuous domains, which has been applied to various online optimization problems, including Cover's universal portfolios [47], bandit linear optimization [79], and online improper learning [82].

7.3 Problem Setting and Assumption

A player is given a convex *feasible region* $\mathcal{K} \subseteq \mathbb{R}^d$ and a number T of rounds of decision making, where d is a positive integer standing for the dimensionality of the feasible region. For each $t = 1, 2, \ldots, T$, the player chooses an *action* $a_t \in \mathcal{K}$, and an environment chooses a convex function $f_t : \mathcal{K} \to [-1, 1]$ at the same time. The player observes feedback of $f_t(a_t)$ before choosing the next action a_{t+1}. We assume that $\mathcal{K}$ has a positive volume, i.e., $\int_{x\in\mathcal{K}} 1 \mathrm{d}x > 0$. We assume that f_t is σ-strongly convex and β-smooth, i.e., that the following hold for all $x, y \in \mathcal{K}$:

$$f_t(y) \geq f_t(x) + \nabla f_t(x)^\top (y - x) + \frac{\sigma}{2}\|y - x\|_2^2, \tag{7.2}$$

$$f_t(y) \leq f_t(x) + \nabla f_t(x)^\top (y - x) + \frac{\beta}{2}\|y - x\|_2^2, \tag{7.3}$$

where $\nabla f_t(x) \in \mathbb{R}^d$ stands for the gradient of f_t at x.

The performance of the player is evaluated in terms of the *regret* $R_T(x^*)$, which is defined as

$$R_T(x^*) = \sum_{t=1}^{T} f_t(a_t) - \sum_{t=1}^{T} f_t(x^*). \tag{7.4}$$

In this chapter, we suppose that a player arbitrarily chooses a convex *benchmark set* $\mathcal{K}' \subseteq \mathcal{K}$. We consider regret $R_T(x^*)$ for $x^* \in \mathcal{K}'$, i.e., we care about the value of $\sup_{x^*\in\mathcal{K}'} \mathbf{E}[R_T(x^*)]$, the expected gap between the cumulative losses for the algorithm's outputs and for the optimal single action x^* belonging to $\mathcal{K}'$. The value $\sup_{x^*\in\mathcal{K}'} \mathbf{E}[R_T(x^*)]$ is equal to the standard worst-case regret $\sup_{x^*\in\mathcal{K}} \mathbf{E}[R_T(x^*)]$, if the optimal single action $x^* \in \arg\min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$ belongs to $\mathcal{K}'$.

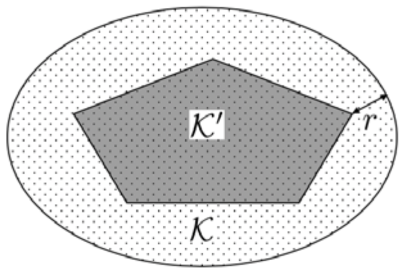


Figure 7.1: Geometric interpretation of $\mathcal{K}'$ that consists of r-interiors of $\mathcal{K}$.

A point $x \in \mathbb{R}^d$ is called a *γ-interior* of $\mathcal{K}$ if $\|y - x\|_2 \leq \gamma$ implies $y \in \mathcal{K}$. For example, if $\mathcal{K}$ is expressed by m linear inequalities, i.e., $\mathcal{K}$ can be expressed as $\mathcal{K} = \{x \in \mathbb{R}^d \mid a_j^\top x \leq b_j \quad (j \in [m])\}$ with $a_j \in \mathbb{R}^d, b_j \in \mathbb{R}$ such that $\|a_j\|_2 = 1$, then the convex set $\mathcal{K}'$ defined by $\mathcal{K}' = \{x \in \mathbb{R}^d \mid a_j^\top x \leq b_j - r \quad (j \in [m])\}$ consists of r-interiors of $\mathcal{K}$. For general benchmark set $\mathcal{K}' \subseteq \mathcal{K}$, let $r \geq 0$ be a non-negative real value for which all members of $\mathcal{K}'$ are r-interiors. Figure 7.1 shows a geometric interpretation of how r is determined for $\mathcal{K}$ and $\mathcal{K}' \subseteq \mathcal{K}$. For the special case in which $\mathcal{K}' = \mathcal{K}$, r is equal to zero.

We assume that we have access to a *membership oracle* for $\mathcal{K}'$. This means that, given $x \in \mathbb{R}^d$, we can determine if $x \in \mathcal{K}'$ or not, by calling on the membership oracle. If $\mathcal{K}'$ is expressed by m inequalities ($\mathcal{K}' = \{x \in \mathbb{R}^d \mid g_j(x) \leq 0 \quad (j \in [m])\}$), then we have access to the membership oracle for $\mathcal{K}'$ because one can check if $x \in \mathcal{K}$ by evaluating $g_i(x)$ for $i \in [m]$. Further, it is known that we will have a polynomial-time membership oracle for $\mathcal{K}'$ if we can solve linear optimization problems over $\mathcal{K}'$ in polynomial time [150].

7.4 Algorithm

7.4.1 Preliminary

Notation For a vector $x = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$, let $\|x\|_2$ denote the ℓ_2 norm of x, i.e., $\|x\|_2 = \sqrt{x^\top x} = \sqrt{\sum_{i=1}^d x_i^2}$. For a matrix $X \in \mathbb{R}^{d \times d}$, $\|X\|_2$ denotes the ℓ_2-operator norm, i.e., $\|X\|_2 = \max\{\|Xy\|_2 \mid y \in \mathbb{R}^d, \|y\|_2 = 1\}$. If X is a symmetric matrix, $\|X\|_2$ is equal to the maximum absolute value of eigenvalues of X. Given a positive semidefinite matrix $A \in \mathbb{R}^{d \times d}$ and a vector $x \in \mathbb{R}^d$, define $\|x\|_A$ by $\|x\|_A = \sqrt{x^\top A x} = \|A^{1/2} x\|_2$. Similarly, for a matrix $X \in \mathbb{R}^d$, let $\|X\|_A = \|A^{1/2} X A^{1/2}\|_2$. Given two symmetric matrices $A, B \in \mathbb{R}^{d \times d}$, denote $A \bullet B = \mathrm{tr}(AB)$.

Smoothed convex function Let v and u be random variables that follow uniform distributions over $\mathbb{B}^d = \{v \in \mathbb{R}^d \mid \|v\|_2 \leq 1\}$ and $\mathbb{S}^d = \{u \in \mathbb{R}^d \mid \|u\|_2 = 1\}$, respectively. For a convex function f over $\mathbb{R}^d$ and a positive-definite matrix $B \in \mathbb{R}^{d \times d}$, define the *smoothed function* $\hat{f}$ by

$$\hat{f}_B(x) = \mathbf{E}[f(x + Bv)]. \tag{7.5}$$

Then we have the following:

Lemma 7.4.1 ([80]). *The gradient of $\hat{f}_B$ can be expressed as*

$$\nabla \hat{f}_B(x) = \mathbf{E}[d \cdot f(x + Bu) B^{-1} u]. \tag{7.6}$$

If f is β-smooth, it holds that

$$0 \leq \hat{f}_B(x) - f(x) \leq \frac{\beta}{2}\|B^\top B\|_2 = \frac{\beta}{2}\lambda_1(B^\top B). \tag{7.7}$$

If f is σ-strongly convex then so is $\hat{f}_B$.

Equation (7.6) can be shown using Stokes' theorem, and (7.7) follows from the definition of β-smoothness. In the bandit feedback setting, though unbiased estimators of the gradient of f_t are unavailable, those for the smoothed ones $\hat{f}_t$ can be constructed through (7.6). Differences between f_t and $\hat{f}_t$ will be bounded by means of (7.7).

Log-concave distribution A probability distribution over a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ is called a *log-concave distribution* if its probability density function $p : \mathcal{K} \to \mathbb{R}$ can be expressed as $p(x) = \exp(-g(x))$ with a convex function $g : \mathcal{K} \to \mathbb{R}$, i.e., the logarithmic of $p(x)$ is a concave function. Our algorithm maintains log-concave distributions. Random samples from log-concave distributions can be efficiently generated under mild assumptions. Indeed, as shown in [127], there are computationally efficient MCMC algorithms for sampling from p that work given a membership oracle for $\mathcal{K}$ and an evaluation oracle for g. Accordingly, we can efficiently compute estimators of the mean $\mu(p)$ and the covariance matrix $\mathrm{Cov}(p)$ for p. The following lemma is useful for bounding the variance of log-concave distributions:

Lemma 7.4.2 (Prop. 10.1. in [149]). *Suppose that a log-concave distribution over $\mathcal{K}$ has a probability density function $p(x) = \exp(-g(x))$, where g is a σ-strongly convex function. Then, the covariance matrix Σ of p satisfies $\|\Sigma\|_2 \leq 1/\sigma$.*

The following lemma is used to prove a regret bound for our algorithm.

Lemma 7.4.3 (Lemma 5.7 in [127]). *Let X be a random point drawn from a log-concave distribution on $\mathbb{R}$. Assume that $\mathbf{E}[X^2] \leq 1$. Then for every $t > 1$, $\mathrm{Prob}[|X| > t] \leq \exp(-t+1)$.*

We use the following lemma to guarantee that the outputs a_t of our algorithm are included in $\mathcal{K}$.

Lemma 7.4.4 (Lemma 5.5 (a) and Lemma 5.12 in [127]). *Let p be a log-concave distribution over $\mathcal{K}$. The ellipsoid $\{x \in \mathbb{R}^d \mid \|x - \mu(p)\|_{\mathrm{Cov}(p)^{-1}} \leq 1/\mathrm{e}\}$ will then be included in $\mathcal{K}$.*

7.4.2 Continuous Multiplicative Weight Update

In our algorithm, we maintain a function z_t over $\mathcal{K}'$ using the *multiplicative weight update* method [11]. We initialize z_t by $z_1(x) = \sigma\|x\|_2^2/2$. In each round, let p_t be a probability distribution over $\mathcal{K}'$ with density proportional to $\exp(-\eta z_t(x))$, i.e., p_t is defined by

$$Z_t := \int_{x \in \mathcal{K}'} \exp(-\eta z_t(x))\mathrm{d}z, \quad p_t(x) = \frac{\exp(-\eta z_t(x))}{Z_t}. \tag{7.8}$$

Let μ_t and Σ_t denote the mean and the covariance matrix for p_t. We then compute estimators $\hat{\mu}_t$ and $\hat{\Sigma}_t$ for them such that

$$\|\hat{\mu}_t - \mu_t\|_{\Sigma_t^{-1}} \leq 1/9, \quad \|\hat{\Sigma}_t - \Sigma_t\|_{\Sigma_t^{-1}} \leq 1/9, \quad \mathbf{E}[\hat{\mu}_t|\mu_t] = \mu_t. \tag{7.9}$$

Algorithm 11 Continuous multiplicative weight update method for bandit convex optimization

```
Require: Time horizon $T \in \mathbb{N}$, learning rate $\eta > 0$, membership oracle $\mathcal{M}$ for $\mathcal{K}'$, exploration
    parameters $\{\alpha_t\}_{t=1}^T \subseteq \mathbb{R}_{>0}$, strong-convexity parameter $\sigma > 0$.
1: Set $z_1 : \mathcal{K}' \to \mathbb{R}$ by $z_1(x) = \sigma \|x\|_2^2/2$.
2: for $t = 1, 2, \ldots, T$ do
3:     Compute $\hat{\mu}_t$ and $\hat{\Sigma}_t$ for which (7.9) holds.
4:     Compute a positive-definite matrix $B_t \in \mathbb{R}^{d \times d}$ such that $B_t B_t = \hat{\Sigma}_t$.
5:     Pick $u_t$ from $\mathbb{S}^d$ uniformly at random.
6:     Play $a_t = \hat{\mu}_t + \alpha_t B_t u_t$ and observe $f_t(a_t)$.
7:     Set $\hat{g}_t$ by (7.11) and update $z_t$ by (7.13).
8: end for
```

Specific methods for computing such $\hat{\mu}_t$ and $\hat{\Sigma}_t$ will be discussed in the next subsection. Let $B_t \in \mathbb{R}^{d\times d}$ be a positive-definite matrix for which $B_t B_t = \hat{\Sigma}_t$. Such a matrix can be computed, e.g., via eigenvalue decomposition. Consider smoothing f_t as (7.5) with $B = \alpha_t B_t$:

$$\hat{f}_t(x) := \mathbf{E}[f_t(x + \alpha_t B_t v)], \tag{7.10}$$

where α_t is the *exploration parameter* that we will adjust later, and v follows a uniform distribution over $\mathbb{B}^d$. An unbiased estimator of the gradient of $\hat{f}_t$ can then be constructed as follows. Choose u_t from a unit sphere $\mathbb{S}^d = \{u \in \mathbb{R}^d \mid \|u\|_2 = 1\}$, uniformly at random. Play an action of $a_t = \mu_t + \alpha_t B_t u_t$, and then observe $f_t(a_t)$. On the basis of this observation, define $\hat{g}_t \in \mathbb{R}^d$ by

$$\hat{g}_t = d \cdot f_t(a_t)(\alpha_t B_t)^{-1} u_t. \tag{7.11}$$

This is an unbiased estimator of the gradient $\nabla \hat{f}_t(\hat{\mu}_t)$, i.e., given $\hat{\mu}_t$ and B_t, the conditional expectation of $\hat{g}_t$ satisfies

$$\mathbf{E}[\hat{g}_t] = \mathbf{E}[d \cdot f_t(\hat{\mu}_t + \alpha_t B_t u_t)(\alpha_t B_t)^{-1} u_t] = \nabla \hat{f}_t(\hat{\mu}_t), \tag{7.12}$$

where the second inequality follows from (7.6). By means of this unbiased estimator $\hat{g}_t$, we update z_t as

$$z_{t+1}(x) = z_t(x) + \hat{g}_t^\top (x - \hat{\mu}_t) + \frac{\sigma}{2}\|x - \hat{\mu}_t\|_2^2. \tag{7.13}$$

7.4.3 Computation of Approximate Mean $\hat{\mu}_t$ and Covariance $\hat{\Sigma}_t$

Estimators $\hat{\mu}_t$ and $\hat{\Sigma}_t$ satisfying (7.9) can be computed from samples $x_t^{(1)}, \ldots, x_t^{(M)}$ generated by p_t as follows:

$$\hat{\mu}_t = \frac{1}{M}\sum_{j=1}^M x_t^{(j)}, \quad \hat{\Sigma}_t = \frac{1}{M}\sum_{j=1}^M (x_t^{(j)} - \hat{\mu}_t)(x_t^{(j)} - \hat{\mu}_t^{(j)}).$$

If we set M sufficiently large, (7.9) holds with high probability.

The remaining problem is how to get samples from p_t. A simple way for this is to use normal distributions; since p_t is defined by $z_1(x) = \sigma\|x\|_2^2/2$, (7.8) and (7.13), the distribution p_t is a multidimensional truncated normal distribution over $\mathcal{K}$ expressed as

$$p_t(x) \propto \exp(-\sigma\eta t\|x-\theta_t\|_2^2/2) \qquad (x \in \mathcal{K}'),$$
$$p_t(x) = 0 \qquad (x \in \mathbb{R}^d \setminus \mathcal{K}'),$$

where $\theta_t = \frac{1}{t}\sum_{j=1}^{t-1}(\hat{\mu}_j - \frac{1}{\sigma}\hat{g}_j)$. Hence, by sampling x from a normal distribution $\mathcal{N}(\theta_t, \frac{1}{\sigma\eta t}I)$ until $x \in \mathcal{K}'$, we can get x following p_t. Note that this procedure cannot, however, always terminate in polynomial time, though it will be practical enough in many cases. Even if the simple procedure does not work well, we can apply an alternative polynomial-time sampling method based on MCMC [127], since p_t is a log-concave distribution. For more efficient ways of computing $\hat{\mu}$ and $\hat{\Sigma}$ and sampling from p_t, see, e.g., [20, 135].

7.4.4 Choice of Exploration Parameter α_t

It is necessary to choose α_t so that $a_t = \hat{\mu}_t + \alpha_t B_t u_t$ is a feasible solution, i.e., $a_t \in \mathcal{K}$. The following proposition provides a sufficient condition for this.

Proposition 7.4.1. *If α_t is bounded as $0 < \alpha_t \le 1/9 + r\sqrt{t\eta\sigma/2}$ then $a_t = \hat{\mu}_t + \alpha_t B_t u_t$ is in $\mathcal{K}$.*

Proof. Let α_{t1} and α_{t2} be positive numbers such that $\alpha_{t1} \le 1/9$, $\alpha_{t2} \le r\sqrt{t\eta\sigma/2}$ and $\alpha_t = \alpha_{t1} + \alpha_{t2}$. Since a_t can be expressed as $a_t = \hat{\mu}_t + \alpha_{t1}B_t u_t + \alpha_{t2} B_t u_t$ and since all points of $\mathcal{K}'$ are r-interior of $\mathcal{K}$, it suffices to show that (i) $\hat{\mu}_t + \alpha_{t1}B_t u_t \in \mathcal{K}'$ and (ii) $\|\alpha_{t2}B_t u_t\|_2 \le r$.

From Lemma 7.4.4, $\|\hat{\mu}_t + \alpha_{t1}B_t u_t - \mu_t\|_{\Sigma_t^{-1}} \le 1/\mathrm{e}$ implies $\hat{\mu}_t + \alpha_{t1}B_t u_t \in \mathcal{K}'$. From the triangle inequality, we have

$$\begin{aligned}\|\hat{\mu}_t + \alpha_{t1}B_t u_t - \mu_t\|_{\Sigma_t^{-1}} &\le \|\hat{\mu}_t - \mu_t\|_{\Sigma_t^{-1}} + \alpha_{t1}\|B_t u_t\|_{\Sigma_t^{-1}} \\ &\le \frac{1}{9} + \frac{1}{9}\|\Sigma_t^{-1/2}B_t u_t\|_2 \le \frac{1}{9}(1 + \|\Sigma_t^{-1/2}B_t\|_2). \end{aligned} \tag{7.14}$$

From (7.9), we have $\|\Sigma_t^{-1/2}B_t\|_2 \le 2$. Combining this and (7.14), we have $\|\hat{\mu}_t + \alpha_{t1}B_t u_t - \mu_t\|_{\Sigma_t^{-1}} \le 1/3 \le 1/\mathrm{e}$, which implies that (i) holds.

Since $\eta z_t(x)$ is a $(t\eta\sigma)$-strongly convex function, from Lemma 7.4.2, the covariance matrix $\Sigma_t = \mathrm{Cov}(p_t)$ is bounded as $\|\Sigma_t\|_2 \le 1/(t\eta\sigma)$. From this and (7.9), we have $\|\hat{\Sigma}_t\|_2 \le 2/(t\eta\sigma)$. Accordingly, we have

$$\|B_t\|_2 \le \sqrt{\|\hat{\Sigma}_t\|_2} \le \sqrt{2/(t\eta\sigma)}. \tag{7.15}$$

From this, $\|u_t\| = 1$, and $\alpha_t \le r\sqrt{t\eta\sigma/2}$, we have (ii). □

Proposition 7.4.1 implies that, under the assumption of $0 < \alpha_t \le 1/9 + r\sqrt{t\eta\sigma/2}$, for arbitrary $x \in \mathcal{K}'$, the value of $\hat{f}_t(x)$ can be defined by (7.10). In addition, we have $\hat{f}_t(x) \in [-1, 1]$ for $x \in \mathcal{K}'$ since $\hat{f}_t(x)$ is defined to be a convex combination of values of $f_t(y)$ for $y \in \mathcal{K}$. Hereafter, we suppose that $0 < \alpha_t \le 1/9 + r\sqrt{t\eta\sigma/2}$ holds.

