Structured Online Learning with Full and Bandit Information

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nicholas Johnson

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

ARINDAM BANERJEE

September, 2016

Acknowledgements

I would like to thank many people who supported me throughout my graduate career. First, I would like to thank my advisor Professor Arindam Banerjee for taking me on as a student, teaching me everything I know about machine learning, encouraging me to pursue my own intellectual curiosities, providing career opportunities and guidance, and for his overall enthusiasm and professionalism in research and life. I will forever be grateful. To the rest of my committee – Professor Maria Gini for providing incredible advice whenever I needed it, Professor Rui Kuang for helping start my graduate career in a positive way, and Professor Jarvis Haupt for the thought-provoking questions.

To my fellow graduate students in the Artificial Intelligence lab: Steve Damer, Mohamed Elidrisi, Julio Godoy, Will Groves, Elizabeth Jensen, Marie Manner, Ernesto Nunes, James Parker, Erik Steinmetz, and Mark Valovage, and in the Machine Learning lab: Soumyadeep Chatterjee, Sheng Chen, Konstantina Christakopoulou, Puja Das, Farideh Fazayeli, Jamal Golmohammady, Andre Goncalves, Qilong Gu, Igor Melnyk, Vidyashankar Sivakumar, Karthik Subbian, Amir Taheri, and Huahua Wang. I learned from each of you and will never forget the support you provided especially during the late nights in Walter Library B22 and Keller Hall 6-210 working toward paper deadlines.

To my collaborators at Amazon – Aaron Dykstra, Ted Sandler, Houssam Nassif, Jeff Bilger, and Charles Elkan for teaching me how to use machine learning in the wild.

Finally, to my mom and dad for their continuous love and support and for instilling in me the value of hard work, passion in my interests, and balance in life. To the rest of my family and friends for always being there.

The research in this thesis was supported in part by NSF grants IIS-1447566, IIS-1422557, CCF-1451986, CNS-1314560, IIS-0953274, IIS-1029711, IIS-0916750, NSF CAREER award IIS-0953274, NASA grant NNX12AQ39A, Adobe, IBM, and Yahoo.

Dedication

To my family.

Abstract

Numerous problems require algorithms to repeatedly interact with the environment which may include humans, physical obstacles such as doors or walls, biological processes, or even other algorithms. One popular application domain where such repeated interaction is necessary is in social media. Every time a user uses a social media application, an algorithm must make a series of decisions on what is shown ranging from news content to friend recommendations to trending topics. After which, the user provides feedback frequently in the form of a click or no click. The algorithm must use such feedback to learn the user's likes and dislikes. Similar scenarios play out in medical treatments, autonomous robot exploration, online advertising, and algorithmic trading.

In such applications, users often have high expectations of the algorithm such as immediately showing relevant recommendations, quickly administering effective treatments, or performing profitable trades in fractions of a second. Such demands require algorithms to have the ability to learn in a dynamic environment, learn efficiently, and provide high quality solutions. Designing algorithms which meet such user demands poses significant challenges for researchers.

In this thesis, we design and analyze machine learning algorithms which interact with the environment and specifically study two aspects which can help alleviate challenges: (1) learning online where the algorithm selects an action after which it receives the outcome (i.e., loss) of selecting such an action and (2) using the structure (sparsity, group sparsity, low-rankness) of a solution or user model. We explore such aspects under two feedback models: full and bandit information. With full information feedback, the algorithm observes the loss of each possible action it could have selected, for example, a trading algorithm can observe the price of each stock it could have invested in at the end of the day. With bandit information feedback, the algorithm can only observe the loss of the action taken, for example, a medical treatment algorithm can only observe whether a patient's health improved for the treatment provided. We measure the performance of our algorithms by their regret which is the difference between the cumulative loss received by the algorithm and the cumulative loss received by the best fixed or time-varying actions in hindsight.

In the first half of this thesis, we focus on full information settings and study online learning algorithms for general resource allocation problems motivated, in part, by applications in algorithmic trading. The first two topics we explore are controlling the cost of updating an allocation, for example, one's stock portfolio, and learning to allocate a resource across groups of objects such as stock market sectors. In portfolio selection, making frequent trades may incur huge amounts of transaction costs and hurt one's bottom line. Moreover, groups of stocks, may perform similarly and investing in a few groups may lead to higher returns. We design and analyze two efficient algorithms and present new theoretical regret bounds. Further, we experimentally show the algorithms earn more wealth than existing algorithms even with transaction costs.