7.5 Regret Analysis

This section shows regret upper bounds for Algorithm 11.

7.5.1 Main Results

We analyze the expected regret for the case in which α_t is defined by

$$\alpha_t = \min\left\{\frac{1}{9} + r\sqrt{\frac{t\eta\sigma}{2}}, \sqrt{d}\right\}. \tag{7.16}$$

We also assume that the learning rate η is bounded as

$$\eta \leq \frac{\alpha_t}{2d\log(50T)}. \tag{7.17}$$

We then have the following regret bound:

Theorem 7.5.1. *Suppose that α_t is chosen as (7.16) and that η satisfies (7.17). Then, for the output of Algorithm 11 and for arbitrary $x^* \in \mathcal{K}'$, the regret is bounded as*

$$\mathbf{E}[R_T(x^*)] = O\left(\frac{d\beta\log T}{\sigma\eta} + \eta dT + \min\left\{\eta d^2 T, \frac{d^2\log d}{r^2\sigma}\right\}\right),$$

where the expectation is taken w.r.t. the randomness of the algorithm.

From this theorem, by setting $\eta = \Theta(T^{-1/2})$ (ignoring factors in d, β, σ, r), we obtain regret bound of $\tilde{O}(\sqrt{T})$.[3] More precisely, if $r > 0$, we have the following regret bound:

Corollary 7.5.1. *If we set $\eta = \sqrt{(\beta\log T)/(\sigma T)}$, and if (7.16), (7.17), and $r > 0$ hold, for all $x^* \in \mathcal{K}'$, we have*

$$\mathbf{E}[R_T(x^*)] = O\left(d\sqrt{\frac{\beta T\log T}{\sigma}} + \frac{d^2\log d}{r^2\sigma}\right).$$

In addition, even if $r = 0$, e.g., even when $\mathcal{K}' = \mathcal{K}$, we have the following regret bound:

Corollary 7.5.2. *If we set $\eta = \sqrt{(\beta\log T)/(d\sigma T)}$, and if (7.16) and (7.17) hold, for all $x^* \in \mathcal{K}'$, we have*

$$\mathbf{E}[R_T(x^*)] = O\left(d^{3/2}\sqrt{\frac{\beta T\log T}{\sigma}}\right).$$

7.5.2 Auxiliary Lemmas

Lemma 7.5.1. *Suppose that $\mathcal{K} \subseteq \mathbb{R}^d$ is a convex set including the zero vector, and that there exists a σ-strongly convex function $f : \mathcal{K} \to [-1, 1]$. Then, for all $x \in \mathcal{K}'$, we have $\sigma\|x\|_2^2 \leq 16$.*

Proof. From the σ-strong convexity of f, we have

$$f(0) + f(x) - 2f\left(\frac{1}{2}x\right) \geq \sigma\left\|\frac{1}{2}x\right\|_2^2 = \frac{\sigma}{4}\|x\|_2^2.$$

Hence, we have

$$\sigma\|x\|_2^2 \leq 4\left(f(0) + f(x) - 2f\left(\frac{1}{2}x\right)\right) \leq 16,$$

where the last inequality follows from $f(y) \in [-1, 1]$ for $y \in \mathcal{K}$. □

[3]Note that, if the number T of rounds is sufficiently large and if the parameter η is of order $O(T^{-1/2})$, then the condition (7.17) is automatically satisfied.

7.5.3 Proof of Theorem 7.5.1

To prove Theorem 7.5.1, we start by bounding the regret $R_T(x^*)$ by means of the smoothed objectives $\hat{f}_t$ defined by (7.10), on the basis of Lemmas 7.4.1 and 7.4.2.

Lemma 7.5.2. *For arbitrary $x^* \in \mathcal{K}'$, the regret for Algorithm 11 is bounded as*

$$\mathbf{E}[R_T(x^*)] \leq \sum_{t=1}^{T} \left(\mathbf{E}[\hat{f}_t(\hat{\mu}_t) - \hat{f}_t(x^*)] + \frac{\beta(\alpha_t^2 + 1)}{\eta\sigma t} \right).$$

Proof. Since f_t is a β-smooth function, we have

$$\begin{aligned}\mathbf{E}[f_t(a_t)] = \mathbf{E}[f_t(\hat{\mu}_t + \alpha_t B_t u_t)] &\leq \mathbf{E}\left[f_t(\hat{\mu}_t) + \alpha_t \nabla f_t(\hat{\mu}_t)^\top B_t u_t + \frac{\beta}{2}\|\alpha_t B_t u_t\|_2^2\right] \\ &\leq \mathbf{E}\left[f_t(\hat{\mu}_t)\right] + \frac{\beta\alpha_t^2\|B_t\|_2^2}{2} \leq \mathbf{E}\left[\hat{f}_t(\hat{\mu}_t)\right] + \frac{\beta\alpha_t^2}{t\eta\sigma},\end{aligned}$$

where the first inequality comes from (7.3), the second inequality follows from that $\mathbf{E}[u_t] = 0$ and that $\|u_t\|_2 = 1$, and the last inequality comes from (7.7) and (7.15). Similarly, from (7.7) and (7.15), we have

$$-\mathbf{E}[f_t(x^*)] \leq -\mathbf{E}[\hat{f}_t(x^*) + \beta\|B_t\|_2^2/2] \leq -\mathbf{E}[\hat{f}_t(x^*) + \beta/(t\eta\sigma)].$$

By combining the above two inequalities and taking the sum for $t \in [T]$, we obtain the bound for $\mathbf{E}[R_T(x^*)]$. □

We can provide a bound for the value $\sum_{t=1}^{T}(\hat{f}_t(x^*) - \hat{f}_t(\hat{\mu}_t))$ by combining an analysis for continuous multiplicative weight update methods [11, 79], (7.12), Lemma 7.4.2, and assumption of (7.9).

Lemma 7.5.3. *Suppose that η satisfies* (7.17). *For arbitrary $x^* \in \mathcal{K}'$, we have*

$$\sum_{t=1}^{T} \mathbf{E}[\hat{f}_t(\hat{\mu}_t) - \hat{f}_t(x^*)] \leq \frac{4d}{\eta}\log\frac{1}{\gamma} + 4\gamma T + \sum_{t=1}^{T}\left(\frac{4\eta d^2}{\alpha_t^2} + \frac{2\beta}{\eta\sigma t}\right) + 8.$$

Proof. Since f_t is σ-strongly convex, from Lemma 7.4.1, $\hat{f}_t$ is also σ-strongly convex. Hence, it holds for all $x^* \in \mathcal{K}'$ that

$$\mathbf{E}[\hat{f}_t(\hat{\mu}_t) - \hat{f}_t(x^*)] \leq -\mathbf{E}\left[\nabla\hat{f}_t(\hat{\mu}_t)^\top(x^* - \hat{\mu}_t) + \frac{\sigma}{2}\|x - \hat{\mu}_t\|_2^2\right] = -\mathbf{E}\left[\hat{g}_t^\top(x^* - \hat{\mu}_t) + \frac{\sigma}{2}\|x - \hat{\mu}_t\|_2^2\right],$$

where the inequality follows from (7.2) for strongly convex functions, and the equality follows from (7.12). By taking the sum of the above for $t = 1, 2, \ldots, T$ and using (7.13), we have

$$\sum_{t=1}^{T}\mathbf{E}[\hat{f}_t(\hat{\mu}_t) - \hat{f}_t(x^*)] \leq -\mathbf{E}\left[\sum_{t=1}^{T}\left(\hat{g}_t^\top(x^* - \hat{\mu}_t) + \frac{\sigma}{2}\|x - \hat{\mu}_t\|_2^2\right)\right] = -\mathbf{E}[z_{T+1}(x^*) - z_1(x^*)]. \tag{7.18}$$

We are able to give a bound for $z_{T+1}(x^*)$ by means of Z_{T+1} defined by (7.8). For arbitrary $x^* \in \mathcal{K}'$, define $\mathcal{K}_0 := (1-\gamma)x^* + \gamma\mathcal{K}' \subseteq \mathcal{K}'$. Then, Z_{T+1} can be evaluated as follows:

$$\begin{aligned} Z_{T+1} &= \int_{x\in\mathcal{K}'} \exp(-\eta z_{T+1}(x))\mathrm{d}x \geq \int_{x\in\mathcal{K}_0} \exp(-\eta z_{T+1}(x))\mathrm{d}x \\ &= \int_{x\in\mathcal{K}'} \gamma^d \exp(-\eta z_{T+1}((1-\gamma)x^* + \gamma x))\mathrm{d}x \\ &\geq \int_{x\in\mathcal{K}'} \gamma^d \exp(-\eta(1-\gamma)z_{T+1}(x^*) - \eta\gamma z_{T+1}(x))\mathrm{d}x \\ &= \exp(d\log\gamma - \eta(1-\gamma)z_{T+1}(x^*)) \int_{x\in\mathcal{K}'} \exp(-\eta\gamma z_{T+1}(x))\mathrm{d}x, \end{aligned}$$

where the first inequality follows from the fact that $\mathcal{K}_0 \subseteq \mathcal{K}'$ and that the integrand is nonnegative, the second equality is given by the change of variables $x \leftarrow (1-\gamma x^*) + \gamma x$, and the second inequality holds since z_{T+1} is a convex function. Hence, $-z_{T+1}(x^*)$ can be bounded as

$$-z_{T+1}(x^*) \leq \frac{1}{\eta(1-\gamma)}\left(d\log\frac{1}{\gamma} + \log(Z_{T+1}) - \log\left(\int_{x\in\mathcal{K}'} \exp(-\eta\gamma z_{T+1}(x))\mathrm{d}x\right)\right) \tag{7.19}$$

Define $\tilde{x} \in \operatorname*{arg\,max}_{x\in\mathcal{K}'} \sum_{t=1}(\hat{g}_t^\top(x-\hat{\mu}_t) + \frac{\sigma}{2}\|x-\hat{\mu}_t\|_2^2)$. We then have

$$\begin{aligned} &\int_{x\in\mathcal{K}'} \exp(-\eta\gamma z_{T+1}(x))\mathrm{d}x = \int_{x\in\mathcal{K}'} \exp\left(-\frac{\eta\gamma\sigma}{2}\|x\|_2^2 - \eta\gamma\sum_{t=1}^T(\hat{g}_t^\top(x-\hat{\mu}_t) + \frac{\sigma}{2}\|x-\hat{\mu}_t\|_2^2)\right)\mathrm{d}x \\ &\geq \int_{x\in\mathcal{K}'} \exp\left(-\frac{\eta\gamma\sigma}{2}\|x\|_2^2 - \eta\gamma\sum_{t=1}^T(\hat{g}_t^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2)\right)\mathrm{d}x \\ &= \int_{x\in\mathcal{K}'} \exp\left(-\frac{\eta\gamma\sigma}{2}\|x\|_2^2\right)\mathrm{d}x \cdot \exp\left(-\eta\gamma\sum_{t=1}^T(\hat{g}_t^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2)\right) \\ &\geq \int_{x\in\mathcal{K}'} \exp\left(-\frac{\eta\sigma}{2}\|x\|_2^2\right)\mathrm{d}x \cdot \exp\left(-\eta\gamma\sum_{t=1}^T(\hat{g}_t^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2)\right) \\ &= Z_1 \exp\left(-\eta\gamma\sum_{t=1}^T(\hat{g}_t^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2)\right), \end{aligned}$$

where the first equality follows from (7.13), the first inequality follows from the definition of $\tilde{x}$, the second inequality follows from $\gamma \leq 1$, and the last equality follows from the definition (7.8) of Z_t and $z_1(x) = \frac{\sigma}{2}\|x\|_2^2$. Hence, we have

$$\begin{aligned} -\mathbf{E}\left[\log\int_{x\in\mathcal{K}'} \exp(-\eta\gamma z_{T+1}(x))\mathrm{d}x\right] &\leq -\log Z_1 + \eta\gamma\sum_{t=1}^T \mathbf{E}\left[\hat{g}_t^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2\right] \\ &= -\log Z_1 + \eta\gamma\sum_{t=1}^T \mathbf{E}\left[\nabla\hat{f}_t(\hat{\mu}_t)^\top(\tilde{x}-\hat{\mu}_t) + \frac{\sigma}{2}\|\tilde{x}-\hat{\mu}_t\|_2^2\right] \\ &\leq -\log Z_1 + \eta\gamma\sum_{t=1}^T \mathbf{E}\left[\hat{f}_t(\tilde{x}) - \hat{f}_t(\hat{\mu}_t)\right] \\ &\leq -\log Z_1 + 2\eta\gamma T, \end{aligned}$$

where the equality follows from (7.12), the second inequality holds since $\hat{f}_t$ is σ-strongly convex from Lemma 7.4.1 and hence (7.2) applies, and the last inequality follows from $\hat{f}_t(x) \in [-1, 1]$ for $x \in \mathcal{K}$.

Combining this and (7.19), we have

$$-\mathbf{E}[z_{T+1}(x^*)] \leq \frac{1}{\eta(1-\gamma)} \left(d \log \frac{1}{\gamma} + 2\eta\gamma T + \mathbf{E}[\log Z_{T+1} - \log Z_1] \right). \tag{7.20}$$

The term $\mathbf{E}[\log Z_{T+1} - \log Z_1]$ can be expressed as

$$\mathbf{E}[\log Z_{T+1} - \log Z_1] = \mathbf{E}\left[\log \frac{Z_{T+1}}{Z_1}\right] = \sum_{t=1}^{T} \mathbf{E}\left[\log \frac{Z_{t+1}}{Z_t}\right].$$

We will construct a bound for the left-hand side by bounding the values $\mathbf{E}[\log \frac{Z_{t+1}}{Z_t}]$. We have

$$\begin{aligned}
\frac{Z_{t+1}}{Z_t} &= \int_{x \in \mathcal{K}'} \frac{\exp(-\eta z_t(x))}{Z_t} \exp\left(-\eta \hat{g}_t^\top (x - \hat{\mu}_t) - \frac{\eta\sigma}{2}\|x - \hat{\mu}_t\|_2^2\right) \mathrm{d}x \\
&\leq \int_{x \in \mathcal{K}'} p_t(x) \exp(-\eta \hat{g}_t^\top (x - \hat{\mu}_t)) \mathrm{d}x \\
&\leq \int_{x \in \mathcal{K}'} p_t(x)(1 - \eta \hat{g}_t^\top (x - \hat{\mu}_t) + (\eta \hat{g}_t^\top (x - \hat{\mu}_t))^2) \mathrm{d}x \\
&\qquad + \int_{x \in \mathcal{K}', \eta \hat{g}_t^\top (x - \hat{\mu}_t) < -1} p_t(x) \exp(-\eta \hat{g}_t^\top (x - \hat{\mu}_t)) \mathrm{d}x \\
&= 1 - \eta \hat{g}_t^\top (\mu_t - \hat{\mu}_t) + \int_{x \in \mathcal{K}'} p_t(x)(\eta \hat{g}_t^\top (x - \hat{\mu}_t))^2 \mathrm{d}x \qquad (7.21) \\
&\qquad + \int_{x \in \mathcal{K}', \eta \hat{g}_t^\top (x - \hat{\mu}_t) < -1} p_t(x) \exp(-\eta \hat{g}_t^\top (x - \hat{\mu}_t)) \mathrm{d}x, \qquad (7.22)
\end{aligned}$$

where the first inequality follows from the definition (7.8) of p_t and $\eta\sigma\|x - \hat{\mu}_t\|_2^2 \geq 0$, the second inequality holds since we have $\exp(y) \leq 1 + y + y^2$ for $y \leq 1$, and the last equality follows from the definition of μ_t. The third term of (7.22) can be expressed as follows:

$$\begin{aligned}
\int_{x \in \mathcal{K}'} p_t(x)(\eta \hat{g}_t^\top (x - \hat{\mu}_t))^2 \mathrm{d}x &= \eta^2 \hat{g}_t^\top \left(\int_{x \in \mathcal{K}'} p_t(x)(x - \hat{\mu}_t)(x - \hat{\mu}_t)^\top \mathrm{d}x \right) \hat{g}_t \\
&= \eta^2 \hat{g}_t^\top \left(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top \right) \hat{g}_t \\
&= \frac{d^2 \eta^2 f_t(a_t)^2}{\alpha_t^2} u_t^\top B_t^{-1} \left(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top \right) B_t^{-1} u_t \\
&\leq \frac{d^2 \eta^2}{\alpha_t^2} \left(\|B_t^{-1} \Sigma_t B_t^{-1}\|_2 + \|\mu_t - \hat{\mu}_t\|_{\hat{\Sigma}_t}^2 \right) \leq \frac{4 d^2 \eta^2}{\alpha_t^2}, \qquad (7.23)
\end{aligned}$$

where the second equality follows from the definitions of Σ_t and μ_t, the second equality follows from the definition (7.11) of $\hat{g}_t$, the first inequality follows from $|f_t(a_t)| \leq 1$, $\|u_t\|_2 = 1$ and $B_t B_t = \hat{\Sigma}_t$ and the last inequality follows from (7.9). From (7.23) and Lemma 3.4.2, we can bound the last term of (7.22). Indeed, $-\eta \hat{g}_t^\top (x - \hat{\mu}_t)$ follows a one dimensional log-concave distribution when x follows the log-concave distribution p_t because the log-concavity of distributions is preserved under any linear transformation (see, e.g., [149]), and consequently, we can apply Lemma 3.4.2 to obtain the following inequality:

$$\int_{x \in \mathcal{K}', \eta \hat{g}_t^\top (x - \hat{\mu}_t) < -1} p_t(x) \exp(-\eta \hat{g}_t^\top (x - \hat{\mu}_t)) \mathrm{d}x \leq \frac{\exp(3 - \frac{\alpha_t}{2d\eta})}{1 - \exp(1 - \frac{\alpha_t}{2d\eta})} \leq 50 \exp\left(-\frac{\alpha_t}{2d\eta}\right), \tag{7.24}$$

where the last inequality follows from the assumption of $\alpha_t \geq 4d\eta$. Combining (7.22), (7.23), (7.24) and the fact that $\log(1+y) \leq y$, we have

$$\begin{aligned}
&\mathbf{E}\left[\log \frac{Z_{t+1}}{Z_t}\right] \leq \mathbf{E}\left[\frac{Z_{t+1}}{Z_t}\right] - 1 \\
&\leq -\eta\, \mathbf{E}\left[\hat{g}_t^\top(\mu_t - \hat{\mu}_t)\right] + \eta^2\, \mathbf{E}\left[\hat{g}_t^\top \left(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top\right) \hat{g}_t\right] + 50 \exp\left(-\frac{\alpha_t}{2d\eta}\right). \quad (7.25)
\end{aligned}$$