The third and fourth topics we consider are two different ways to control suitable measures of risk associated with a resource allocation. The first approach is through diversification and the second is through the concept of hedging where, in the application of portfolio selection, a trader borrows shares from the bank and holds both long and short positions. We design and analyze two efficient online learning algorithms which either diversify across groups of stocks or hedge between individual stocks. We establish standard regret bounds and show experimentally our algorithms earn more wealth, in some cases orders of magnitude more, than existing algorithms and incur less risk.

In the second half of this thesis, we focus on bandit information settings and how to use the structure of a user model to design algorithms with theoretically sharper regret bounds. We study the stochastic linear bandit problem which generalizes the widely studied multi-armed bandit. In the multi-armed bandit, an algorithm repeatedly selects an arm (action) from a finite decision set after which it receives a stochastic loss. In the stochastic linear bandit, arms are selected from a decision set with infinitely many arms (e.g., vectors from a compact set) and the loss is a stochastic linear function parameterized by an unknown vector. The first topic we explore is how the regret scales when the unknown parameter is structured (e.g., sparse, group sparse, low-rank). We design and analyze an algorithm which uses the structure to construct tight confidence sets which contain the unknown parameter with high-probability which leads to sharp regret bounds. The second topic we explore is how to generalize the previous algorithm to non-linear losses often used in Generalized Linear Models. We design and analyze a similar algorithm and show the regret is of the same order as with linear losses.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern advances in science and technology have lead to paradigm shifts in the acceptance and use of technology in everyday life. Nearly everyone in the US has access to a computer, tablet, or smartphone. Further, constant use of such devices, especially smartphones, is becoming socially acceptable. Data from the use of such devices, for example, a user's website viewing habits, online purchasing history, email and text communications, and social network connections, are valuable and can be used in numerous ways. Companies, in particular, are interested in using such data to better understand their customer base in order to more accurately target advertisements, provide relevant content, and administer effective medical treatments.

Analyzing the huge amount of heterogeneous data generated each day from such devices for any particular application is a significant challenge. Classical rule based methods such as expert systems in artificial intelligence fail to compute effective solutions in many modern applications due, in part, to the shear amount of data and the highly non-trivial and unintuitive correlations inherent in the data. In response, machine learning algorithms have become popular as a way to intelligently use data to compute effective solutions to modern problems.

Machine learning algorithms are designed to learn complex models from data, rather than relying on a set of rules as in expert systems, which make them a natural choice in the "Big Data" era. Many modern applications accumulate data over time and require algorithms to learn and adapt efficiently and effectively in real-time to a changing environment and update models dynamically. However, traditional machine learning

algorithms are quickly becoming insufficient for such problems because they often rely on the availability of a static dataset from which a model can be learned and are unable to learn and adapt efficiently in real-time. Therefore, there is an increasing need for new machine learning algorithms to learn to compute solutions dynamically through repeated interactions with an environment. Such a need has lead to increased and ongoing research [52, 49, 85] in the area of interactive machine learning which has found real-world applications from drug discovery [125] to user interface design [51] to teaching robots to bake a cake [117].

1.1 Interactive Machine Learning

Modern applications require algorithms to interact with an environment which may include humans, physical obstacles such as doors or walls, biological processes such as a metabolic pathway, or even interact with other algorithms in order to learn and adapt to changes. For example, in social media applications, users are actively engaged with the application through clicking on recommended news articles, friend profiles, and advertisements. The application can observe the user's actions and such actions provide valuable insights into the user's likes and dislikes. In order to take advantage of such feedback and learn about the user to provide a satisfying user experience, the application must use algorithms which can learn through interaction. Interactive learning is not exclusive to social media applications as similar interactive scenarios can also be found in medical treatments, autonomous robot exploration, and algorithmic trading.

There are a number of common technical challenges present in such problems. First, simply the act of learning and adapting in a dynamic environment such as learning a user's media preferences or a patient's sensitivity to medical treatments which may change over time is challenging. Second, many problems require algorithms to be efficient and compute solutions in fractions of a second. For example, a recommendation algorithm must compute a new set of recommendations which can be shown to the user almost immediately after the user loads the application – users are often not willing to wait minutes to view their recommendations. Third, many problems and users expect high quality solutions such as relevant recommendations, effective medical treatments, and investment portfolios which earn money with low risk.

In this thesis, we consider the design and analysis of interactive machine learning algorithms. We study two aspects which can help alleviate the aforementioned challenges in such problems: (1) learning incrementally (i.e., online) to facilitate efficient computation and adaptation in dynamic environments and (2) using the structure (sparsity, group sparsity, low-rankness) of a solution or user model to compute effective solutions.

1.1.1 Online Learning

Online learning is a technique where an algorithm sequentially learns a model. As new data is made available, an online learning algorithm updates its current model using only the new data. Such a learning process is in contrast to traditional (offline) learning algorithms which learn a single model using a fixed dataset. Online learning algorithms can efficiently compute new models and quickly adapt to changes in a dynamic environment which make them a natural choice for interactive learning problems.

1.1.2 Structured Learning

Structured learning is a type of learning problem where the learning algorithm computes a solution or learns a model with a specific type of structure. Often the type of structure is known a priori, for example, in movie recommendation applications it may be known that the user prefers only a few movie genres so the user preference vector has a sparse structure. The goal of the algorithm is to learn which of the genres the user prefers with knowledge that there are only a few. The use of structure decreases the number of possible user models the algorithm must consider thereby allowing the algorithm to efficiently compute a model. In this thesis, we will consider a solution or user model to be structured if it has a small value according to some measure of complexity such as a norm. Popular types of structure we will consider include sparsity, group sparsity, and low-rank which can all be captured using norms (1-norm, 2-norm, nuclear norm).

1.1.3 Full and Bandit Information

In interactive machine learning, the algorithm provides a solution and the user (or some entity 'Nature') observes the solution and provides feedback. For example, if the solution is a ranked list of websites, the user's feedback will be a click or no-click on one of

the website links and if the solution is a mixture of drugs, the patient's feedback will be an improvement or decline in patient health. We consider two models of user feedback: full information and bandit information. In problems with full information feedback, the algorithm gets to observe the outcome (i.e., loss) of each action it could have taken, for example, a trading algorithm can observe the price of each stock at the end of the day and can compare how much wealth it could have earned had it invested in a different stock. In problems with bandit information feedback, the algorithm can only observe the outcome of the action taken, for example, a medical treatment algorithm can only observe whether a patient's health improved for the treatment provided. Bandit information problems are a subset of full information problems since with full information the algorithm can observe the outcome of the action it took as in bandit information problems as well as the outcome for all other actions. Full information problems provide the algorithm with more information with which it can use to compute its next solution. Therefore, learning in such problems is often easier than learning in problems with bandit feedback which we will see throughout this thesis.

1.2 Overview and Contributions

In the first half of this thesis, we focus on problems which admit full information. We design and analyze interactive online learning algorithms where the algorithm must decide how to distribute an (almost) infinitely divisible resource (e.g., wealth, time, food) across a number of objects (e.g., assets, compute jobs, people). We specifically focus on one sequential resource allocation problem: algorithmic trading.

In algorithmic trading, the algorithm must decide how to distribute its wealth across a number of assets such as stocks often with the goal of maximizing its cumulative wealth over a trading period. The algorithm must repeatedly interact with the market in order to learn which stocks perform well (e.g., earn wealth and/or reduce risk) and, often, must do so in fractions of a second.

We focus on designing algorithms which learn to construct a stock portfolio which is structured such as sparse (only investing in a few stocks), group sparse (only invested in a few market sectors), diverse (investing in a variety of different stocks), or hedged (hold long and short positions of similar stocks). Each type of structured solution we

consider is motivated by real-world uses and the work in the first half of the thesis is designed to close the gap between theory and practice which will lead to the use of such algorithms in real-world applications.

In the second half of this thesis, we study problems which admit bandit information. We specifically consider one such problem: the stochastic linear bandit. The problem requires the algorithm to select a sequence of points at which it can observe a noisy linear loss function value parameterized by an unknown parameter. The goal of the algorithm is to select a sequence of points which minimizes the total loss it receives. The problem is rather abstract but has a number of real-world applications, for example, in medical treatments where an algorithm must sequentially administer a mixture of drugs to a patient to determine the mixture which improves the patient's health the most.

1.2.1 Online Lazy Updates

The first topic we explore in this thesis is controlling the cost of updating one's portfolio. In algorithmic trading, the challenge is deciding how to allocate one's money across a variety of assets such as stocks. Frequently changing such an allocation may incur huge amounts of transaction costs and become detrimental to one's bottom line. The key idea is to trade infrequently (i.e., sparse trading) and only when it is worth the cost.