The first term of (7.25) can be bounded as follows:

$$\begin{aligned}
\mathbf{E}[\hat{g}_t^\top(\hat{\mu}_t - \mu_t)] &= \mathbf{E}[\nabla \hat{f}_t(\hat{\mu}_t)^\top(\hat{\mu}_t - \mu_t)] \leq \mathbf{E}[\nabla \hat{f}_t(\mu_t)^\top(\hat{\mu}_t - \mu_t)] + \beta\|\hat{\mu}_t - \mu_t\|_2^2 \\
&= \beta\|\hat{\mu}_t - \mu_t\|_2^2 \leq \beta\|\hat{\mu}_t - \mu_t\|_{\Sigma_t^{-1}}^2 \|\Sigma_t\|_2 \leq \frac{\beta}{\eta\sigma t}, \quad (7.26)
\end{aligned}$$

where the first equality comes from (7.12), the first inequality follows from (7.3) for β-smooth convex functions, the second equality follows from $\mathbf{E}[\hat{\mu}_t|\mu_t] = \mu_t$ in (7.9), and the last inequality follows from (7.9) and Lemma 7.4.2 and the $(\eta\sigma t)$-strong convexity of z_t. The second term of (7.25) can be bounded as

$$\begin{aligned}
\mathbf{E}\left[\hat{g}_t^\top \left(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top\right) \hat{g}_t\right] &= \frac{d^2 f_t(a_t)^2}{\alpha_t^2}(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top) \bullet (\mathbf{E}[B_t^{-1} u_t u_t^\top B_t^{-1}]) \\
&\leq \frac{d}{\alpha_t^2}(\Sigma_t + (\mu_t - \hat{\mu}_t)(\mu_t - \hat{\mu}_t)^\top) \bullet \hat{\Sigma}_t^{-1} \leq \frac{2d^2}{\alpha_t^2}, \quad (7.27)
\end{aligned}$$

where the first equality follows from (7.11), and the last ineqality follows form (7.9). Combining (7.25), (7.26) and (7.27), we have

$$\mathbf{E}\left[\log \frac{Z_{t+1}}{Z_t}\right] \leq \frac{\beta}{\sigma t} + \frac{2\eta^2 d^2}{\alpha_t^2} + 50 \exp\left(-\frac{\alpha_t}{2d\eta}\right). \quad (7.28)$$

Hence, we have

$$\mathbf{E}[\log Z_T - \log Z_1] = \sum_{t=1}^T \mathbf{E}\left[\frac{Z_{t+1}}{Z_t}\right] \leq 1 + \sum_{t=1}^T \left(\frac{2\eta^2 d^2}{\alpha_t^2} + \frac{\beta}{\sigma t}\right). \quad (7.29)$$

Combining this and (7.20), we obtain

$$\begin{aligned}
-\mathbf{E}[z_{T+1}(x^*)] &\leq \frac{1}{\eta(1-\gamma)}\left(d \log \frac{1}{\gamma} + 2\eta\gamma T + 1 + \sum_{t=1}^T \left(\frac{2\eta^2 d^2}{\alpha_t^2} + \frac{\beta}{\sigma t}\right)\right) \\
&\leq \frac{4d}{\eta} \log \frac{1}{\gamma} + 4\gamma T + \sum_{t=1}^T \left(\frac{4\eta d^2}{\alpha_t^2} + \frac{2\beta}{\eta\sigma t}\right),
\end{aligned}$$

where the second inequality can be derived from $\gamma \leq 1/2$. From this and (7.18), we have

$$\mathbf{E}[\hat{f}_t(\hat{\mu}_t) - \hat{f}_t(x^*)] \leq \frac{4d}{\eta} \log \frac{1}{\gamma} + 4\gamma T + \sum_{t=1}^T \left(\frac{4\eta d^2}{\alpha_t^2} + \frac{2\beta}{\eta\sigma t}\right) + z_1(x^*).$$

From Lemma 7.5.1, we have $z_1(x^*) = \frac{\sigma}{2}\|x^*\|_2^2 \leq 8$. By combining this and the above-displayed inequality, we obtain Lemma 7.5.3. $\square$

We shall complete the proof of Theorem 7.5.1 by means of Lemmas 7.5.2 and 7.5.3.

Proof of Theorem 7.5.1. By combining the above and Lemmas 7.5.2 and 7.5.3, and by setting $\gamma = \frac{1}{T}$, we obtain

$$\mathbf{E}[R_T(x^*)] \leq 12 + \frac{4d}{\eta} \log T + \sum_{t=1}^{T} \left(\frac{4\eta d^2}{\alpha_t^2} + \frac{\beta(\alpha_t^2 + 3)}{\eta \sigma t} \right).$$

If we set α_t by (7.16), since $\alpha_t^2 \leq d$ holds, we have

$$\sum_{t=1}^{T} \frac{\beta(\alpha_t^2 + 3)}{\eta\sigma t} \leq \frac{4d\beta}{\eta} \sum_{t=1}^{T} \frac{1}{t} \leq \frac{4d\beta \log(\mathrm{e}T)}{\eta}.$$

Combining the above two displayed inequalities and the fact that $1 \leq \beta/\sigma$, we have

$$\mathbf{E}[R_T(x^*)] \leq 12 + \frac{8d\beta\log(\mathrm{e}T)}{\eta\sigma} + 4\eta d^2 \sum_{t=1}^{T} \frac{1}{\alpha_t^2}. \tag{7.30}$$

Let us consider bounding $\sum_{t=1}^{T} 1/\alpha_t^2$. From (7.16), we have

$$\sum_{t=1}^{T} \frac{1}{\alpha_t^2} = \sum_{t=1}^{T} \max \left\{ \frac{2}{(\sqrt{2}/9 + r\sqrt{t\eta\sigma})^2}, \frac{1}{d} \right\} \leq 81T, \tag{7.31}$$

where the inequality holds for arbitrary $r \geq 0$ (even if $r = 0$) since we have $1/\alpha_t^2 \leq 81$. Assuming $r > 0$, we obtain a tiger bound for it; Denote $T' := \lfloor \frac{2d}{r^2\eta\sigma} \rfloor$. Since $2/(\sqrt{2}/9 + r\sqrt{t\eta\sigma})^2 \leq 1/d$ holds for $t > T'$, we have

$$\begin{aligned} \sum_{t=1}^{T} \frac{1}{\alpha_t^2} &\leq \sum_{t=1}^{T'} \frac{2}{(\sqrt{2}/9 + r\sqrt{t\eta\sigma})^2} + \frac{\max\{T - T', 0\}}{d} \leq \frac{2}{r^2\eta\sigma} \sum_{t=1}^{T'} \frac{1}{t + 2/(81r^2\eta\sigma)} + \frac{T}{d} \\ &\leq \frac{2}{r^2\eta\sigma} \log(1 + 81r^2\eta\sigma T'/2) + \frac{T}{d} \leq \frac{2}{r^2\eta\sigma} \log(1 + 81d) + \frac{T}{d}, \end{aligned} \tag{7.32}$$

where the third inequality follows from the fact that $\sum_{t=1}^{T'} 1/(t+y) \leq \log(1 + T'/y)$ holds for any $y > 0$, and the last inequality follows from $T' \leq \frac{2d}{r^2\eta\sigma}$. Combining (7.30), (7.31) and (7.32), we obtain

$$\begin{aligned} \mathbf{E}[R_T(x^*)] &\leq 12 + \frac{8d\beta\log(\mathrm{e}T)}{\eta\sigma} + 4\eta d^2 \min\left\{ 81T, \frac{2}{r^2\eta\sigma}\log(1 + 81d) + \frac{T}{d} \right\} \\ &= O\left(\frac{d\beta \log T}{\eta\sigma} + \eta d T + \min\left\{ \eta d^2 T, \frac{d^2 \log d}{r^2 \sigma} \right\} \right). \end{aligned}$$

□

7.6 Discussion

We discuss the possibility of removing the assumption of $r > 0$, i.e., the assumption that the optimal solutions (or the benchmark set) are interiors, in Corollary 7.5.1.

From Lemmas 7.5.2 and 7.5.3, if we can set $\alpha_t = \tilde{\Theta}(\sqrt{d})$, we have

$$\mathbf{E}[R_T(x^*)] = \tilde{O}\left(\frac{d\beta}{\sigma\eta} + \eta d T \right) = \tilde{O}\left(d\sqrt{\frac{\beta T}{\sigma}} \right) \tag{7.33}$$

by setting $\eta = \tilde{\Theta}(\sqrt{\frac{\beta}{\sigma T}})$. Setting $\alpha_t = \tilde{\Theta}(\sqrt{d})$, however, may cause an infeasible action $a_t = \hat{\mu}_t + \alpha_t B_t u_t \notin \mathcal{K}$ in Algorithm 11 without the assumption on r, and consequently, may make it impossible to bound the regret. The possibility of $a_t \notin \mathcal{K}$ seems to be, on the other hand, quite small if we set $\alpha_t = c\sqrt{d}$ with a small constant $c > 0$, even when $\mathcal{K}' = \mathcal{K}$. Under the assumption that the possibility of $a_t \notin \mathcal{K}$ is sufficiently small, our analysis in Section 7.5 works similarly and leads to a regret bound of $\tilde{O}(d\sqrt{T})$. A sufficient condition for this assumption of small possibilities can be formulated as follows:

Conjecture 7.6.1. *Let $\mathcal{K} \subseteq \mathbb{R}^d$ be a d-dimensional convex set. Let p be a multidimensional truncated normal distribution over $\mathcal{K}$, i.e., $p(x) \propto \exp(-\|x\|_2^2/2)$ for $x \in \mathcal{K}$ and $p(x) = 0$ for $x \in \mathbb{R}^d \setminus \mathcal{K}$. Let $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$ be the mean and the covariance matrix of p, respectively. Then, for a random variable u following a uniform distribution $U(\mathbb{S}^{\mathrm{d}})$ over the unit sphere $\mathbb{S}^d$, the probability of that $\mu + \alpha\Sigma^{1/2}u$ is not in $\mathcal{K}$ is bounded as*

$$\operatorname*{Prob}_{u \sim U(\mathbb{S}^{\mathrm{d}})}\left[\mu + \alpha\Sigma^{1/2}u \notin \mathcal{K}\right] \leq \exp\left((1 + \log d)^{O(1)} - \frac{\sqrt{d}}{\alpha \cdot (1 + \log d)^{O(1)}}\right)$$

for all $\alpha > 0$.

If this conjecture holds, we have a regret bound of $\mathbf{E}[R_T(x^*)] = \tilde{O}(d\sqrt{T})$ for arbitrary $x^* \in \mathcal{K}$, without the assumption of interior optimal solutions, i.e., the bound holds even for the case of $\mathcal{K}' = \mathcal{K}$.

Since a truncated normal distribution is a log-concave distribution, in Conjecture 7.6.1, the probability of $\mu + \alpha\Sigma^{1/2}u \notin \mathcal{K}$ is equal to zero for $\alpha < 1/\mathrm{e}$, from Lemma 7.4.4. This fact is used to prove Proposition 7.4.1. The question is if we can obtain a bound of the probability for $\alpha = \tilde{\Omega}(\sqrt{d})$.

7.7 Conclusion

This chapter considered bandit convex optimization problems with strongly-convex and smooth objectives. We provided an algorithm with tight regret bounds, w.r.t. the number T of rounds as well as the dimension d of the feasible region, under milder assumptions than existing works. More precisely, we gave a regret bound of $\tilde{O}(d\sqrt{T} + d^2/r^2)$ under the assumption that the optimal solutions (or the benchmark set) are r-interiors, and without this assumption, our algorithm achieves a regret of $\tilde{O}(d^{3/2}\sqrt{T})$. Our algorithm, further, works given only a membership oracle for the feasible region, without self-concordant barriers.

The assumption of interior optimal solutions, however, might be abundant, and the tight regret bounds might apply to more general problem settings. To prove that our algorithm achieves $\tilde{O}(d\sqrt{T})$-regret without the assumption, we introduced an approach based on the inequality for probability regarding log-concave distributions (more precisely, multi-dimensional truncated normal distributions) in Conjecture 7.6.1. We leave it as a feature work to prove this conjecture.

Another future direction is to pursue tighter regret bounds for general BCO without assumptions of strong convexity and smoothness of objective functions. For this more general problem, there is a larger gap w.r.t. d between the upper bound of $\tilde{O}(d^{9.5}\sqrt{T})$ and the lower bound of $\Omega(d\sqrt{T})$. If a lower bound of $\Omega(d^{3/2}\sqrt{T})$ would hold as conjectured in [34], it would, together with our results, imply a $\tilde{\Omega}(\sqrt{d})$ gap between minimax regrets for BCO with strongly-convex and smooth losses and that for general BCO.

Chapter 8

Online Portfolio Selection with Combinatorial Constraints

Online portfolio selection is a sequential decision-making problem in which a learner repetitively selects a portfolio over a set of assets, aiming to maximize long-term return. In this chapter, we study the problem with the cardinality constraint that the number of assets in a portfolio is restricted to be at most k, and consider two scenarios: (i) in the *full-feedback setting*, the learner can observe price relatives (rates of return to cost) for all assets, and (ii) in the *bandit-feedback setting*, the learner can observe price relatives only for invested assets. We propose efficient algorithms for these scenarios, which achieve sublinear regrets. We also provide regret (statistical) lower bounds for both scenarios which nearly match the upper bounds when k is a constant. In addition, we give a computational lower bound, which implies that no algorithm maintains both computational efficiency, as well as a small regret upper bound.

8.1 Introduction

Online portfolio selection [47, 123] is a fundamental problem in financial engineering, in which a learner sequentially selects a portfolio over a set of assets, aiming to maximize cumulative wealth. For this problem, principled algorithms (e.g., the universal portfolio algorithm [47]) have been proposed, which behave as if one knew the empirical distribution of future market performance. On the other hand, these algorithms work only under the strong assumption that we can hold portfolios of arbitrary combinations of assets, and that we can observe price relatives, the multiplicative factors by which prices change, for all assets. Due to these limitations, this framework does not directly apply to such real-world applications as investment in advertising or R&D, where the available combination of assets is restricted and/or price relatives (return on investment) are revealed only for assets that have been invested in.

In order to overcome such issues, we consider the following problem setting: Suppose that there are T rounds and a market has d assets, represented by $[d] := \{1, \ldots, d\}$. In each round t, we design a portfolio, that represents the proportion of the current wealth invested in each of the d assets. That is, a *portfolio* can be expressed as a vector $x_t = [x_{t1}, \ldots, x_{td}]^\top$ such that $x_{ti} \geq 0$ for all $i \in [d]$ and $\sum_{i=1}^{d} x_{ti} \leq 1$. The combination of assets is restricted with a set of *available combinations* $\mathcal{S} \subseteq 2^{[d]}$, that is, a portfolio x_t must satisfy $\mathrm{supp}(x_t) = \{i \in [d] \mid x_{ti} \neq 0\} \in \mathcal{S}$. Thus, in each period t, we choose S_t from $\mathcal{S}$ and determine a portfolio x_t only from assets in S_t.

Table 8.1: Regret bounds for the full-feedback setting.

Constraints	Upper bound by Algorithm 12	Lower bound
Single asset ($\mathcal{S} = \mathcal{S}_1$)	$R_T = O(\sqrt{T \log d})$	$R_T = \Omega(\sqrt{T \log d})$
Combination ($\mathcal{S} = \mathcal{S}_k$)	$R_T = O\left(\sqrt{Tk \log \frac{d}{k}}\right)$ (run in $T\binom{d}{k}\mathrm{poly}(k)$-time)	$R_T = \Omega\left(\sqrt{T \log \frac{d}{k}}\right)$ for $d \geq 17k$ and no $\mathrm{poly}(d,k,T)$-time algorithm achieves $R_T \leq T^{1-\delta}\mathrm{poly}(d,k)$

Table 8.2: Regret bounds for the bandit-feedback setting.

Constraint	Upper bound by Algorithm 13	Lower bound
Single asset ($\mathcal{S} = \mathcal{S}_1$)	$R_T = O(\sqrt{dT \log T})$	$R_T = \Omega(\sqrt{dT})$
Combination ($\mathcal{S} = \mathcal{S}_k$)	$R_T = O\left(\sqrt{Tk\binom{d}{k} \log T}\right)$ (run in $T\mathrm{poly}(d,k)$-time)	$R_T = \Omega\left(\sqrt{T \left(\frac{d}{Ck^3}\right)^k}\right)$ for $d > k$ and no $\mathrm{poly}(d,k,T)$-time algorithm achieves $R_T \leq T^{1-\delta}\mathrm{poly}(d,k)$

A typical example of $\mathcal{S}$ can be given by cardinality constraints, i.e., $\mathcal{S}_k := \{S \subseteq [d] \mid |S| = k\}$ for some $k \leq d$. We denote by $r_t = [r_{t1}, \ldots, r_{td}]^\top$ a *price relative vector*, where $1 + r_{ti}$ is the price relative for the i-th asset in the t-th period. Then the wealth A_T resulting from the sequentially rebalanced portfolios $x_1, \ldots, x_t$ is given by $A_T = \prod_{t=1}^T (1 + r_t^\top x_t)$. The best constant portfolio strategy earns the wealth $A_T^* := \max_x \prod_{t=1}^T (1 + r_t^\top x)$ subject to the constraint that x is a portfolio satisfying $\mathrm{supp}(x) \in \mathcal{S}$. The performance of our portfolio selection is measured by $R_T = \log A_T^* - \log A_T$, which we call *regret*. The reason that we use $\log A_T$ rather than A_T comes from capital growth theory [72, 106]. In terms of the observable information, we consider two different settings: (i) in the *full-feedback setting*, we can observe all the price relatives r_{ti} for $i = 1, \ldots, d$, and (ii) in the *bandit-feedback setting*, we can observe the price relatives r_{ti} only for $i \in S_t$. Note that in each round t a portfolio x_t has to be determined before knowing r_{ti} in either of the settings. Note also that we do not make any statistical assumption about the behavior of r_{ti}, but we assume that r_{ti} is bounded in a closed interval $[C_1, C_2]$, where C_1 and C_2 are constants satisfying $-1 < C_1 \leq C_2$.

Our problem is a generalization of the standard online portfolio selection problem. In fact, if portfolios combining all assets are available, i.e., if $\mathcal{S} = 2^{[d]}$, then our problem coincides with the standard online portfolio selection problem. For this special case, it has been shown that some online convex optimization (OCO) methods [81, 77, 152] (e.g., the online Newton step method) achieve regret of $O(d \log T)$, and that any algorithm will suffer from regret of $\Omega(d \log T)$ in the worst case [139].

Our contribution is twofold; algorithms with sublinear regret upper bounds, and analyses proving regret lower bounds. First, we propose the following two algorithms:

- Algorithm 12 for the full-feedback setting, achieving regret of $O(\sqrt{T \log |\mathcal{S}|})$.
- Algorithm 13 for the bandit-feedback setting, achieving regret of $O(\sqrt{Tk|\mathcal{S}| \log T})$, where k denotes the largest cardinality among elements in $\mathcal{S}$, i.e., $k = \max_{S \in \mathcal{S}} |S|$.