Contributions. We make the following contributions to the literature. First, we design and analyze an efficient online learning algorithm which performs sparse or lazy updates to a portfolio. The algorithm is the first in the literature to consider transaction costs in the design which guide what stocks are invested in and which includes theoretical guarantees on performance. Second, we present a theoretical analysis of our algorithm's performance. The analysis extends existing theory and shows our algorithm's performance is competitive with the best fixed and time-varying portfolios in hindsight under reasonable assumptions. Third, we perform extensive experiments on real-world stock price datasets and show our algorithm is able to earn more wealth over the trading periods than existing algorithms even with transaction costs.

1.2.2 Online Lazy Updates with Group Sparsity

The second topic we explore is how to compute a portfolio which invests in groups of assets such as stocks. Frequently in algorithm trading, groups of assets, such as market sectors, will perform similarly and investing in all assets of a few top performing groups leads to higher returns. However, the top performing groups of assets may change over time therefore, an algorithm must be able to repeatedly interact with and adapt to market trends focusing specifically on groups of assets.

Contributions. We make the following contributions to the literature. First, we design and analyze an efficient online learning algorithm which can utilize group information and identify which groups of assets to invest in. No previous portfolio selection algorithms have considered group information in their algorithmic framework. Second, we build on our previous theoretical analysis and show our algorithm's performance is competitive with the best fixed and time-varying portfolios which utilize group information in hindsight under reasonable assumptions. Third, we perform extensive experiments using two real-world stock price datasets where the stocks are grouped according to market sectors. The experimental results show our algorithm is able to accurately identify top performing market sectors and earn more wealth than existing algorithms including our previous algorithm which only considered transaction costs.

1.2.3 Online Structured Diversification

The third topic we consider is how to control the risk of allocating wealth across a set of assets such as stocks. One well-known approach to control risk is through diversifying one's portfolio however, naive diversification may be ineffective. In particular, distributing one's wealth across stocks that are similar may not accomplish the goal of reducing the risk of exposure to market crashes. The key idea is to distribute one's wealth across stocks that are not similar such that if one stock's price crashes the other stock's price will not crash and one can control the amount of wealth lost.

Contributions. We make the following contributions to the literature. First, we present a general framework and an efficient online learning algorithm which computes portfolios which are diverse across groups of stocks. The novel component of our formulation is the introduction of a new overlapping group norm which is used to limit the

maximum amount of wealth invested in any group which forces diversification across groups. No previous works have considered any mechanism to compute diversified portfolios. Second, we experiment with our algorithm using two real-world stock price datasets. We show our algorithm is able to earn more wealth than our previous two algorithms and existing algorithms in the literature. We also show our algorithm is effective at controlling risk and typically has less exposure to risk than competing algorithms. Third, we present a theoretical analysis of our algorithm's performance and show it is competitive with the best fixed portfolio in hindsight under reasonable assumptions.

1.2.4 Online Structured Hedging

In the previous three topics, we only consider investing with the current wealth on hand. In the final topic under full information feedback, we study how to design a learning algorithm which is allowed to borrow shares of stock from the bank with which to invest and how to use such borrowed stock to control risk. The idea is to sell the borrowed shares when the market price is high then buy them back when the price is low and return the shares to the bank. Such a strategy is known as holding a short position or shorting the market in finance. When shorting is used in conjunction with typical investing where one purchases shares of stock, i.e., holding a long position, one can strategically hold both long and short positions to hedge the market and reduce risk.

Contributions. We make the following contributions to the literature. First, we present a general framework and an efficient online learning algorithm which computes hedged portfolios using borrowed wealth and shares of stock. The algorithm is able to learn when to hold long and short positions in order to take advantage of both bear and bull markets. All previous algorithms could only hold long positions and they lose wealth when the market crashes. Moreover, we present the first algorithm in the literature to also use short and long positions to control risk through hedging. Second, we experiment with our algorithm using five real-world stock price datasets and show our algorithm is able to earn orders of magnitude more wealth than all of our previous algorithms and all existing algorithms in the literature. Moreover, our algorithm is exposed to less risk than previous algorithms under various measures of risk. Third, we present a theoretical analysis our algorithm and show its performance is competitive with the best fixed portfolio in hindsight under reasonable assumptions.