Tables 8.1 and 8.2 summarize the regret bounds for the special case in which the cardinality of assets is restricted to be at most 1 or at most k. As shown in Table 8.1, Algorithm 12 can achieve regret of $O(\sqrt{T}\mathrm{poly}(d))$ even if $k = \Omega(d)$ when $\mathcal{S}$ has an exponentially large size with respect to d. In such a case, however, Algorithm 12 requires exponentially large computational time. For the bandit-feedback setting, the regret upper bound can be exponential w.r.t. d if $k = \Omega(d)$, but it is still sublinear in T. One main idea behind our algorithms is to combine the multiplicative weight update method (MWU) [11, 60] (in the full-feedback setting) / multi-armed bandit algorithms (MAB) [16, 30] (in the bandit-feedback setting) with OCO. Specifically, for choosing the combination S_t of assets, we employ MWU/MAB, which are online decision making methods over a finite set of actions. For maintaining the proportion x_t of portfolios, we use OCO, that is, online decision making methods for convex objectives over a convex set of actions.

Second, we show regret lower bounds for both the full-feedback setting and the bandit-feedback setting where $\mathcal{S} = \mathcal{S}_k$, which give insight into the tightness of regret upper bounds achieved with our algorithms. As shown in Table 8.1, the proven lower bounds for the full-feedback setting are tight up to the $O(\sqrt{k})$ term. For the bandit-feedback setting, the lower bounds are also tight up to the $O(\sqrt{\log T})$ term, if $k = O(1)$. Note that, if $k = d$ then the problem coincides with the standard online portfolio selection problem, and hence, there exist algorithms achieving $R_T = O(\sqrt{T \log d})$. This implies that the assumption of $d = \Omega(k)$ is essential for proving the lower bounds of $\Omega(\sqrt{T})$. We also note that these *statistical* lower bounds are valid for arbitrary learners, including exponential-time algorithms. Besides statistical ones, we also show *computational* lower bounds suggesting that there is no polynomial-time algorithm achieving a regret bound with a sublinear term in T and a polynomial term w.r.t. d and k, unless $\mathbf{NP} \subseteq \mathbf{BPP}$. This means that we cannot improve the computational efficiency of Algorithm 12 to $O(\mathrm{poly}(d, k, T))$-time while preserving its regret upper bound.

To prove the regret lower bounds, we use three different techniques: for the statistical lower bound for the full-feedback setting, we consider a completely random market and evaluate how well the "best" strategy worked after observing the market behavior, in a similar way to that for the lower bound for MWU [11]; for the bandit-feedback setting, we construct a "good" combination $S^* \in \mathcal{S}$ of assets so that it is hard to distinguish it from the others, and bound the number of choosing this "good" combination via a technique similar to that used in the proof of the regret lower bound for MAB [16]; to prove the computational lower bound, we reduce the 3-dimensional matching problem (3DM), one of Karp's 21 NP-complete problems [104], to our problem.

8.2 Related Work

Online portfolio selection has been studied in many research areas, including finance, statistics, machine learning, and optimization [4, 47, 99, 123, 122] since Cover [47] formulated the problem setting and proposed a *universal portfolio algorithm* that achieves regret of $O(d \log T)$ with exponential computation cost. This regret upper bound was shown to be optimal by Ordentlich and Cover [139]. The computation cost was reduced by the celebrated work on the online gradient method of Zinkervich [169] for solving *online convex optimization* (OCO) [77, 152], a general framework including online portfolio selection, but the regret bound is $O(d\sqrt{T})$ and suboptimal for online portfolio selection. A breakthrough w.r.t. this suboptimality came with the *online Newton step* and the *follow-the-approximation-leader* method of Hazan et al. [81], which are computationally efficient and achieve regret of $O(d \log T)$ for a special case of OCO, including online portfolio selection. Among studies on online portfolio selection, the work by Das et al. [53] has a motivation similar to ours: the aim of selecting portfolios with a group-sparse structure. However, their problem setting differs from ours in that they did not put constraints about sparsity but, rather, defined regret containing regularizer inducing group sparsity, and that they supposed that a learner can observe price relatives for all assets after determining portfolios. In contrast to this, our work deals with the sparsity constraint on portfolios, and our methods work even for the bandit-feedback setting, in which feedbacks are observed only on assets that have been invested in.

Another closely related topic is the multi-armed bandit problem (MAB) [15, 16, 30]. For non-stochastic MAB problems, a nearly optimal regret bound is achieved by the Exp3 algorithm [16], which our algorithm strongly relies on. For combinatorial bandit problems [38, 42, 45] in which each arm corresponds to a subset, the work by Chen et al. [42] gives solutions to a wide range of problems. However, this work does not directly apply to our setting, because we need to maintain not only subsets S_t but also continuous variables x_t, and both of them affect regret.

Remark 8.2.1. When the reward A_T changes multiplicatively, the expectation of the logarithm $\mathbf{E}[\log A_T]$ can be regarded to be a more reasonable evaluation metrics than would be the expected reward $\mathbf{E}[A_T]$. This is supported by the following example: suppose that $(X_t)_{t=1}^T = ((X_t^{(1)}, X_t^{(2)}))_{t=1}^T$ are Bernoulli random variables such that $X_t^{(1)} = \begin{cases} 1.3 & \text{w. p. } 0.5 \\ 0.9 & \text{w. p. } 0.5 \end{cases}$, $X_t^{(2)} = \begin{cases} 2.0 & \text{w. p. } 0.5 \\ 0.4 & \text{w. p. } 0.5 \end{cases}$, and that X_t and $X_{t'}$ are independent random variables for $t \neq t'$. Note that we do not assume $X_t^{(1)}$ and $X_t^{(2)}$ to be independent. Define $A_T^{(1)} = \prod_{t=1}^T X_t^{(1)}$ and $A_T^{(2)} = \prod_{t=1}^T X_t^{(2)}$. Then, since $\mathbf{E}[X_t^{(1)}] = 1.1$ and $\mathbf{E}[X_t^{(1)}] = 1.2$, we have $\mathbf{E}[A_T^{(1)}] = 1.1^T < \mathbf{E}[A_T^{(2)}] = 1.2^T$, which implies that we prefer $A_T^{(1)}$ to $A_T^{(2)}$ when determining on the basis of the expectation. However, we can show that $\lim_{T\to\infty} A_T^{(1)} = \infty$ and $\lim_{T\to\infty} A_T^{(2)} = 0$ with probability one, respectively. In fact, if $A_T = \prod_{t=1}^T X_t$ is the product of i.i.d. random variables, we have

$$\lim_{T\to\infty} (A_T)^{\frac{1}{T}} = \exp\left(\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \log X_t\right) = \exp(\mathbf{E}[\log X_1]) \tag{8.1}$$

with probability one, where the last equality comes from the law of large numbers. Applying (8.1) to $A_T^{(1)}$ and $A_T^{(2)}$, we obtain $\lim_{T\to\infty}(A_T^{(2)})^{\frac{1}{T}} < 1 < \lim_{T\to\infty}(A_T^{(1)})^{\frac{1}{T}}$ with probability one. In general, if $A_T^{(1)} = \prod_{t=1}^T X_t^{(1)}$ and $A_T^{(2)} = \prod_{t=1}^T X_t^{(2)}$ are products of i.i.d. random variables, then $\mathbf{E}[\log X_1^{(1)}] > \mathbf{E}[\log X_1^{(2)}]$ if and only if $\lim_{t=1}^T A_T^{(1)}/A_T^{(2)} = \infty$ with probability one. These arguments imply that, in the case of a multiplicative reward model, it is reasonable to compare reward logarithms if we focus on events expected to happen with high probability.

8.3 Upper Bounds

8.3.1 Notation and Preliminary Consideration

Let us introduce some notations. For $S \subseteq [d]$, denote by Δ^S the set of portfolios whose supports are included in S, i.e., $\Delta^S = \left\{x \mid x_i \geq 0 \ (i \in [d]), \sum_{i=1}^d x_i \leq 1, \mathrm{supp}(x) \subseteq S\right\}$. Let (S^*, x^*) denote the optimal fixed strategy for T rounds, i.e., $(S^*, x^*) \in \arg\max_{S\in\mathcal{S}, x\in\Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)$. Let

x_t denote the output of an algorithm for the t-th round. Then the regret R_T of the algorithm can be expressed as

$$R_T = \max_{S \in \mathcal{S}, x \in \Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x) - \sum_{t=1}^T \log(1 + r_t^\top x_t) = \sum_{t=1}^T \log(1 + r_t^\top x^*) - \sum_{t=1}^T \log(1 + r_t^\top x_t).$$

The algorithms presented in this section maintain vectors $x_t^S \in \Delta^S$ for all $S \in \mathcal{S}$ at the beginning of the t-th round. They then choose S_t from $\mathcal{S}$, and output $(S_t, x_t^{S_t})$. Although other vectors x_t^S $(S \neq S_t)$ do not appear in the output, they are used to compute outputs in subsequent rounds.

In the computation of x_{t+1}^S, we refer to the following vectors g_t and matrices H_t^S:

$$g_t^S = \frac{r_t|_S}{1 + r_t^\top x_t^S}, \quad H_t^S = \frac{(1+C_1)^2}{(1+C_2)^2} g_t^S g_t^{S\top} = C_3 g_t^S g_t^{S\top}, \tag{8.2}$$

where $r_t|_S = [r'_{t1}, \ldots, r'_{td}]^\top$ is defined by $r'_{ti} = r_{ti}$ for $i \in S$ and $r'_{ti} = 0$ for $i \in [d] \setminus S$. These g_t^S and H_t^S have the following property which plays an important role in our analysis:

Lemma 8.3.1. *For any $x \in \Delta^S$, it holds that*

$$\log(1 + r_t^\top x) - \log(1 + r_t^\top x_t^S) \leq g_t^{S\top}(x - x_t^S) - \frac{1}{2}(x - x_t^S)^\top H_t^S (x - x_t^S). \tag{8.3}$$

Proof. Since it holds for all $x, x_0 \in [C_1, C_2]$ that $\frac{\mathrm{d}}{\mathrm{d}x}\log(1+x) = \frac{1}{1+x}$ and $\frac{\mathrm{d}^2}{\mathrm{d}x^2}\log(1+x) = -\frac{1}{(1+x)^2}$, we have $\log(1+x) - \log(1+x_0) \leq \frac{x-x_0}{1+x_0} - \frac{(x-x_0)^2}{2(1+C_2)^2} \leq \frac{x-x_0}{1+x_0} - \frac{C_3}{2}(\frac{x-x_0}{1+x_0})^2$, where we set $C_3 = \frac{(1+C_1)^2}{(1+C_2)^2}$. Hence, by substituting $x = r_t^\top x, x_0 = r_t^\top x_t^S$ for arbitrary $t \in [T]$, $S \in \mathcal{S}$ and $x \in \Delta^S$, we obtain $\log(1 + r_t^\top x) - \log(1 + r_t^\top x_t^S) \leq \frac{r_t^\top(x - x_t^S)}{1 + r_t^\top x_t^S} - \frac{C_3}{2}\left(\frac{r_t^\top(x - x_t^S)}{1 + r_t^\top x_t^S}\right)^2 \leq g_t^{S\top}(x - x_t^S) - \frac{C_3}{2}(g_t^{S\top}(x - x_t^S))^2 = g_t^{S\top}(x - x_t^S) - \frac{1}{2}(x - x_t^S)^\top H_t^S(x - x_t^S)$, □

8.3.2 Algorithm for the Full-Feedback Setting

We propose an algorithm for the full-feedback setting, created by combining the multiplicative weight update method (MWU) [11] and the follow-the-approximate-leader method (FTAL) [81]. More specifically, our proposed algorithm updates the probability of choosing a subset $S \in \mathcal{S}$ by MWU and updates the portfolio vector x_t^S by FTAL. The entire algorithm is summarized in Algorithm 12.

Our algorithm maintains *weight* $w_t^S \geq 0$ and a portfolio vector x_t^S for each subset $S \in \mathcal{S}$ at the begining of the t-th round, where w_1^S and x_1^S are initialized by $w_1^S = 1$ and $x_1^S = 0$ for all $S \in \mathcal{S}$. In each round t, a subset S_t is chosen with a probability proportional to w_t^S. Given the feedback r_t, the algorithm computes w_{t+1}^S and x_{t+1}^S. The weight w_{t+1}^S is obtained from w_t^S by multiplying $(1 + r_t^\top x_t^S)^\eta$, where $\eta > 0$ is a parameter we optimize later. The portfolio vector x_{t+1}^S is computed by FTAL as follows:

$$x_{t+1}^S \in \arg\max_{x \in \Delta^S} \left\{ \sum_{j=1}^t \left(g_j^{S\top}(x - x_j^S) - \frac{1}{2}(x - x_j^S)^\top H_j^S (x - x_j^S) \right) - \frac{\beta}{2}\|x\|_2^2 \right\}, \tag{8.4}$$

where β is a regularization parameter optimized later, and $\|\cdot\|$ stands for the ℓ_2 norm: $\|[x_1, \ldots, x_d]^\top\|_2^2 = \sum_{i=1}^d x_i^2$. Since (8.4) is a convex quadratic programming problem with

Algorithm 12 An algorithm for the full-feedback setting.

```
Require: The number $T$ of rounds. The number $d$ of assets. The set of available subsets
    $\mathcal{S} \subseteq 2^{[d]}$. Parameters $\eta > 0$ and $\beta > 0$.
1: Set $w_1 = (w_1^S)_{S \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S}}$ and $(x_1^S)_{S \in \mathcal{S}}$ by $w_1^S = 1$ and $x_1^S = 0$, respectively, for $S \in \mathcal{S}$.
2: for $t = 1, \ldots, T$ do
3:     Set $S_t$ by randomly choosing $S \in \mathcal{S}$ with a probability proportional to $w_t^S$, i.e., choose $S$
       with probability $w_t^S / \|w_t\|_1$.
4:     Output $S_t$ and $x_t = x_t^{S_t}$ and observe $r_{ti}$ for all $i \in [d]$.
5:     Update $w_t$; set $w_{t+1}$ by $w_{t+1}^S = w_t^S (1 + r_t^\top x_t^S)^\eta$ for $S \in \mathcal{S}$.
6:     Update $x_t^S$; set $x_{t+1}^S$ by equation (8.4) for $S \in \mathcal{S}$.
7: end for
```

linear constraints, x_{t+1}^S can be computed efficiently by, e.g., interior point methods [125]. Recently, Ye et al. [167] have proposed a more efficient algorithm for solving (8.4). For the special case of the single asset selection setting, i.e., if $\mathcal{S} = \mathcal{S}_1 = \{\{i\} \mid i \in [d]\}$, then $x_{t+1}^{\{i\}} = (0, \ldots, 0, x_{t+1,i}, 0, \ldots, 0)$ has a closed-form expression: $x_{t+1,i} = \pi_{[0,1]}\left(\frac{\sum_{j=1}^t g_{ji}}{\beta + C_3 \sum_{j=1}^t g_{ji}^2}\right)$, where $g_{ji} := \frac{r_{ji}}{1 + r_{ji} x_{ji}}$ and $\pi_{[0,1]}(\cdot)$ stands for a projection onto $[0,1]$ defined by $\pi_{[0,1]}(y) = 0$ for $y < 0$, $\pi_{[0,1]}(y) = y$ for $0 \leq y \leq 1$, and $\pi_{[0,1]}(y) = 1$ for $y > 1$.

Our algorithm achieves the regret described below for arbitrary inputs, where constants C_3, C_4, C_5 are given by $C_3 = \frac{(1+C_1)^2}{(1+C_2)^2}$, $C_4 = \log \frac{1+C_2}{1+C_1}$, and $C_5 = \frac{\max\{C_1^2, C_2^2\}}{(1+C_1)^2}$.

Theorem 8.3.1. *Algorithm 12 achieves the following regret upper bound if $\eta \leq 1/C_4$:*

$$\mathbf{E}[R_T] \leq \frac{\log |\mathcal{S}|}{\eta} + C_4^2 \eta T + \frac{1}{2}\beta + \frac{k}{C_3} \log\left(1 + \frac{C_3 C_5 T}{\beta}\right). \tag{8.5}$$

In particular, setting $\eta = \frac{1}{C_4} \min\left\{1, \sqrt{\frac{\log |\mathcal{S}|}{T}}\right\}$ and $\beta = 1$, we obtain

$$\mathbf{E}[R_T] = O\left(\sqrt{T \log |\mathcal{S}|} + k \log T + \log |\mathcal{S}|\right). \tag{8.6}$$

Proof. In the following, we denote $f_t(x) = \log(1 + r_t^\top x) - \log(1 + C_1)$. The regret R_T can be expressed as

$$R_T = \left(\sum_{t=1}^T f_t(x^*) - \sum_{t=1}^T f_t(x_t^{S^*})\right) + \left(\sum_{t=1}^T f_t(x_t^{S^*}) - \sum_{t=1}^T f_t(x_t^{S_t})\right). \tag{8.7}$$

Since S_t is chosen by MWU taking the input $(F_t^S)_{S \in \mathcal{S}} = (f_t(x_t^S))_{S \in \mathcal{S}}$, the second term on the right-hand side of (8.7) can be bounded as follows (see e.g., [11]):

$$\mathbf{E}\left[\sum_{t=1}^T f_t(x_t^{S^*}) - \sum_{t=1}^T f_t(x_t^{S_t})\right] \leq \frac{\log |\mathcal{S}|}{\eta} + C_4^2 \eta T. \tag{8.8}$$

Since $x_t^{S^*}$ is computed by FTAL, the first term on the right-hand side of (8.7) can be bounded as follows (see e.g., [81]):

$$\sum_{t=1}^T f_t(x^*) - \sum_{t=1}^T f_t(x_t^{S^*}) \leq \frac{\beta}{2} + \frac{|S^*|}{C_3} \log\left(1 + \frac{C_3 C_5 T}{\beta}\right). \tag{8.9}$$

Combining (8.7), (8.8) and (8.9), we obtain (8.5). □

Algorithm 13 An algorithm for the bandit-feedback setting.

```
Require: The number $T$ of rounds. The number $d$ of assets. The set of available subsets
    $\mathcal{S} \subseteq 2^{[d]}$. Parameters $\eta > 0$, $\gamma \in (0,1)$ and $\beta > 0$.
1: Set $w_1 = (w_1^S)_{S \in \mathcal{S}} \in \mathbb{R}^{\mathcal{S}}$ and $(x_1^S)_{S \in \mathcal{S}}$ by $w_1^S = 1$ and $x_1^S = 0$, respectively, for $S \in \mathcal{S}$.
2: for $t = 1, \ldots, T$ do
3:     Set the probability vector $p_t = (p_t^S)_{S \in \mathcal{S}} \in [0,1]^{\mathcal{S}}$ by $p_t^S = \frac{\gamma}{|\mathcal{S}|} + (1-\gamma)\frac{w_t^S}{\|w_t\|_1}$.
4:     Randomly choose $S_t \in \mathcal{S}$ on the basis of the probability vector $p_t$.
5:     Output $S_t$ and $x_t = x_t^{S_t}$, and observe $r_{ti}$ for $i \in S_t$.
6:     Update $w_t$; set $w_{t+1}^S$ by $w_{t+1}^{S_t} = w_{ti_t}\left(\frac{1+r_t^\top x_t}{1+C_1}\right)^{\eta/p_{ti_t}}$ and $w_{t+1}^S = w_t^S$ for $S \in \mathcal{S} \setminus \{S_t\}$.
7:     Update $x_t^S$; set $x_{t+1}^S$ by equation (8.11).
8: end for
```

Running time If (8.4) can be computed in $p(k)$-time, Algorithm 12 runs in $O(|\mathcal{S}|p(k))$-time per round. If $\mathcal{S}$ is an exponentially large set, e.g., if $\mathcal{S} = \{S \subseteq [d] \mid |S| = k\}$ and $k = \Theta(d)$, the computational time for $O(|\mathcal{S}|p(k))$ will be exponentially large w.r.t. d. This computational complexity is shown to be inevitable in Section 8.4.1. For the special case of the single asset selection setting, i.e., if $\mathcal{S} = \mathcal{S}_1 = \{\{i\} \mid i \in [d]\}$, Algorithm 12 runs in $O(d)$-time per round since each $x_t^{\{i\}}$ can be updated in constant time.

8.3.3 Algorithm for the Bandit-Feedback Setting

We construct an algorithm for the bandit-feedback setting by combining the Exp3 algorithm [16] for the multi-armed bandit problem and FTAL. Similarly to the process used in Algorithm 12, the algorithm updates the probability of choosing $S_t \in \mathcal{S}$ by the Exp3 algorithm (in place of MWU) and updates portfolios x_t^S by FTAL. The main difficulty comes from the fact that the learner cannot observe all the entries of $(r_{ti})_{i=1}^d$. Due to this limitation, we cannot always update x_t^S for all $S \in \mathcal{S}$. In order to deal with this problem, we construct unbiased estimators of g_t^S and H_t^S for each $S \in \mathcal{S}$ by

$$\hat{g}_t^{S_t} = \frac{g_t^{S_t}}{p_t^{S_t}}, \quad \hat{H}_t^{S_t} = \frac{H_t^{S_t}}{p_t^{S_t}}, \qquad \hat{g}_t^S = 0, \quad \hat{H}_t^S = O \quad (S \in \mathcal{S} \setminus \{S_t\}), \tag{8.10}$$

where p_t^S is the probability of choosing S in round t, which is computed by a procedure similar to that used in the Exp3 algorithm. Note that $\hat{g}_t^S$ and $\hat{H}_t^S$ can be calculated from the observed information alone. Using these unbiased estimators, we compute the portfolio vectors x_{t+1}^S by FTAL as follows:

$$x_{t+1}^S \in \arg\max_{x \in \Delta^S} \left\{ \sum_{j=1}^t \left(\hat{g}_j^{S\top}(x - x_j^S) - \frac{1}{2}(x - x_j^S)^\top \hat{H}_j^S (x - x_j^S) \right) - \frac{1}{2}\beta\|x\|_2^2 \right\}. \tag{8.11}$$

Note that $x_{t+1}^S = x_t^S$ for each $S \in \mathcal{S} \setminus \{S_t\}$ since $\hat{g}_t^S = 0$ and $\hat{H}_t^S = O$. Hence the convex quadratic programming problem (8.11) is solved only once in each round. The entire algorithm is summarized in Algorithm 13.

Theorem 8.3.2. *Algorithm 13 achieves the following regret upper bound if $\eta \leq \frac{\gamma}{C_4|\mathcal{S}|}$:*

$$\mathbf{E}[R_T] \leq \frac{\log|\mathcal{S}|}{\eta} + (C_4^2\eta|\mathcal{S}| + C_4\gamma)T + \frac{1}{2}\beta + \frac{k|\mathcal{S}|}{C_3\gamma}\log\left(1 + \frac{C_3C_5T}{\beta}\right). \tag{8.12}$$

Setting $\gamma = \min\left\{1, \sqrt{\frac{k|\mathcal{S}|\log(1+T)}{T}}\right\}$, $\eta = \frac{\gamma}{C_4|\mathcal{S}|}\min\left\{1, \sqrt{\frac{\log|S|}{k\log(1+T)}}\right\}$ *and* $\beta = C_3C_5$, *we obtain*

$$\mathbf{E}[R_T] = O\left(\sqrt{T|\mathcal{S}|k\log T} + |S|\sqrt{k\log|S|\log T} + |\mathcal{S}|k\right).$$

Proof. The regret R_T can be expressed as (8.7). Since S_t is chosen by Exp3 taking the input $(F_t^S)_{S\in\mathcal{S}} = (f_t(x_t^S))_{S\in\mathcal{S}}$, the second term on the right-hand side of (8.7) can be bounded as follows (see e.g., [16]):

$$\mathbf{E}\left[\sum_{t=1}^T f_t(x_t^{S^*}) - \sum_{t=1}^T f_t(x_t^{S_t})\right] \leq \frac{\log|\mathcal{S}|}{\eta} + (C_4^2\eta|\mathcal{S}| + C_4\gamma)T. \tag{8.13}$$

The first term on the right-hand side of (8.7) can be bounded as follows:

$$\begin{aligned}\mathbf{E}\left[\sum_{t=1}^T f_t(x^*) - \sum_{t=1}^T f_t(x_t^{S^*})\right] &\leq \mathbf{E}\left[\sum_{t=1}^T (g_t^{S^*\top}(x^* - x_t^{S^*}) - \frac{1}{2}(x^* - x_t^{S^*})^\top H_t^{S^*}(x^* - x_t^{S^*}))\right] \\ &= \mathbf{E}\left[\sum_{t=1}^T (\hat{g}_t^{S^*\top}(x^* - x_t^{S^*}) - \frac{1}{2}(x^* - x_t^{S^*})^\top \hat{H}_t^{S^*}(x^* - x_t^{S^*}))\right],\end{aligned} \tag{8.14}$$

where the inequality comes from (8.3) and the equality comes from the fact that $\hat{g}_t^S$ and $\hat{H}_t^S$ are unbiased estimators of g_t^S and H_t^S, respectively. Since $x_t^{S^*}$ is computed by FTAL as in (8.11), the right-hand side of can be bounded as follows (see e.g., [81]):

$$\begin{aligned}&\sum_{t=1}^T (\hat{g}_t^{S^*\top}(x^* - x_t^{S^*}) - \frac{1}{2}(x^* - x_t^{S^*})^\top \hat{H}_t^{S^*}(x^* - x_t^{S^*})) \\ &\leq \frac{\beta\|x^*\|_2^2}{2} + \sum_{t=1}^T \hat{g}_t^{S^*\top}(\beta I + \sum_{j=1}^t \hat{H}_j^{S^*})^{-1}\hat{g}_t^{S^*} \\ &\leq \frac{\beta}{2} + \frac{|\mathcal{S}|}{C_3\gamma}\sum_{t=1}^T C_3 p_t^{S^*}\hat{g}_t^{S^*\top}(\beta I + \sum_{j=1}^t C_3 p_j^{S^*}\hat{g}_j^{S^*}\hat{g}_j^{S^*\top})^{-1}\hat{g}_t^{S^*} \\ &\leq \frac{\beta}{2} + \frac{|\mathcal{S}|}{C_3\gamma}\log\frac{\det(\beta I + C_3\sum_{j=1}^T p_j^{S^*}\hat{g}_j^{S^*}\hat{g}_j^{S^*\top})}{\det\beta I},\end{aligned} \tag{8.15}$$

where the first and third inequalities come from the standard analysis of FTAL, and the second inequality holds since $p_t^S|\mathcal{S}|/\gamma \leq 1$ from the definition of p_t^S. Denote $M_T = C_3\sum_{j=1}^T p_j^{S^*}\hat{g}_j^{S^*}\hat{g}_j^{S^*\top}$. Since $\|g_t^{S^*}\|_0 \leq |S^*| \leq k$, the eigenvalues $\{\lambda_1, \ldots, \lambda_d\}$ of M_T include at least $d-k$ zero eigenvalues. From this and the fact that $\lambda_i \geq 0$ and $\sum_{j=1}^d \lambda_j = \mathrm{tr}(M_t)$, we have $\det(\beta I + M_T) = \prod_{j=1}^d(\beta + \lambda_i) \leq \beta^{d-k}(\beta + \frac{1}{k}\mathrm{tr}(M_T))^k$. This inequality and Jensen's inequality yield $\mathbf{E}[\log(\det(\beta I + M_T))] \leq (d-k)\log\beta + \mathbf{E}[k\log(\beta + \frac{1}{k}\mathrm{tr}(M_T))] \leq (d-k)\log\beta + k\log(\beta + \frac{1}{k}\mathbf{E}[\mathrm{tr}(M_T)])$. Since $\mathbf{E}[\mathrm{tr}(M_T)] = \sum_{t=1}^T \mathbf{E}[\mathrm{tr}(\hat{H}_t^{S^*})] = \sum_{t=1}^T \mathbf{E}[\mathrm{tr}(H_t^{S^*})] \leq TkC_3C_5$, we have $\mathbf{E}[\log(\det(\beta I + M_T))] \leq (d-k)\log\beta + k\log(\beta + TC_3C_5)$. Combining this with (8.14) and (8.15), we obtain (8.12). □

Running time Algorithm 13 runs in $O(p(k) + \log^2(|\mathcal{S}|))$-time per round, assuming that (8.11) can be computed in $p(k)$-time. In fact, from the definition (8.10) of $\hat{g}_t^S$ and $\hat{H}_t^S$, the update of x_t^S given by (8.11) is needed only for $S = S_t$. Furthermore, for $\mathcal{S} = \{S_1, S_2, \ldots, S_{|\mathcal{S}|}\}$, both updating w_t^S for some $S \in \mathcal{S}$ and computing the prefix sum $\sum_{j=1}^{i} w_t^{S_j}$ for some $i \in [|\mathcal{S}|]$ can be performed in $O(\log |\mathcal{S}|)$-time by using a Fenwick tree [55]. This implies that sampling S_t w.r.t. $p_t^S = \frac{\gamma}{|\mathcal{S}|} + \frac{w_t^S}{\|w_t\|^S}$ can be performed in $O(\log^2 |\mathcal{S}|)$-time.

8.4 Lower Bounds

In this section, we present lower bounds on regrets achievable by algorithms for the online portfolio selection problem. We focus on the case of $\mathcal{S} = \mathcal{S}_k = \{S \subseteq [d] \mid |S| = k\}$ throughout this section.

8.4.1 Computational Complexity

We show that, unless the complexity class **BPP** includes **NP**, there exists no algorithm for the online problem with a cardinality constraint such that its running time is polynomial both on d and T and its regret is bounded by a polynomial in d and sublinear in T. This fact is shown by presenting a reduction from the 3-dimensional matching problem (**3DM**). An instance U of **3DM** consists of 3-tuples $(x_1, y_1, z_1), \ldots, (x_d, y_d, z_d) \in [k] \times [k] \times [k]$. Two tuples, (x_i, y_i, z_i) and (x_j, y_j, z_j), are called disjoint if $x_i \neq x_j$, $y_i \neq y_j$, and $z_i \neq z_j$. The task of **3DM** is to determine whether or not there exist k pairwise-disjoint tuples; if they do exist, we write $U \in$ **3DM**.

From a **3DM** instance $U = \{(x_j, y_j, z_j)\}_{j=1}^{d}$, we construct an input sequence $(r_t)_{t=1,\ldots,T}$ of the online portfolio selection problem as follows. Let $A = (a_{ij}) \in \{0,1\}^{3k \times d}$ be a matrix such that $a_{ij} = 1$ if $i = x_j$ or $i = k + y_j$ or $i = 2k + z_j$, and $a_{ij} = 0$ otherwise. From A, we construct $B \in \mathbb{R}^{3k \times (d+1)}$ by $B = \frac{1}{3k}[A, -1_{3k}]$, where 1_{3k} is the all-one vector of dimension $3k$. Let $T \geq \max\{(4 \cdot 5184k^4)^2, (5184k^4 \cdot p_2(d))^{\frac{1}{\delta}}\}$ for an arbitrary polynomial p_2 and an arbitrary positive parameter δ. For each $t \in [T]$, take z_t from the uniform random distribution on $\{-1,1\}^{3k}$, independently. Then, r_t can be defined by $r_t = 1_{d+1} + B^\top z_t$ for each $t \in [T]$. Note that $r_t \in [0,2]^{(d+1)}$ holds for each $t \in [T]$.

We give the sequence $(r_t)_{t=1,\ldots,T}$ to an algorithm $\mathcal{A}$. Let $(x_t)_{t=1,\ldots,T}$ denote the sequence output by $\mathcal{A}$. We determine that $U \in$ **3DM** if $\sum_{t=1}^{T} \log(1 + r_t^\top x_t) \geq T(\log 2 - \frac{1}{5184k^4})$ holds, while otherwise we determine $U \notin$ **3DM** to hold. We can prove that this determination is correct with a probability of at least 2/3.

Theorem 8.4.1. *Let δ be an arbitrary positive number, and p_1 and p_2 be arbitrary polynomials. Assume that there exists a $p_1(d,T)$-time algorithm $\mathcal{A}$ for the full-feedback online portfolio selection problem with $\mathcal{S} = \mathcal{S}_{k+1}$ that achieves regret $R_T \leq p_2(d)T^{1-\delta}$ with a probability of at least 2/3. Then, given a* **3DM** *instance $U \subseteq [k] \times [k] \times [k]$, one can decide if $U \in$* **3DM** *with a probability of at least 2/3 in $p_1(|U|, \max\{k^8, (k^4 p_2(|U|))^{\frac{1}{\delta}}\})$-time.*

Proof. From a **3DM** instance $U = \{(x_j, y_j, z_j)\}_{j=1}^{d}$, we construct an input sequence $(r_t)_{t=1,\ldots,T}$ for algorithm $\mathcal{A}$ as follows. Let $A = (a_{ij}) \in \{0,1\}^{3k \times d}$ be a matrix such that $a_{ij} = 1$ if $i = x_j$ or $i = k + y_j$ or $i = 2k + z_j$, and $a_{ij} = 0$ otherwise. From A, we construct $B \in \mathbb{R}^{3k \times (d+1)}$ by $B = \frac{1}{3k}[A, -1_{3k}]$, where 1_{3k} is an all-one vector of dimension $3k$. Let $T \geq \max\{(4 \cdot 5184k^4)^2, (5184k^4 \cdot p_2(d))^{\frac{1}{\delta}}\}$. For each $t \in [T]$, take z_t from the uniform random distribution on $\{-1,1\}^{3k}$, independently. Then, r_t can be defined by $r_t = 1_{d+1} + B^\top z_t$ for each $t \in [T]$. Note that $r_t \in [0,2]^{(d+1)}$ holds for each $t \in [T]$.

We give the sequence $(r_t)_{t=1,\ldots,T}$ to $\mathcal{A}$. Let $(x_t)_{t=1,\ldots,T}$ denote the sequence output by $\mathcal{A}$. We determine that $U \in \mathbf{3DM}$ if $\sum_{t=1}^{T} \log(1 + r_t^\top x_t) \geq T(\log 2 - \frac{1}{5184k^4})$ holds, while otherwise we determine that $U \notin \mathbf{3DM}$ holds. Below, we prove that this determination is correct with a probability of at least 2/3.

Assume that $U \in \mathbf{3DM}$. Then, there exists $y^* \in \{0,1\}^d$ such that $\|y^*\|_0 = k$ and $Ay^* = 1_{3k}$, and there exists $y^* \in \{0,1\}^d$ such that $\|y^*\|_0 = \|y^*\|_1 = k$ and $Ay^* = 1_{3k}$. Define $x^* := \frac{1}{k+1}\begin{bmatrix} y^* \\ 1 \end{bmatrix}$. The vector x^* satisfies $x^* \in \Delta^S$ for some $S \in \mathcal{S}_{k+1}$. Moreover, it holds that $r_t^\top x^* = 1_{d+1}^\top x^* + z_t^\top B x^* = 1 + \frac{1}{3k(k+1)} z_t^\top (Ay^* - 1_{3k}) = 1$. Hence, we obtain

$$\max_{S \in \mathcal{S}_{k+1}, x \in \Delta^S} \sum_{t=1}^{T} \log(1 + r_t^\top x) \geq \sum_{t=1}^{T} \log(1 + r_t^\top x^*) = T \log 2.$$

From this inequality and $R_T \leq p_2(d) T^{1-\delta}$ (with a probability $\geq 2/3$), we obtain

$$\begin{aligned}\sum_{t=1}^{T} \log(1 + r_t^\top x_t) &\geq \max_{S \in \mathcal{S}_{k+1}, x \in \Delta^S} \sum_{t=1}^{T} \log(1 + r_t^\top x) - R_T \\ &\geq T \log 2 - p_2(d) T^{1-\delta} \geq T\left(\log 2 - \frac{1}{5184k^4}\right),\end{aligned}$$

where the last inequality comes from $T \geq (5184k^4 \cdot p_2(d))^{\frac{1}{\delta}}$. This inequality means that the decision is correct with a probability $\geq 2/3$ if $U \in \mathbf{3DM}$.

For the remainder of the proof, we assume that $U \notin \mathbf{3DM}$. This assumption implies that, for all $y \in \mathbb{R}^d_{\geq 0}$ satisfying $\|y\|_0 \leq k$, we have $\min_{1 \leq i \leq 3k} (Ay)_i = 0$. Moreover, since each column of A has at least one entry of value 1, we have $\max_{1 \leq i \leq 3k}(Ay)_i \geq \|y\|_\infty$ for all $y \in \mathbb{R}^d_{\geq 0}$.

We first prove that $\|Bx\|_2 \geq \frac{1}{12k^2}\|x\|_1$ holds for all $x \in \mathbb{R}^{d+1}_{\geq 0}$ satisfying $\|x\|_0 \leq k+1$. We consider the following two cases: the last entry of x is either positive or zero. The former case is when x is expressed as $x = \begin{bmatrix} y \\ y_0 \end{bmatrix}$ with $y \in \mathbb{R}^d_{\geq 0}$, $\|y\|_0 \leq k$ and $y_0 > 0$. In this case, we have

$$\begin{aligned}\|Bx\|_\infty &\geq \frac{1}{3k} \max\left\{ |\min_{1 \leq i \leq 3k} (Ay)_i - y_0|, |\max_{1 \leq i \leq 3k} (Ay)_i - y_0| \right\} \\ &= \frac{1}{3k} \max\left\{ |y_0|, |\max_{1 \leq i \leq 3k}(Ay)_i - y_0| \right\} \\ &\geq \frac{1}{3k} \max\left\{ |y_0|, \frac{1}{2}|\max_{1 \leq i \leq 3k}(Ay)_i| \right\} \geq \frac{1}{3k} \max\left\{ |y_0|, \frac{1}{2}\|y\|_\infty \right\} \geq \frac{1}{6k}\|x\|_\infty,\end{aligned}$$

where the second inequality comes from the fact that arbitrary y_0 satisfies $\max\{|y_0|, |a - y_0|\} \geq |a|/2$. In the latter case, namely, when $x = \begin{bmatrix} y \\ 0 \end{bmatrix}$ with some $x \in \mathbb{R}^d_{\geq 0}$ such that $\|y\|_0 \leq k+1$, we have $\|Bx\|_\infty \geq \frac{1}{3k}\|y\|_\infty = \frac{1}{3k}\|x\|_\infty$. Accordingly, in both of these cases, we have $\|Bx\|_\infty \geq \frac{1}{6k}\|x\|_\infty$, and hence, we have $\|Bx\|_2 \geq \|Bx\|_\infty \geq \frac{1}{6k}\|x\|_\infty \geq \frac{1}{6k(k+1)}\|x\|_1 \geq \frac{1}{12k^2}\|x\|_1$.