1.2.5 Structured Stochastic Linear Bandits

In the second half of this thesis, we switch our focus to bandit information problems where we study the stochastic linear bandit problem. The first topic we explore is how the regret, which is the difference between the cumulative loss of the algorithm and the cumulative loss of the best fixed vector in hindsight, scales when we assume the unknown parameter is structured (e.g., sparse, group sparse, low-rank).

Contribution. We make one significant contribution to the literature. We design and analyze an algorithm which constructs tight confidence sets which contain the unknown parameter with high-probability. Using such confidence sets, we show sharp theoretical regret bounds which depend linearly on the structure rather than linearly on the ambient dimensionality of the problem. Only a couple previous works have considered a similar setting where the unknown parameter is sparse. They show sharp regret bounds only for the sparse setting. We match their regret bounds and generalize the results to any norm structured parameter using a single unified algorithm and analysis.

1.2.6 Structured Stochastic Generalized Linear Bandits

The last topic we study is how to generalize the results of our previous algorithm for the structured stochastic linear bandit problem. We consider settings where the noisy loss function is non-linear rather than linear. Specifically, we consider loss functions which arise naturally from the Generalized Linear Models framework in statistics.

Contribution. We make one significant contribution to the literature. We design and analyze an algorithm which again constructs tight confidence sets which contain the unknown parameter with high-probability. Using such confidence sets, we show sharp theoretical regret bounds for a wider class of loss functions and when the unknown parameter is structured. We show the regret bounds match the bounds for linear loss functions and for all types of structure. One previous work has considered a non-linear loss function and showed matching bounds however, the work only considered one specific loss function and did not consider structured settings.

Chapter 2

Formal Setting

In this chapter, we will review background material focusing mostly on convex optimization which is necessary to understand the following chapters. We will then formally introduce the problem setting, measures of performance, types of feedback and structure, and the two algorithmic frameworks we will build on.

2.1 Preliminaries

We review some mathematical definitions of a number of objects which will play an important role in the design and analysis of our algorithms. In particular, we heavily rely on properties and techniques from convex optimization. The following definitions, which we repeat here for completeness, can be found in [20].

2.1.1 Notation

We will use the following notations. Vectors are either denoted as lower-case variables x or bold lower-case variables $\mathbf{x}$ and element i in vector x is denoted $x(i)$. Matrices are similarly denoted as upper-case variables X or upper-case bold variables $\mathbf{X}$. We denote the set of real numbers, n-dimensional real numbers, non-negative real numbers, and strictly positive real numbers as $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ respectively. For a vector $x \in \mathbb{R}^n$ we denote an L_p norm as $\|x\|_p = (\sum_{i=1}^n |x(i)|^p)^{1/p}$ and use $\|x\|$ to represent the L_2 norm by default. For two vectors x, y let $\langle x, y \rangle$ denote the inner product. Scalar-valued functions are denoted by the lower-case notation f and the gradient of a function f at

$x \in \mathbb{R}^n$ is $\nabla f(x)$. If $x \in \mathbb{R}$ then the derivative is $f'(x)$. $\nabla^2 f(x)$ is the Hessian of the function and $f''(x)$ is the second derivative. The trace of a matrix is trace(X) and is the sum of the diagonal elements of X. We use $\top$ to denote the transpose, e.g., $x^\top$.

2.1.2 Convex Sets and Functions

A set $\mathcal{X}$ is convex if the line segment between any two points in $\mathcal{X}$ lies in $\mathcal{X}$. In other words, for any $x_1, x_2 \in \mathcal{X}$ and any $0 \leq \theta \leq 1$ we have $\theta x_1 + (1-\theta)x_2 \in \mathcal{X}$. Examples of convex sets which we will use include:

- (closed) unit L_p norm balls $\bar{B}_p^n := \{x \in \mathbb{R}^n : \|x\|_p \leq 1\}$,
- ellipsoids $C := \{x \in \mathbb{R}^n : (x-v)^\top D(x-v) \leq 1\}$ centered at some vector v with axes defined through the positive definite matrix D,
- the n-dimensional probability simplex $\Delta_n := \{x \in \mathbb{R}^n : x(i) \geq 0 \; \forall i, \sum_{i=1}^n x(i)=1\}$,
- polytopes $\mathcal{P} := \{x \in \mathbb{R}^n : a_i^\top x \leq b_i, c_j^\top x = d_j, \text{ for } i = 1, \ldots, s \text{ and } j = 1, \ldots, t\}$ which can be interpreted as the intersection of a finite number of halfspaces and hyperplanes.