Then, since $\log(1+y) \leq \log 2 + \frac{1}{2}(y-1) - \frac{1}{18}(y-1)^2$ for $y \in [0,2]$, and since z_t are statistically independent of x_t and $\mathbf{E}[z_t] = 0, \mathbf{E}[z_t z_t^\top] = I$, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{z_t,x_t}[\log(1 + r_t^\top x_t)] \\
&\leq \log 2 + \mathop{\mathbf{E}}_{z_t,x_t}\left[\frac{1}{2}(\|x_t\|_1 + z_t^\top B x_t - 1) - \frac{1}{18}(\|x_t\|_1 + z_t^\top B x_t - 1)^2\right] \\
&= \log 2 - \frac{1}{2} + \mathop{\mathbf{E}}_{x_t,z_t}\left[\frac{1}{2}\|x_t\|_1 - \frac{1}{18}\left(x_t^\top B^\top z_t z_t^\top B x_t - 2z_t^\top B x_t(\|x_t\|_1 - 1) + (\|x_t\|_1 - 1)^2\right)\right] \\
&\leq \log 2 - \frac{1}{2} + \mathop{\mathbf{E}}_{x_t}\left[\frac{1}{2}\|x_t\|_1 - \frac{1}{18}\|Bx_t\|_2^2\right] \\
&\leq \log 2 - \frac{1}{2} + \mathop{\mathbf{E}}_{x_t}\left[\frac{1}{2}\|x_t\|_1 - \frac{1}{18(12k^2)^2}\|x_t\|_1^2\right] \leq \log 2 - \frac{1}{2592k^4}.
\end{aligned}$$

This inequality means that the stochastic process $\{X_t\}_{t=1}^T$ defined by $X_t = \sum_{j=1}^t \log(1 + r_t x_t) - t(\log 2 - \frac{1}{2592k^4})$ is a sub-martingale. From the definition, $\{X_t\}_{t=1}^T$ satisfies $|X_t - X_{t+1}| < \log 3$ for all t. Hence, from the Azuma-Hoeffding inequality [9], X_T is bounded as $X_T < 4\sqrt{T}$ with a probability of at least $2/3$. Consequently, we have

$$\sum_{t=1}^T \log(1 + r_t^\top x_t) < T(\log 2 - \frac{1}{2592k^4}) + 4\sqrt{T} \leq T\left(\log 2 - \frac{1}{5184k^4}\right),$$

where the last inequality comes from $T \geq (4 \cdot 5184k^4)^2$. This means that the decision is correct with a probability of at least $2/3$. $\square$

Corollary 8.4.1. *Under the assumption of* $\mathbf{NP} \not\subseteq \mathbf{BPP}$, *if an algorithm achieves* $O(p(d,k)T^{1-\delta})$ *regret for arbitrary* d *and arbitrary* k, *the algorithm will not run in polynomial time, i.e., the running time will be larger than any polynomial for some* d *and some* k.

Note that the computational lower bounds described in Theorem 8.4.1 and Corollary 8.4.1 are also valid for the bandit-feedback setting, since algorithms for the bandit-feedback settings can be used for the full-feedback setting.

8.4.2 Regret Lower Bound for the Full-Feedback Setting

We show here that, for the full-feedback setting of the online portfolio selection problem with $\mathcal{S} = \mathcal{S}_k$, every algorithm (including exponential-time algorithms) suffers from regret of $\Omega\left(\sqrt{T \log \frac{d}{k}}\right)$ in the worst case. We can show this by analyzing the behavior of an algorithm for a certain random input. In the analysis, we use the fact that the following two inequalities hold when r_t follows the discrete uniform distribution on $\{0,1\}^d$ independently:

$$\mathop{\mathbf{E}}_{r_t,x_t}\left[\sum_{t=1}^T \log(1 + r_t^\top x_t)\right] \leq T \mathop{\mathbf{E}}_{X}\left[\log\left(1 + \frac{1}{k}X\right)\right],$$

$$\mathop{\mathbf{E}}_{r_t,x_t}\left[\max_{S \in \mathcal{S}_k, x \in \Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)\right] \geq T \cdot \mathop{\mathbf{E}}_{X}\left[\log\left(1 + \frac{1}{k}X\right)\right] + \Omega\left(\sqrt{T \log \frac{d}{k}}\right),$$

where X is a binomial random variable following $B(k, 1/2)$.

Theorem 8.4.2. *Let $d \geq 17k$, and consider the online portfolio selection problem with d assets and available combinations $\mathcal{S} = \mathcal{S}_k$. There is a probability distribution of input sequences $\{r_t\}_{t=1}^T$ such that the regret of any algorithm for the full-feedback setting is bounded as $\mathbf{E}[R_T] = \Omega\left(\sqrt{T \log \frac{d}{k}}\right)$, where the expectation is with respect to the randomness of both r and the algorithm.*

To prove Theorem 8.4.2, let us start with the following lemma.

Lemma 8.4.1. *If $0 \leq p_1 \leq p_2 \leq 1$ and random variables X_1, X_2 follow the binomial random distributions $B(k, p_1), B(k, p_2)$, respectively, then we have*

$$\mathop{\mathbf{E}}_{X_2 \sim B(k,p_2)}\left[\log\left(1 + \frac{1}{k}X_2\right)\right] - \mathop{\mathbf{E}}_{X_1 \sim B(k,p_1)}\left[\log\left(1 + \frac{1}{k}X_1\right)\right] \geq \frac{p_2 - p_1}{2} \tag{8.16}$$

Proof. Define $Y_1 = k - X_1$ and $Y_2 = k - X_2$. Then we have $Y_1 \sim B(k, 1-p_1)$ and $Y_2 \sim B(k, 1-p_2)$. From the Maclaurin series of $\log(2-x) = \log 2 - \frac{1}{2}x - \frac{1}{2 \cdot 2^2}x^2 - \cdots = \log 2 - \sum_{n=1}^{\infty} \frac{x^n}{n 2^n}$, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{X_2 \sim B(k,p_2)}\left[\log\left(1 + \frac{1}{k}X_2\right)\right] - \mathop{\mathbf{E}}_{X_1 \sim B(k,p_1)}\left[\log\left(1 + \frac{1}{k}X_1\right)\right] \\
&= \mathop{\mathbf{E}}_{Y_2 \sim B(k,1-p_2)}\left[\log\left(2 - \frac{1}{k}Y_2\right)\right] - \mathop{\mathbf{E}}_{Y_1 \sim B(k,1-p_1)}\left[\log\left(2 - \frac{1}{k}Y_1\right)\right] \\
&= \sum_{n=1}^{\infty} \frac{1}{n(2k)^n}\left(\mathop{\mathbf{E}}_{Y_1 \sim B(k,1-p_1)}[Y_1^n] - \mathop{\mathbf{E}}_{Y_2 \sim B(k,1-p_2)}[Y_2^n]\right) \\
&\geq \frac{1}{2k}\left(\mathop{\mathbf{E}}_{Y_1 \sim B(k,1-p_1)}[Y_1] - \mathop{\mathbf{E}}_{Y_2 \sim B(k,1-p_2)}[Y_2]\right) = \frac{p_2 - p_1}{2}.
\end{aligned}$$

□

We are now ready to prove Theorem 8.4.2.

Proof of Theorem 8.4.2. We construct an input sequence $\{r_t\}_{t=1,2,\ldots}$ so that entries r_{ti} follow a uniform random distribution over $\{0, 1\}$ independently. We can show that

$$\mathop{\mathbf{E}}_{r,x}[R_T(r)] = \mathop{\mathbf{E}}_{r,x}\left[\max_{S \in \mathcal{S}_k, x \in \Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x) - \sum_{t=1}^T \log(1 + r_t^\top x_t)\right] = \Omega\left(\sqrt{T \log \frac{d}{k}}\right) \tag{8.17}$$

for all algorithms, by means of considering the following two inequalities:

$$\mathop{\mathbf{E}}_{r_t, x_t}[\log(1 + r_t^\top x_t)] \leq \mathop{\mathbf{E}}_{X_1}\left[\log\left(1 + \frac{1}{k}X_1\right)\right], \tag{8.18}$$

$$\mathop{\mathbf{E}}_{r_t, x_t}\left[\max_{S \in \mathcal{S}_k, x \in \Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)\right] \geq T \cdot \mathop{\mathbf{E}}_{X_1}\left[\log\left(1 + \frac{1}{k}X_1\right)\right] + \Omega\left(\sqrt{T \log \frac{d}{k}}\right), \tag{8.19}$$

where X_1 is a binomial random variable following $B(k, 1/2)$.

First, let us prove the inequality (8.18). Consider a function $x \mapsto \mathop{\mathbf{E}}_{r_t}[\log(1 + r_t^\top x)]$, and suppose $S \in \mathcal{S}_k$. We can then confirm that this is a concave function and that, for the optimization problem $\arg\max_{x\in\Delta^S} \mathop{\mathbf{E}}_{r_t}[\log(1+r_t^\top x)]$, the vector $\frac{1}{k}1_S$ is the unique point satisfying KKT conditions, where 1_S stands for the indicator vector of S, i.e., $1_S = [\chi_1, \ldots, \chi_d]^\top$ where $\chi_i = 1$ if $i \in S$ and $\chi_i = 0$ if $i \in [d] \setminus S$. Consequently, we have $\max_{x\in\Delta^S} \log(1 + r_t^\top x) = \mathop{\mathbf{E}}_{r_t}[\log(1 + \frac{1}{k}1_S^\top r_t)] = \mathop{\mathbf{E}}_{X_1}[\log(1 + \frac{1}{k}X_1)]$ since $1_S^\top r_t$ follows the binomial distribution $B(k, 1/2)$. Since $x_t \in \Delta^{S_t}$ for some $S_t \in \mathcal{S}_k$ and S_t, x_t are stochastically independent of r_t, we obtain $\mathop{\mathbf{E}}_{r_t,x_t}\left[\log(1 + r_t^\top x_t)\right] \le \mathop{\mathbf{E}}_{r_t}\left[\log\left(1 + \frac{1}{k}r_t^\top 1_{S_t}\right)\right] = \mathop{\mathbf{E}}_{X_1}\left[\log\left(1 + \frac{1}{k}X_1\right)\right]$.

Next, let us prove the inequality (8.19). For each $i \in [d]$, define $r_i := \sum_{t=1}^T r_{ti}$. Since r_{ti} follows a Bernoulli distribution with parameter $1/2$ independently, r_i follows the binomial distribution $B(T, 1/2)$. Let $\sigma : [d-k] \to [d-k]$ be a permutation such that $r_{\sigma(1)} \ge r_{\sigma(2)} \ge \cdots \ge r_{\sigma(d-k)}$. Since the posterior random distribution of r_{ti} given r_i is the Bernoulli distribution of parameter r_i/T, for $x_2 = \frac{1}{k}1_{\{\sigma(1),\sigma(2),\ldots,\sigma(k)\}}$ and for arbitrary constant $s \ge T/2$, we have

$$\begin{aligned}\mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}\ge s\}} \cdot \sum_{t=1}^T \log(1 + r_t^\top x_2)\right] &\ge \mathop{\mathbf{E}}_{X_2\sim B(k,\frac{s}{T})}\left[\chi_{\{r_{\sigma(k)}\ge s\}} \cdot \sum_{t=1}^T \log(1 + \frac{1}{k}X_2)\right] \\ &= T \cdot \mathrm{Prob}[r_{\sigma(k)} \ge s] \cdot \mathop{\mathbf{E}}_{X_2\sim B(k,\frac{s}{T})}\left[\log(1 + \frac{1}{k}X_2)\right],\end{aligned}$$

where χ_A stands for the indicator function for arbitrary events A. Moreover, since $r_{d-k+1}, \ldots, r_d$ are independent of $r_{\sigma(k)}$, for $x_1 = \frac{1}{k}1_{\{d-k+1,\ldots,d\}}$, we have

$$\mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}< s\}} \sum_{t=1}^T \log(1 + r_t^\top x_1)\right] = T \cdot \mathrm{Prob}[r_{\sigma(k)} < s] \cdot \mathop{\mathbf{E}}_{X_1\sim B(k,\frac{1}{2})}\left[\log(1 + \frac{1}{k}X_1)\right].$$

Hence, we obtain

$$\begin{aligned}&\mathop{\mathbf{E}}_{r}\left[\max_{S\in\mathcal{S}_k, x\in\Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)\right] \\ &= \mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}\ge s\}} \max_{S\in\mathcal{S}_k, x\in\Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)\right] + \mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}< s\}} \max_{S\in\mathcal{S}_k, x\in\Delta^S} \sum_{t=1}^T \log(1 + r_t^\top x)\right] \\ &\ge \mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}\ge s\}} \sum_{t=1}^T \log(1 + r_t^\top x_2)\right] + \mathop{\mathbf{E}}_{r}\left[\chi_{\{r_{\sigma(k)}< s\}} \sum_{t=1}^T \log(1 + r_t^\top x_1)\right] \\ &\ge T \cdot \mathrm{Prob}[r_{\sigma(k)} \ge s] \cdot \mathop{\mathbf{E}}_{X_2}\left[\log(1 + \frac{1}{k}X_2)\right] + T \cdot \mathrm{Prob}[r_{\sigma(k)} \le s] \cdot \mathop{\mathbf{E}}_{X_1}\left[\sum_{t=1}^T \log(1 + \frac{1}{k}X_1)\right] \\ &\ge T \cdot \mathop{\mathbf{E}}_{X_1}\left[\log(1 + \frac{1}{k}X_1)\right] + T \cdot \mathrm{Prob}[r_{\sigma(k)} \ge s] \cdot \frac{1}{2} \cdot \left(\frac{s}{T} - \frac{1}{2}\right),\end{aligned}$$

where $X_1 \sim B(k, \frac{1}{2})$, $X_2 \sim B(k, \frac{s}{T})$ and the last inequality comes from Lemma 8.4.1. We now can show that we have $\mathrm{Prob}[r_{\sigma(k)} \ge s] = \Omega(1)$ for $s = \frac{T}{2} + \Omega(\sqrt{T \log \frac{d}{k}})$, which proves (8.19). Let $F : \mathbb{R} \to [0, 1]$ denote the cumulative distribution function of $B(T, 1/2)$, i.e., $F(x) = \mathrm{Prob}[r_i \le x]$. From a standard concentration lemma of a binomial distribution (see, e.g., Proposition 7.3.2

in [129]), we have $F(\frac{T}{2}+t) \le 1 - \frac{1}{15}\exp\left(-16\frac{t^2}{T}\right)$. Hence, setting $t = \frac{1}{4}\sqrt{T\log\frac{d-k}{15k}}$, we obtain

$$\begin{aligned}&\mathrm{Prob}\left[r_{\sigma(k)} \ge \frac{T}{2}+t\right] = \mathrm{Prob}\left[F(r_{\sigma(k)}) \ge F\left(\frac{T}{2}+t\right)\right]\\ &\ge \mathrm{Prob}\left[F(r_{\sigma(k)}) \ge 1 - \frac{1}{15}\exp\left(-16\frac{t^2}{T}\right)\right] = \mathrm{Prob}\left[F(r_{\sigma(k)}) \ge 1 - \frac{k}{d-k}\right]\end{aligned}$$

Since $F(r_i)$ follows the uniform distribution on $[0,1]$ independently, $F(r_{\sigma(k)})$ follows the probability distribution of the order statistic sampled from the standard uniform distribution, which is the beta distribution $\mathrm{Beta}(d-2k+1,k)$ (see, e.g., [67]). This means that $\mathrm{Prob}[1-F(r_{\sigma(k)}) \le \frac{k}{d-k}] \ge 1/2$. Combining the above two inequalities, for $s = \frac{T}{2} + \frac{1}{4}\sqrt{T\log\frac{d-k}{15k}}$, we have

$$\begin{aligned}&\mathop{\mathbf{E}}_{r}\left[\max_{S\in\mathcal{S}_k, x\in\Delta^S}\sum_{t=1}^{T}\log(1+r_t^\top x)\right]\\ &\ge T\cdot\mathop{\mathbf{E}}_{X_1}\left[\log(1+\frac{1}{k}X_1)\right] + \frac{1}{2}\mathrm{Prob}[r_{\sigma(k)} \ge s]\cdot\left(s-\frac{T}{2}\right)\\ &\ge T\cdot\mathop{\mathbf{E}}_{X_1}\left[\log(1+\frac{1}{k}X_1)\right] + \frac{1}{16}\left(\sqrt{T\log\frac{d-k}{15k}}\right)\\ &\ge T\cdot\mathop{\mathbf{E}}_{X_1}\left[\log(1+\frac{1}{k}X_1)\right] + \Omega\left(\sqrt{T\log\frac{d}{k}}\right),\end{aligned}$$

where the last inequality comes from $d \ge 17k$. Consequently, we obtain (8.19). From (8.18) and (8.19) we have (8.17). □

8.4.3 Regret Lower Bound for the Bandit-Feedback Setting

In this subsection, we consider the bandit-feedback setting of the online portfolio selection problem with $\mathcal{S} = \mathcal{S}_k$. We show that every algorithm (including exponential-time algorithms) for this setting suffers from regret of $\Omega\left(\sqrt{T(\frac{d}{Ck^3})^k}\right)$ when the input sequence is defined as follows. Let $S^* \in \mathcal{S}_k$. We define a random distribution D_{S^*} on $\{-1,1\}^d$ so that a random vector $z = [z_1,\ldots,z_d]^\top$ following this distribution satisfies

$$\prod_{i\in S^*} z_i = \begin{cases} 1 & \text{w.p. } 1/2-\varepsilon \\ -1 & \text{w.p. } 1/2+\varepsilon\end{cases}, \quad \prod_{i\in S} z_i = \begin{cases} 1 & \text{w.p. } 1/2 \\ -1 & \text{w.p. } 1/2\end{cases} \quad (S \in 2^{[d]}\setminus\{\emptyset, S^*\}).$$

Such a distribution can be constructed as follows: fix an index $i^* \in S^*$, let $z_i = \begin{cases} 1 & \text{w.p. } 1/2 \\ -1 & \text{w.p. } 1/2\end{cases}$ for each $i \in [d]\setminus\{i^*\}$, and let $z_0 = \begin{cases} 1 & \text{w.p. } 1/2-\varepsilon \\ -1 & \text{w.p. } 1/2+\varepsilon\end{cases}$ independently. Define $z_{i^*} = z_0\prod_{i\in S^*\setminus\{i^*\}} z_i$. Then $z = [z_1,\ldots,z_d]^\top \sim D_{S^*}$. The price relative vector r_t in the t-th round can be defined by $r_t = 1_d - z_t$, where $z_t \sim D_S^*$ independently for $t\in[T]$. We can show that $r_t|_S$ follows a uniform distribution for any $S \in \mathcal{S}_k\setminus\{S^*\}$ and only $r_t|_{S^*}$ follows a slightly different distribution. Because of this, it is difficult for algorithms to distinguish S^* from others, which makes their regrets large.

Theorem 8.4.3. *Let $d \geq k-1$, and consider the online portfolio selection problem with d assets and available combinations $\mathcal{S} = \mathcal{S}_k$. There is a probability distribution of input sequences $\{r_t\}_{t=1}^T$ such that the regret of any algorithm for the bandit-feedback setting is bounded as $\mathbf{E}[R_T] = \Omega\left(\min\left\{\frac{T}{k(Ck)^k}, \sqrt{T(\frac{d}{Ck^3})^k}\right\}\right)$, where the expectation is with respect to the randomness of both r and the algorithm, and C is a constant depending on C_1 and C_2.*

We introduce the following lemma to prove Theorem 8.4.3.

Lemma 8.4.2. *For arbitrary $\varepsilon \in [0, 1/2]$, let $z_0, z_1, \ldots, z_k \in \{-1, 1\}$ be independent random variables such that $z_0 = \begin{cases} 1 & \text{w.p. } 1/2 - \varepsilon \\ -1 & \text{w.p. } 1/2 + \varepsilon \end{cases}$ and $z_i = \begin{cases} 1 & \text{w.p. } 1/2 \\ -1 & \text{w.p. } 1/2 \end{cases}$ for $i = 1, \ldots, d$. Set $X_1 = \sum_{i=1}^k z_i$ and $X_2 = \sum_{i=1}^{k-1} z_i + z_0 \prod_{j=1}^{k-1} z_j$. We then have*

$$\mathbf{E}\left[\log\left(2 - \frac{1}{k}X_2\right)\right] - \mathbf{E}\left[\log\left(2 - \frac{1}{k}X_1\right)\right] \geq \frac{2\varepsilon}{k(2k)^k}. \tag{8.20}$$

Proof. Denote $w = z_0 \prod_{j=1}^{k-1} z_j$. Let $n_1, n_2, \ldots, n_k$ be arbitrary non-negative integers. Set m_i to be n_i modulo 2, i.e., $m_i = 0$ if n_i is even and $m_i = 1$ if n_i is odd, for $i = 1, \ldots, k$. We then have

$$\mathbf{E}[z_1^{n_1} z_2^{n_2} \cdots z_{k-1}^{n_{k-1}} z_k^{n_k}] = \mathbf{E}[z_1^{m_1} z_2^{m_2} \cdots z_{k-1}^{m_{k-1}} z_k^{m_k}] = \begin{cases} 1 & \text{if } m_1 = \cdots = m_k = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{E}[z_1^{n_1} z_2^{n_2} \cdots z_{k-1}^{n_{k-1}} w^{n_k}] = \mathbf{E}[z_1^{m_1} z_2^{m_2} \cdots z_{k-1}^{m_{k-1}} w^{m_k}] = \begin{cases} 1 & \text{if } m_1 = \cdots = m_k = 0 \\ -2\varepsilon & \text{if } m_1 = \cdots = m_k = 1 \\ 0 & \text{otherwise} \end{cases},$$

which means $\mathbf{E}[z_1^{n_1} \cdots z_{k-1}^{n_{k-1}} z_k^{n_k}] \geq \mathbf{E}[z_1^{n_1} \cdots z_{k-1}^{n_{k-1}} w^{n_k}]$. Hence, $X_1 = \sum_{i=1}^k z_i$ and $X_2 = \sum_{i=1}^{k-1} z_i + w$ satisfies $\mathbf{E}[X_1^n] \geq \mathbf{E}[X_2^n]$ for all non-negative integers n and $\mathbf{E}[X_1^k] - \mathbf{E}[X_2^k] = 2\varepsilon$. From the Maclaurin series of $\log(2 - x) = \log 2 - \sum_{n=1}^\infty \frac{x^n}{n 2^n}$, we have

$$\begin{aligned}\mathbf{E}\left[\log\left(2 - \frac{1}{k}X_2\right)\right] - \mathbf{E}\left[\log\left(2 - \frac{1}{k}X_1\right)\right] &= \sum_{n=1}^\infty \frac{1}{n(2k)^n}(\mathbf{E}[X_1^n] - \mathbf{E}[X_2^n]) \\ &\geq \frac{1}{k(2k)^k}(\mathbf{E}[X_1^k] - \mathbf{E}[X_2^k]) = \frac{2\varepsilon}{k(2k)^k}.\end{aligned}$$

□

Proof of Theorem 8.4.3

Proof. For each $S^* \in \mathcal{S}_k$, we define a random distribution D_{S^*} on $\{-1, 1\}^d$ so that $z = [z_1, \ldots, z_d]^\top \sim D_{S^*}$ satisfies

$$\prod_{i \in S^*} z_i = \begin{cases} 1 & \text{w. p. } 1/2 - \varepsilon \\ -1 & \text{w. p. } 1/2 + \varepsilon \end{cases}, \quad \prod_{i \in S} z_i = \begin{cases} 1 & \text{w. p. } 1/2 \\ -1 & \text{w. p. } 1/2 \end{cases} \quad (S \in 2^{[d]} \setminus \{\emptyset, S^*\}). \tag{8.21}$$

Such a distribution can be constructed as follows: fix an index $i^* \in S^*$ and, for $i \in [d] \setminus \{i^*\}$, let $z_i = \begin{cases} 1 & \text{w. p. } 1/2 \\ -1 & \text{w. p. } 1/2 \end{cases}$ and $z_0 = \begin{cases} 1 & \text{w. p. } 1/2 - \varepsilon \\ -1 & \text{w. p. } 1/2 + \varepsilon \end{cases}$ independently. Define $z_{i^*} =$

$z_0 \prod_{i \in S^* \setminus \{i^*\}} z_i$. Suppose that the input sequence r_t is given by $r_t = 1 - z_t$, where $z_t \sim D^*_S$ independently for $t = 1, 2, \ldots, T$. If z follows D^*_S, for any $S \in \mathcal{S}_k \setminus \{\emptyset S^*\}$, $z|_S$ follows the uniform distribution on $\{-1, 1\}^S$, and hence, we have $\max_{x \in \Delta^{S'}} \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + r_t^\top x\right)\right] = \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + \frac{1}{k} r_t^\top 1_S\right)\right]$. From Lemma 8.4.2, for $S \in \mathcal{S}_k \setminus \{\emptyset S^*\}$, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + \frac{1}{k} r_t^\top 1_{S^*}\right)\right] - \max_{x \in \Delta^S} \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + r_t^\top x\right)\right] \\
&\geq \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + \frac{1}{k} r_t^\top 1_{S^*}\right)\right] - \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + \frac{1}{k} r_t^\top 1_{S}\right)\right] \geq \frac{2\varepsilon}{k(2k)^k}. \qquad (8.22)
\end{aligned}$$

Since any randomized algorithm is equivalent to an a priori random choice from the set of all deterministic strategies, and since the input defined above is oblivious to the output of the algorithm, it suffices to prove a lower bound on the expected regret of any *deterministic* algorithm (this is not crucial for the proof but simplifies the notation). We consider an arbitrary deterministic algorithm and let $\{(S_t, x_t)\}_{t=1}^T$ denote the output for the random input sequence $\{r_t\}_{t=1}^T$ given by $r_t = 1 + z_t$ and $z_t \sim D_{S^*}$. Let N_S be a random variable denoting the number of $t \in [T]$ such that $S_t = S$, i.e., $N_S = |\{t \in [T] \mid S_t = S\}|$. From the equation (8.22), we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{z_t \sim D_{S^*}} [R_T] &\geq \sum_{t=1}^T \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + \frac{1}{k} r_t^\top 1_S\right)\right] - \sum_{t=1}^T \mathop{\mathbf{E}}_{S_t} \left[\max_{x \in \Delta^{S_t}} \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} \left[\log\left(1 + r_t^\top x\right)\right]\right] \\
&\geq \left(T - \mathop{\mathbf{E}}_{z_t \sim D_{S^*}} [N_{S^*}]\right) \frac{2\varepsilon}{k(2k)^k}. \qquad (8.23)
\end{aligned}$$

Let us evalute $\mathop{\mathbf{E}}_{z_t \sim D_{S^*}} [N_{S^*}]$. Define D_0 to be the uniform probabilistic distribution on $\{-1, 1\}^d$. Then, for all $S \in \mathcal{S}_k \setminus \{\emptyset, S^*\}$, we have $D_{S^*}|_S = D_0|_S$, i.e., if $z \sim D_{S^*}$ and $z' \sim D_0$, then $z|_S$ and $z'|_S$ follows the same distribution (a uniform distribution on $\{-1, 1\}^S$). Hence, in the same way as in Lemma A.1. of [16], we can show that

$$\mathop{\mathbf{E}}_{z_t \sim D_{S^*}} [N_{S^*}] - \mathop{\mathbf{E}}_{z_t \sim D_0} [N_{S^*}] \leq \frac{T}{\sqrt{2}} \sqrt{\mathop{\mathbf{E}}_{z_t \sim D_0} [N_{S^*}] \cdot \mathrm{KL}\Big(D_0|_{S^*} \,\Big|\Big|\, D_{S^*}|_{S^*}\Big)} \qquad (8.24)$$

where $\mathrm{KL}(P||Q) = \mathop{\mathbf{E}}_P(\log \frac{\mathrm{d}P}{\mathrm{d}Q})$ is the Kullback-Leibler divergence. The chain rule for relative entropy (see, e.g., Theorem 2.5.3 of [48]) gives, for $S^* = \{i_1, \ldots i_k\}$,

$$\begin{aligned}
\mathrm{KL}\Big(D_0|_{S^*} \,\Big|\Big|\, D_{S^*}|_{S^*}\Big) &= \mathrm{KL}(\mathop{\mathrm{Prob}}_{z \sim D_0}[(z_i)_{i \in S^*}] \mid\mid \mathop{\mathrm{Prob}}_{z \sim D_{S^*}}[(z_i)_{i \in S^*}]) \\
&= \sum_{j=1}^k \mathop{\mathbf{E}}_{(z_{i_s})_{s<j}} \left[\mathrm{KL}(\mathop{\mathrm{Prob}}_{z \sim D_0}[z_{i_j} \mid (z_{i_s})_{s<j}] \mid\mid \mathop{\mathrm{Prob}}_{z \sim D_{S^*}}[z_{i_j} \mid (z_{i_s})_{s<j}])\right] \\
&= \mathop{\mathbf{E}}_{(z_{i_s})_{s<k}} \left[\mathrm{KL}(\mathop{\mathrm{Prob}}_{z \sim D_0}[z_{i_k} \mid (z_{i_s})_{s<k}] \mid\mid \mathop{\mathrm{Prob}}_{z \sim D_{S^*}}[z_{i_k} \mid (z_{i_s})_{s<k}])\right] \\
&= -\frac{1}{2} \log(1 - 4\varepsilon^2). \qquad (8.25)
\end{aligned}$$

In the above equations, the third equality holds because $\mathop{\mathrm{Prob}}_{z \sim D_0}[z_{i_j} \mid (z_{i_s})_{s<j}]$ and $\mathop{\mathrm{Prob}}_{z \sim D_{S^*}}[z_{i_j} \mid (z_{i_s})_{s<j}]$ are equal to the Bernoulli distribution of parameter 1/2 for $j < k$. The last equality holds because $\mathop{\mathrm{Prob}}_{z \sim D_0}[z_{i_k} \mid (z_{i_s})_{s<k}]$ follows Bernoulli distribution of parameter 1/2 and $\mathop{\mathrm{Prob}}_{z \sim D_{S^*}}[z_{i_k} \mid$

$(z_{i_s})_{s<k}]$ follows Bernoulli distribution of parameter $1/2+\varepsilon$ or $1/2-\varepsilon$. Combining (8.23), (8.24) and (8.25), we have

$$\mathop{\mathbf{E}}_{z_t \sim D_{S^*}}[R_T] \geq \left(T - \mathop{\mathbf{E}}_{z_t \sim D_0}[N_{S^*}] - \frac{T}{2}\sqrt{- \mathop{\mathbf{E}}_{z_t \sim D_0}[N_{S^*}] \log(1-\varepsilon^2)} \right) \frac{2\varepsilon}{k(2k)^k}.$$

Suppose that S^* is chosen at random uniformly from $\mathcal{S}_k$, before play begins. Then, from the above inequality, the expected regret is bounded as

$$\begin{aligned}\mathop{\mathbf{E}}_{S^*, z_t \sim D_{S^*}}[R_T] &\geq \frac{1}{|\mathcal{S}_k|} \sum_{S^* \in \mathcal{S}_k} \left(T - \mathop{\mathbf{E}}_{z_t \sim D_0}[N_{S^*}] - \frac{T}{2}\sqrt{- \mathop{\mathbf{E}}_{z_t \sim D_0}[N_{S^*}] \log(1-4\varepsilon^2)} \right) \frac{2\varepsilon}{k(2k)^k} \\ &\geq \left(T - \frac{T}{|\mathcal{S}_k|} - \frac{T}{2}\sqrt{-\frac{T}{|\mathcal{S}_k|}\log(1-4\varepsilon^2)} \right) \frac{2\varepsilon}{k(2k)^k},\end{aligned}$$

where the second inequality comes from $\sum_{S \in \mathcal{S}_k} \mathop{\mathbf{E}}_{z_t \sim D_0}[N_S] = T$ and $\sum_{S \in \mathcal{S}_k} \sqrt{\mathop{\mathbf{E}}_{z_t \sim D_0}[N_S]} \leq \sqrt{T|\mathcal{S}_k|}$. Using the inequality $-\log(1-x) \leq x/2$ for $x \in [0, 1/4]$, we have

$$\mathop{\mathbf{E}}_{S^*, z_t \sim D_{S^*}}[R_T] \geq T \left(1 - \frac{1}{|\mathcal{S}_k|} - \varepsilon\sqrt{\frac{T}{2|\mathcal{S}_k|}} \right) \frac{2\varepsilon}{k(2k)^k}$$

for $\varepsilon \in [0, 1/4]$. By setting $\varepsilon = \min\{1/4, \frac{1}{2}\sqrt{\frac{|\mathcal{S}_k|}{T}}\}$, we obtain

$$\mathop{\mathbf{E}}_{S^*, z_t \sim D_{S^*}}[R_T] = \Omega\left(\min\left\{ \frac{T}{k(2k)^k}, \sqrt{\frac{T|S_k|}{k^2(2k)^{2k}}} \right\} \right) \geq \Omega\left(\min\left\{ \frac{T}{k(2k)^k}, \sqrt{T\left(\frac{d}{5k^3}\right)^k} \right\} \right), \tag{8.26}$$

where the second inequality follows from $|\mathcal{S}_k| = \binom{d}{k} \geq (\frac{d}{k})^k$ and $k^2 = O((\frac{5}{4})^k)$.

Consider an arbitrary randomized algorithm and let λ denote the algorithm's internal randomization. Then, since λ is probabilistically independent from S^*, r and (8.26) for all deterministic algorithms, we have

$$\mathop{\mathbf{E}}_{S^*, \{r\}}\left[\mathop{\mathbf{E}}_{\lambda}[R_T]\right] = \mathop{\mathbf{E}}_{\lambda}\left[\mathop{\mathbf{E}}_{S^*, \{r\}}[R_T]\right] = \Omega\left(\min\left\{ \frac{T}{k(2k)^k}, \sqrt{T\left(\frac{d}{5k^3}\right)^k} \right\} \right).$$

□

8.5 Experimental Evaluation

We show the empirical performance of our algorithms through experiments over synthetic and real-world data. In this section, we consider the online portfolio selection problem with $\mathcal{S} = \mathcal{S}_1$. A problem instance is parameterized by a tuple $(d, T, \{r_t\}_{t=1}^T)$. A synthetic instance is generated as follows: given parameters d, T, C_1, and C_2, we randomly choose an asset i^* from $[d]$, and generate $r_{ti^*} \sim U((C_2+C_1)/2, C_2)$ and $r_{ti} \sim U(C_1, C_2)$ for $i \in [d] \setminus \{i^*\}$.

We also conduct our experiments for two real-world instances. The first is based on crypto coin historical data[1], including dates and price data for 19 crypto coins. From this data, we select 7 crypto coins, each having 929 prices, and obtain price relatives r_{ti} of coin i at time t by $(p_{ti}/p_{t-1,i}) - 1$, where p_{ti} indicates the price of coin i at time t. Thus, $d = 7$ and $T = 928$ in this instance. The other instance is based on S&P 500 stock data[2], including dates and price data for 505 companies. From this data, we choose $d = 470$ companies, each having 1259 stock prices, and compute $T = 1258$ price relatives for each company in the same way.

For purposes of comparison, we prepare three baseline algorithms: Exp3_cont, Exp3_disc, and MWU_disc. MWU_disc (based on MWU [11]) works in the full-feedback setting and is compared with Algorithm 12. Exp3_cont and Exp3_disc (based on Exp3 [16]) work in the bandit-feedback setting and are compared with Algorithm 13. These baseline algorithms have different ways of updating x_t^S from those of Algorithms 12 and 13. Note that since $\mathcal{S} = \mathcal{S}_1 = \{\{i\} \mid i \in [d]\}$, x_t^S can be expressed as $x_t^S = x_t^{\{i\}} = [0, \ldots, 0, x_{ti}, 0, \ldots, 0]^\top$. Below, we offer a brief explanation of the comparisons.

MWU_disc Set $x_{ti} = 1$ if $\sum_{j=1}^{t-1} r_{ji} \geq 0$ and $x_{ti} = 0$ otherwise. For each $t \in [T]$, select i_t by MWU, where rewards in the t-th round are given by $[\log(1 + r_{ti}x_{ti})]_{i=1}^d$, and output $i_t, x_t^{\{i_t\}}$.

Exp3_disc Set $x_{ti} = 1$ if $\sum_{j \in [t-1]: i_j = i} r_{ji} \geq 0$ and $x_{ti} = 0$ otherwise. For each $t \in [T]$, select i_t by Exp3, where reward in the t-th round is given by $\log(1 + r_{ti_t}x_{ti_t})$, and output $i_t, x_t^{\{i_t\}}$.

Exp3_cont Set a parameter $B \in \mathbb{N}$, and consider an MAB problem instance with $d(B + 1)$ arms in which the rewards for the $d(B + 1)$ arms in the t-th round are given by $(\log(1 + r_{ti}b/B))_{1 \leq i \leq d, 0 \leq b \leq B}$. Apply Exp3 to this MAB problem instance.

We assess the performance of the algorithms on the basis of regrets for synthetic instances and of cumulative price relatives for real-world instances, where regrets and cumulative price relatives are averaged over 10 executions. We set parameters η according to Theorem 8.3.1 for Algorithm 12 and MWU_disc, and η and γ according to Theorem 8.3.2 for Algorithm 13, Exp3_disc, and Exp3_cont.

Figure 8.1 shows average regrets for a synthetic instance with the following parameters: $(d, T, C_1, C_2) = (20, 10000, -0.5, 0.5)$. We observe that both Algorithms 12 and 13 converge faster than MWU_disc, Exp3_cont, and Exp3_disc. In addition, the results empirically show that our theoretical bounds are correct.

Figures 8.2 and 8.3 show average cumulative price relatives for a real-world instance of S&P 500 stock data with $(d, T, C_1, C_2) = (470, 1258, -0.34, 1.04)$ and for a real-world instance of crypto coin data with $(d, T, C_1, C_2) = (7, 928, -0.7, 3.76)$, respectively. From these figures, we observe that the cumulative price relatives of our algorithms are higher than those of baseline algorithms.