Let $f : \mathcal{X} \to \mathbb{R}$ be a function and $\mathcal{X}$ a convex set, then f is a convex function if for all $x_1, x_2 \in \mathcal{X}$ and $0 \leq \theta \leq 1$ the following holds

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) \; . \tag{2.1}$$

This can be interpreted as for any two points on the function, if a chord is drawn connecting them, then all points on the chord will lie on or above the function. One nice property of convex functions is that every local minima is a global minima. A function is *strictly convex* if (2.1) holds with strict inequality for any $x_1 \neq x_2 \in \mathcal{X}$. A strictly convex function has a unique minimum.

We can consider (2.1) as the zeroth-order definition of a convex function. The first-order definition is the following. Let f be a differentiable function then f is convex if and only if $\mathcal{X}$ is convex and

$$f(x_2) \geq f(x_1) + \langle \nabla f(x_1), x_2 - x_1 \rangle \quad \forall x_2 \in \mathcal{X} \; . \tag{2.2}$$

This shows that a function is convex if its linear approximations lie on or below the function. A strictly convex function is one where (2.2) holds with strict inequality. Note, (2.2) is the first-order Taylor expansion of the f at x_1 which is something we will make use of in Chapters 5-10.

The second-order definition of a convex function is the following. Let f be a twice differentiable function at each point in its convex domain $\mathcal{X}$. Then f is convex if and only if its Hessian is positive semidefinite, i.e., for all $x \in \mathcal{X}$ the condition

$$\nabla^2 f(x) \geq 0 \tag{2.3}$$

holds with element-wise inequality. A function where (2.3) holds with strict inequality is a strictly convex function (note, the converse is not true). Examples of convex functions which we will use include:

- linear and affine functions $y = \langle a, x \rangle + b$ for $a, x \in \mathbb{R}^n$ and $b \in \mathbb{R}$,
- (negative) logarithms $-\log x$ for $x \in \mathbb{R}_{++}$,
- L_p norms $\|x\|_p$ for $p \geq 1$ and $x \in \mathbb{R}^n$,
- the max function $\max\{x_1, \ldots, x_n\}$ for $x \in \mathbb{R}$.

2.1.3 Smooth Convex Functions

Let $f : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable, convex function which has a minimizer $x^* \in \mathcal{X}$. We define f to be smooth if the gradient ∇f is β-Lipschitz for $\beta \geq 0$, i.e., $\forall x_1, x_2 \in \mathcal{X}$ we have

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq \beta \|x_1 - x_2\|$$

Such a condition implies that the function does not have sudden increases or decreases. We will use this condition to show bounds on performance in Chapters 5-8.

2.1.4 Non-smooth Convex Functions

In Chapters 5-7, in addition to smooth functions, we will also use non-smooth functions such as the L_1 norm. To define a non-smooth function, we must first define the sub-differential set. The sub-differential set is defined as

$$\partial f(x_1) := \{g \in \mathbb{R}^n : f(x_2) \geq f(x_1) + \langle g, x_2 - x_1 \rangle\} \quad \forall x_2 \in \mathcal{X} \ .$$

The sub-differential set $\partial f(x_1)$ is convex, compact, and each $g \in \partial f(x)$ is a sub-gradient. Note, the condition for set membership is the Taylor expansion of f at x_1 which we saw in the first-order definition of a convex function in (2.2) where $g = \nabla f(x_1)$. Given the sub-differentiable set, a convex function f is non-smooth if $\partial f(x) \neq \emptyset$ for all $x \in \mathcal{X}$. Moreover, if f is differentiable in $\mathcal{X}$ then $\partial f(x) := \{\nabla f(x)\}$. One popular condition for such functions is that the function f is G-Lipschitz in $\mathcal{X}$ if the condition $\|g\| \leq G$ holds for all g. We will use such a condition to show performance bounds in Chapters 5-8.

2.1.5 Strongly Convex Functions

Another class of functions we will use to show strong performance bounds is the class of strongly convex function. A function f is α-strongly convex if it satisfies the inequality

$$f(x_2) \geq f(x_1) + \langle x_2 - x_1, \nabla f(x_1) \rangle + \frac{\alpha}{2}\|x_2 - x_1\|_2^2$$

We can see that a strongly convex function is lower bounded by a positive quadratic function at each point in its domain. Note, a strongly convex function is also strictly convex but a strictly convex function is not necessarily strongly convex. Another useful definition of a strongly convex function is the following. A function f is strongly convex if the function is twice differentiable and its Hessian is bounded away from zero, i.e., $\nabla^2 f(x) \geq \alpha \mathbb{I}_{n\times n}$ for $\alpha > 0$ where the inequality is taken element-wise and $\mathbb{I}_{n\times n}$ is the n-dimensional identity matrix. This implies that the minimum eigenvalue of the Hessian matrix $\lambda_{\min} \geq \alpha$ for all $x \in \mathcal{X}$. We will use strongly convex functions in Chapters 5-10.

2.1.6 Bregman Divergences

We introduce a popular distance function called the Bregman divergence [21]. Let $\psi : \mathcal{X} \to \mathbb{R}$ be a strictly convex function defined on a convex set $\mathcal{X} \subseteq \mathbb{R}^p$ such that ψ is differentiable on the non-empty relative interior of $\mathcal{X}$. The Bregman divergence $d_\psi : \mathcal{X} \times \text{relint}(\mathcal{X}) \to [0, \infty)$ is defined as

$$d_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle \ . \tag{2.4}$$

A couple useful properties of Bregman divergences we will use are that they are non-negative and convex in the first argument. The following are examples of Bregman divergences.

- For $\psi(x) = \|x\|_2^2$ then $d_\psi(x, y) = \|x - y\|_2^2$ is the squared Euclidean distance.
- For $\psi(x) = x^\top Dx$ where D is a positive definite matrix then $d_\psi(x, y) = (x - y)^\top D(x - y)$ is the Mahalanobis distance when A is the inverse covariance matrix.
- For $\psi(x) = \sum_{i=1}^n x(i) \log(x_i)$ then $d_\psi(x, y) = \sum_{i=1}^n x(i) \log\left(\frac{x(i)}{y(i)}\right)$ is the KL divergence.

2.1.7 Alternating Direction Method of Multipliers

Now that we have reviewed convex sets, convex functions, and Bregman divergences, we will turn to a method for solving convex optimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m \tag{2.5}$$

where the goal is to find an x which minimizes the objective function $f(x)$ subject to the constraints that $f_i(x) \leq b_i \ \forall i$ must be satisfied. We will encounter similar convex optimization problems throughout this thesis. In particular, each of our algorithms in Chapters 5-7 involve a complicated convex optimization problem involving both smooth and non-smooth terms. Methods which can divide such an optimization problem into a series of smaller problems each which is easier to solve or can be solved in parallel is useful, particularly in practice. One such method which has become popular in recent years is the Alternating Direction Method of Multipliers (ADMM) [19]. ADMM is an efficient, distributed optimization method closely related to Bregman iterative algorithms for L_1 problems and proximal point methods [105]. It has been applied in many real-world problems because of its computational benefits and fast convergence in practice. ADMM solves problems of the form

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x) + g(z) \quad \text{s.t.} \quad Ax + Bz = c$$

where $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, $c \in \mathbb{R}^p$, and f and g are convex functions. ADMM splits the typical optimization problem $f(x)$ as in (2.5) into a composite objective function using two primal variables x and z since it allows for decomposability and parallelization. The first step in the method is to form the augmented Lagrangian by moving the constraints to the objective function and scaling them by a dual variable y and adding a similar

quadratic term which provides robustness as

$$L_\rho(x, z, y) = f(x) + g(z) + \langle y, Ax + Bz - c\rangle + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \tag{2.6}$$

for $\rho > 0$. Using the augmented Lagrangian, ADMM decomposes the problem into the following sub-problems which are solved iteratively for $k = 1, 2, \ldots$ until convergence.

$$x^{k+1} := \text{argmin}_x \ f(x) + \langle y^k, Ax + Bz^k - c\rangle + \frac{\rho}{2}\|Ax + Bz^k - c\|_2^2, \tag{2.7}$$

$$z^{k+1} := \text{argmin}_z \ g(z) + \langle y^k, Ax^{k+1} + Bz - c\rangle + \frac{\rho}{2}\|Ax^{k+1} + Bz - c\|_2^2, \tag{2.8}$$

$$y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \ . \tag{2.9}$$

ADMM thus involves a primal variable x minimization step, a primal variable z minimization step, and a closed form gradient ascent update step for the dual variable y with a step size of ρ. Notice, since ADMM decomposes the problem and optimizes over the variables separately, each sub-problem only needs to involve terms related to the optimization variable, which simplifies each of the sub-problems. Moreover, the x-update (2.7) and the z-update (2.8) involve alternating between solving for one primal variable while holding the other variables fixed.