[1]`https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory`

[2]`https://www.kaggle.com/camnugent/sandp500`

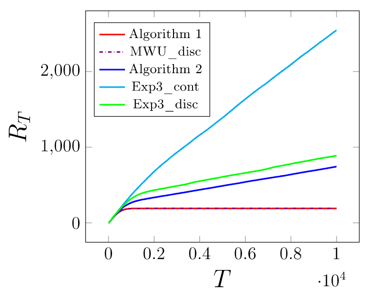


Figure 8.1: The average regrets over the synthetic dataset with $(d, T, C_1, C_2) = (20, 10000, -0.5, 0.5)$

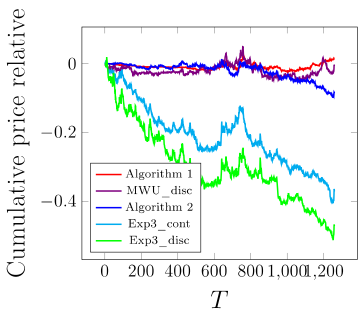


Figure 8.2: The average cumulative price relatives over S&P 500 stock dataset

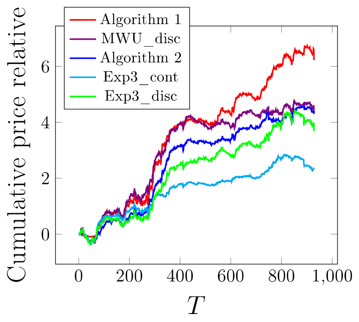


Figure 8.3: The average cumulative price relatives over the cryptocoin historical dataset

Chapter 9

Conclusion

In this thesis, we considered online optimization with limited information. We have presented algorithms and have analyzed the complexity for several problem settings including bandit online optimization, bandit linear combinatorial optimization, submodular function minimization with noisy evaluation oracle, bandit convex optimization, and online portfolio selection. Through these studies, we have assessed the performance of online algorithms and their limitations, as well as computational complexity.

For bandit linear optimization, we provided an oracle-efficient algorithm, by which we proved that the computational complexity of each bandit problem is equivalent to that for underlying offline optimization problem under polynomial-time reduction. Interestingly, we achieved a smaller oracle-complexity bound in stochastic settings than in non-stochastic settings, which might imply that non-stochastic setting is computationally harder. To prove a gap between these two settings in terms of computational complexity, we require lower bounds for the oracle complexity of online optimization, which seems to be currently open.

For some special cases of bandit combinatorial optimization, we presented tighter lower bounds for regrets, in which abundant $\log T$-factors are shaved off. Further, our lower bounds apply to more general problems than existing results. There has been, however, remained a gap of logarithmic factors in the dimensionalities of the feasible region between the upper and lower bounds. Furthermore, tight regret bounds that apply to general feasible regions are still open. To prove this, a novel technique for proving would be required.

We have also introduced the problem setting of submodular function minimization with noisy evaluation oracle, and have provided algorithms and lower bounds. The lower and upper bounds together imply that the proposed algorithms achieve nearly optimal additive errors, modulo $O(\sqrt{n})$ factors. Removing this $O(\sqrt{n})$-gap would be an interesting feature work. For the special cases of k-point feedback settings with a bounded k at least 2, we have provided a tight error bound up to constant factors. For the other cases, we leave it as an open question to find tight bounds. Besides, extending our results to the non-stochastic bandit setting is also left as a feature work.

For bandit convex optimization problems with strongly-convex and smooth objectives, we provided an algorithm with tight regret bounds under milder assumptions than existing works. However, the assumption of the existence of interior optimal solutions might be abundant, and the tight regret bounds might apply to more general problem settings. To prove that our algorithm works without the assumption, we need to show a novel concentration inequality for log-concave distributions (more precisely, multi-dimensional truncated normal distributions).

Finally, we considered online portfolio selection with combinatorial constraints and introduced two problem settings of full-feedback and bandit-feedback. For each setting, we provided an algorithm that achieves a nearly minimax optimal regret bound. Interestingly, the gap between tight regret bounds for these two settings are exponentially large while such a large gap does not appear in problems with linear objectives. This result implies that, compared to linear problems, the price of information gets increased drastically in nonlinear problems.

Through the above studies, we have obtained a deeper understanding of the relationship among three concepts: information, computation, and optimality, in the context of online optimization. In particular for the class of bandit linear optimization including combinatorial bandits, we have concluded that both computational efficiency and optimal regret bound can be achieved simultaneously, which implies that online linear optimization problems are essentially free from trade-off relation among information, computation, and optimality. Even for nonlinear problems such as submodular minimization and convex optimization, we have offered improved regret bounds with efficient algorithms, for some special cases. However, for more general problems, such as bandit convex optimization for general convex loss and bandit submodular minimization, optimal rates have been still open, which would be significant future work.

Bibliography

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] N. Abe and P. M. Long. Associative reinforcement learning using linear probabilistic concepts. In *International Conference on Machine Learning*, pages 3–11, 1999.

[3] M. Abeille, A. Lazaric, et al. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2):5165–5197, 2017.

[4] A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire. Algorithms for portfolio management based on the Newton method. In *International Conference on Machine Learning*, pages 9–16, 2006.

[5] A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Conference on Learning Theory*, pages 28–40, 2010.

[6] A. Agarwal, D. P. Foster, D. J. Hsu, S. M. Kakade, and A. Rakhlin. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1035–1043, 2011.

[7] S. Agrawal and N. 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, 2014.

[8] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

[9] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley & Sons, 2004.

[10] N. Alon, N. Cesa-Bianchi, C. Gentile, S. Mannor, Y. Mansour, and O. Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

[11] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[12] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, pages 217–226, 2009.

[13] J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.

[14] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

[15] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[16] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[17] B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pages 45–53, 2004.

[18] F. Bach. Convex analysis and optimization with submodular functions: A tutorial. *arXiv preprint arXiv:1010.4207*, 2010.

[19] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

[20] A. Belloni, T. Liang, H. Narayanan, and A. Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *Conference on Learning Theory*, pages 240–265, 2015.

[21] G. Bitran and R. Caldentey. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229, 2003.

[22] E. Blais, C. L. Canonne, T. Eden, A. Levi, and D. Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2113–2132, 2018.

[23] A. Blum. On-line algorithms in machine learning. In *Online algorithms*, pages 306–325. Springer, 1998.

[24] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 2005.

[25] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225, 2002.

[26] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[27] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[28] G. W. Brown. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13(1):374–376, 1951.

[29] G. W. Brown and J. Von Neumann. Solutions of games by differential equations. *Ann. Math. Studies*, 24:73–79, 1950.

[30] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[31] S. Bubeck and R. Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*, pages 583–589, 2016.

[32] S. Bubeck, N. Cesa-Bianchi, and S. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41.1–41.14, 2012.

[33] S. Bubeck, O. Dekel, T. Koren, and Y. Peres. Bandit convex optimization: $\sqrt{T}$ regret in one dimension. In *Conference on Learning Theory*, pages 266–278, 2015.

[34] S. Bubeck, Y. T. Lee, and R. Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 72–85, 2017.

[35] P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications.* Springer Science & Business Media, 2011.

[36] F. Caro and J. Gallien. Clearance pricing optimization for a fast-fashion retailer. *Operations Research*, 60(6):1404–1422, 2012.

[37] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[38] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[39] D. Chakrabarty, P. Jain, and P. Kothari. Provable submodular minimization using wolfe's algorithm. In *Advances in Neural Information Processing Systems*, pages 802–809, 2014.

[40] D. Chakrabarty, Y. T. Lee, A. Sidford, and S. C.-w. Wong. Subquadratic submodular function minimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1220–1231, 2017.

[41] L. Chen, M. Zhang, and A. Karbasi. Projection-free bandit convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2047–2056, 2019.

[42] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[43] K. L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM (JACM)*, 42(2):488–499, 1995.

[44] A. Cohen, T. Hazan, and T. Koren. Tight bounds for bandit combinatorial optimization. In *Conference on Learning Theory*, pages 629–642, 2017.

[45] R. Combes, M. S. T. M. Shahi, A. Proutiere, and M. Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.

[46] T. H. Cormen. *Introduction to Algorithms.* MIT press, 2009.

[47] T. M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

[48] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 2012.

[49] D. Dadush, L. A. Végh, and G. Zambelli. Geometric rescaling algorithms for submodular function minimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 832–848. SIAM, 2018.

[50] V. Dani and T. P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 937–943, 2006.

[51] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, pages 355–366, 2008.

[52] V. Dani, S. M. Kakade, and T. P. Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2008.

[53] P. Das. Online convex optimization and its application to online portfolio selection. 2014.

[54] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[55] P. M. Fenwick. A new data structure for cumulative frequency tables. *Software: Practice and Experience*, 24(3):327–336, 1994.

[56] K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, pages 69–88, 2015.

[57] A. Fiat and G. J. Woeginger. *Online Algorithms: The State of the Art*, volume 1442. Springer, 1998.

[58] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[59] D. Foster, S. Kale, and H. Karloff. Online sparse linear regression. In *Conference on Learning Theory*, pages 960–970, 2016.

[60] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[61] K. Fujii and H. Kashima. Budgeted stream-based active learning via adaptive submodular maximization. In *Advances in Neural Information Processing Systems*, pages 514–522, 2016.

[62] K. Fujii and S. Sakaue. Beyond adaptive submodularity: Approximation guarantees of greedy policy with adaptive submodularity ratio. In *International Conference on Machine Learning*, pages 2042–2051, 2019.

[63] S. Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

[64] S. Fujishige. *Submodular Functions and Optimization*, volume 58. Elsevier, 2005.

[65] S. Fujishige and S. Isotani. A submodular function minimization algorithm based on the minimum-norm base. *Pacific Journal of Optimization*, 7(1):3–17, 2011.

[66] D. Garber. Efficient online linear optimization with approximation algorithms. In *Advances in Neural Information Processing Systems*, pages 627–635, 2017.

[67] J. E. Gentle. *Computational Statistics*. Springer Science & Business Media, 2009.

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

[69] L. Gorelick, Y. Boykov, O. Veksler, I. Ayed, and A. Delong. Submodularization for binary pairwise energies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1154–1161, 2014.

[70] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[71] A. György, T. Linder, G. Lugosi, and G. Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8(Oct):2369–2403, 2007.

[72] N. H. Hakansson and W. T. Ziemba. Capital growth theory. *Handbooks in Operations Research and Management Science*, 9:65–86, 1995.

[73] M. E. Halabi and S. Jegelka. Minimizing approximately submodular functions. *arXiv preprint arXiv:1905.12145*, 2019.

[74] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

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

[76] A. Hassidim and Y. Singer. Submodular optimization under noise. In *Conference on Learning Theory*, pages 1069–1122, 2017.

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

[78] E. Hazan and S. Kale. Online submodular minimization. *Journal of Machine Learning Research*, 13(Oct):2903–2922, 2012.

[79] E. Hazan and Z. Karnin. Volumetric spanners: An efficient exploration basis for learning. *Journal of Machine Learning Research*, 17(1):4062–4095, 2016.

[80] E. Hazan and K. Levy. Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems*, pages 784–792, 2014.

[81] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[82] E. Hazan, W. Hu, Y. Li, and Z. Li. Online improper learning with an approximation oracle. In *Advances in Neural Information Processing Systems*, pages 5652–5660, 2018.

[83] D. P. Helmbold and M. K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10(Jul):1705–1736, 2009.

[84] X. Hu, L. Prashanth, A. György, and C. Szepesvári. (Bandit) convex optimization with biased noisy gradient oracles. In *International Conference on Artificial Intelligence and Statistics*, pages 819–828, 2016.

[85] S. Ito. Submodular function minimization with noisy evaluation oracle. In *Advances in Neural Information Processing Systems*, pages 12103–12113, 2019.

[86] S. Ito. An optimal algorithm for bandit convex optimization with strongly-convex and smooth loss. In *International Conference on Artificial Intelligence and Statistics*, to appear, 2020.

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

[88] S. Ito and R. Fujimaki. Optimization beyond prediction: Prescriptive price optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1833–1841, 2017.

[89] S. Ito, D. Hatano, H. Sumita, A. Yabe, T. Fukunaga, N. Kakimura, and K. Kawarabayashi. Online regression with partial information: Generalization and linear projection. In *International Conference on Artificial Intelligence and Statistics*, pages 1599–1607, 2018.

[90] S. Ito, D. Hatano, H. Sumita, A. Yabe, T. Fukunaga, N. Kakimura, and K. Kawarabayashi. Regret bounds for online portfolio selection with a cardinality constraint. In *Advances in Neural Information Processing Systems*, pages 10588–10597, 2018.

[91] S. Ito, D. Hatano, H. Sumita, K. Takemura, T. Fukunaga, N. Kakimura, and K.-I. Kawarabayashi. Improved regret bounds for bandit combinatorial optimization. In *Advances in Neural Information Processing Systems*, pages 12050–12059, 2019.

[92] S. Ito, D. Hatano, H. Sumita, K. Takemura, T. Fukunaga, N. Kakimura, and K.-I. Kawarabayashi. Oracle-efficient algorithms for online linear optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 10590–10599, 2019.

[93] S. Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.

[94] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[95] K. G. Jamieson, R. Nowak, and B. Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2012.

[96] S. Jegelka and J. Bilmes. Online submodular minimization for combinatorial structures. In *International Conference on Machine Learning*, pages 345–352, 2011.

[97] S. Jegelka and J. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1897–1904, 2011.

[98] S. M. Kakade, A. T. Kalai, and K. Ligett. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.

[99] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3(Nov):423–440, 2002.

[100] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[101] S. Kale. Open problem: Efficient online sparse regression. In *Conference on Learning Theory*, pages 1299–1301, 2014.

[102] S. Kale, Z. Karnin, T. Liang, and D. Pál. Adaptive feature selection: Computationally efficient online sparse linear regression under RIP. In *International Conference on Machine Learning*, pages 1780–1788, 2017.

[103] M. Karimi, M. Lucic, H. Hassani, and A. Krause. Stochastic submodular maximization: The case of coverage functions. In *Advances in Neural Information Processing Systems*, pages 6853–6863, 2017.

[104] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[105] E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[106] J. Kelly. A new interpretation of information rate. *Bell Sys. Tech. Journal*, 35:917–926, 1956.

[107] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12(363-366):234, 1960.

[108] R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704, 2005.

[109] P. Kohli and P. H. Torr. Dynamic graph cuts and their applications in computer vision. In *Computer Vision*, pages 51–108. Springer, 2010.

[110] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[111] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[112] J. Komiyama, J. Honda, and H. Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. In *International Conference on Machine Learning*, pages 1152–1161, 2015.

[113] W. Kotłowski and G. Neu. Bandit principal component analysis. *Proceedings of Machine Learning Research*, 99:1–31, 2019.

[114] D. Koushik, J. A. Higbie, and C. Eister. Retail price optimization at intercontinental hotels group. *Interfaces*, 42(1):45–57, 2012.

[115] W. Kumagai. Regret analysis for continuous dueling bandit. In *Advances in Neural Information Processing Systems*, pages 1489–1498, 2017.

[116] P. Lagrée, C. Vernade, and O. Cappe. Multiple-play bandits in the position-based model. In *Advances in Neural Information Processing Systems*, pages 1597–1605, 2016.

[117] T. L. Lai. Adaptive treatment allocation and the multi-armed bandit problem. *Annals of Statistics*, 15(3):1091–1114, 1987.

[118] T. Lattimore and C. Szepesvári. Bandit algorithms. *Preprint, Revision: 1699*, 2019.

[119] T. Lattimore, B. Kveton, S. Li, and C. Szepesvari. Toprank: A practical algorithm for online stochastic ranking. In *Advances in Neural Information Processing Systems*, 2018.

[120] S. Lee. *Study of Demand Models and Price Optimization Performance*. PhD thesis, Georgia Institute of Technology, 2011.

[121] Y. T. Lee, A. Sidford, and S. C.-w. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065, 2015.

[122] B. Li and S. C. Hoi. On-line portfolio selection with moving average reversion. In *International Conference on Machine Learning*, pages 563–570, 2012.

[123] B. Li and S. C. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys*, 46 (3):35, 2014.

[124] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[125] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 284(1-3):193–228, 1998.

[126] L. Lovász. Submodular functions and convexity. In *Mathematical Programming: The State of the Art*, pages 235–257. Springer, 1983.

[127] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

[128] A. Marshall. *Principles of Economics.* Library of Economics and Liberty, 1920.

[129] J. Matoušek and J. Vondrák. The probabilistic method. *Lecture Notes*, 2001.

[130] J. I. McGill and G. J. Van Ryzin. Revenue management: Research overview and prospects. *Transportation Science*, 33(2):233–256, 1999.

[131] H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *International Conference on Computational Learning Theory*, pages 109–123, 2004.

[132] M. Mohri and S. Yang. Adaptive algorithms and data-dependent guarantees for bandit convex optimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 815–824, 2016.

[133] A. Mokhtari, H. Hassani, and A. Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895, 2018.

[134] T. Murata and T. Suzuki. Sample efficient stochastic gradient iterative hard thresholding method for stochastic sparse linear regression with limited attribute observation. In *Advances in Neural Information Processing Systems*, pages 5312–5321, 2018.

[135] H. Narayanan and A. Rakhlin. Efficient sampling from time-varying log-concave distributions. *Journal of Machine Learning Research*, 18(1):4017–4045, 2017.

[136] M. Natter, T. Reutterer, and A. Mild. Dynamic pricing support systems for DIY retailers–A case study from austria. *Marketing Intelligence Review*, 1:17–23, 2009.

[137] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, 1994.

[138] Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[139] E. Ordentlich and T. M. Cover. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research*, 23(4):960–982, 1998.

[140] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[141] R. L. Phillips. *Pricing and Revenue Optimization.* Stanford University Press, 2005.

[142] A. Prékopa. Logarithmic concave measures with application to stochastic programming. *Acta Scientiarum Mathematicarum*, 32:301–316, 1971.

[143] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[144] J. Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[145] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[146] P. Rusmevichientong and J. N. Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[147] P. Rusmevichientong, B. Van Roy, and P. W. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006.

[148] S. Sakaue, M. Ishihata, and S.-i. Minato. Efficient bandit combinatorial optimization algorithm with zero-suppressed binary decision diagrams. In *International Conference on Artificial Intelligence and Statistics*, pages 585–594, 2018.

[149] A. Saumard and J. A. Wellner. Log-concavity and strong log-concavity: A review. *Statistics Surveys*, 8:45, 2014.

[150] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, 1998.

[151] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[152] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[153] O. Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pages 3–24, 2013.

[154] O. Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017.

[155] A. Singla, S. Tschiatschek, and A. Krause. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[156] C. J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, pages 689–705, 1985.

[157] E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4(Oct):773–818, 2003.

[158] M. Tang, I. B. Ayed, and Y. Boykov. Pseudo-bound optimization for binary energies. In *European Conference on Computer Vision*, pages 691–707, 2014.

[159] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[160] T. Uchiya, A. Nakamura, and M. Kudo. Algorithms for adversarial bandit problems with multiple plays. In *International Conference on Algorithmic Learning Theory*, pages 375–389, 2010.

[161] M. Valko, R. Munos, B. Kveton, and T. Kocák. Spectral bandits for smooth graph functions. In *International Conference on Machine Learning*, pages 46–54, 2014.

[162] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[163] V. G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–386, 1990.

[164] J. Wang, R. Fujimaki, and Y. Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2015.

[165] M. K. Warmuth and D. Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(Oct): 2287–2320, 2008.

[166] P. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11(1): 128–149, 1976.

[167] Y. Ye, L. Lei, and C. Ju. Hones: A fast and tuning-free homotopy method for online newton step. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2017, 2018.

[168] J. Zimmert and Y. Seldin. An optimal algorithm for stochastic and adversarial bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 467–475, 2019.

[169] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.