If we combine the linear and quadratic terms in each of the sub-problems and scale the dual variable as $u = \frac{1}{\rho}y$ we get a more convenient way of writing the sub-problems

$$x^{k+1} := \text{argmin}_x \ f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|_2^2, \tag{2.10}$$

$$z^{k+1} := \text{argmin}_z \ g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|_2^2, \tag{2.11}$$

$$u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c \ . \tag{2.12}$$

Notice, the x-update and z-updates can be viewed as proximal operators [105] so if there exists efficient algorithms to solve the proximal operators then ADMM will be efficient. Further, when ADMM solves the above problems in sequence it is guaranteed to converge to the optimal solution at a rate of the order $\frac{1}{T}$ where T is the number of iterations under reasonable conditions [19, 124]. Our algorithms in Chapters 5-7 will involve solving similar ADMM sub-problems.

2.2 Online Convex Optimization

Given that we have reviewed enough background material, we are ready to introduce the main problem setting. In this thesis, we consider the standard online learning setting [110] which can be considered as a game between two players: the algorithm and Nature. The game proceeds in rounds $t = 1, \ldots, T$ where at each round t the algorithm selects an action x_t from some action or decision set $\mathcal{X}$, Nature selects a scalar-valued loss function $\ell_t(\cdot)$, the algorithm receives some feedback $\mathfrak{F}_t$, and the algorithms suffers a loss of $\ell_t(x_t)$. Note, the setting is quite general and, in particular, we have not specified *how* Nature selects the loss function. We do not make any statistical assumptions regarding how the losses are selected. In fact, we allow Nature to select the losses adversarially in response to what action the algorithm selected. The goal of the algorithm is to minimize its cumulative loss over the T rounds $\min_{x_1,\ldots,x_T} \sum_{t=1}^{T} \ell_t(x_t)$. However, given the possibly adversarial sequence of loss functions, we cannot hope to minimize the cumulative loss in general. Instead, we will measure the performance of the algorithm using the regret which will be discussed in the next section.

The setting naturally models numerous real-world problems. For example, in user movie recommendations, an algorithm must select a movie x_t to recommend the user from a set $\mathcal{X}$ of movies available after which the user either watches the movie or not thereby providing the algorithm with a loss of $\ell_t(x_t) = 0$ or $\ell_t(x_t) = 1$ respectively. In this problem, the feedback the algorithm receives is only whether the user watched the movie recommended, i.e., $\mathfrak{F}_t = \{\ell_t(x_t)\}$. The algorithm must use such (bandit) feedback to decide what movie to recommend next. Other real-world problems can similarly be modeled using the online learning setting such as medical treatments, online advertising, robot exploration, and algorithmic trading.

The online learning setting is general enough to be useful for many applications however, in order to provide theoretical guarantees on performance and computationally efficient algorithms we need to make additional assumptions on various aspects of the setting. The most common set of assumptions which we use in this thesis are that the set of actions and loss functions are both convex which gives rise to the popular Online Convex Optimization (OCO) setting [110, 130] shown in Figure 2.1.

For $t = 1, \ldots, T$

1. Algorithm selects an action x_t from a convex set $\mathcal{X}$
2. Nature selects a convex loss function $\ell_t : \mathcal{X} \to \mathbb{R}$
3. Algorithm receives feedback $\mathfrak{F}_t$
4. Algorithm suffers a loss of $\ell_t(x_t)$

Figure 2.1: Online Convex Optimization.

2.3 Regret

Since Nature is allowed to select the sequence of loss functions in an adversarial way, one cannot hope to minimize the cumulative loss in general. Instead, we will measure the performance of the algorithm with respect to a comparator class where, after the algorithm is finished playing the game, we can compute the cumulative loss of each member of the class using the same sequence of loss functions. Therefore, if Nature selects a difficult sequence of loss functions then any member of the class will perform poorly and our goal is to perform not much worse than the best member in the class.

The performance measure is called the *regret* and we consider two classes of comparators: fixed solutions $x \in \mathcal{X}$ and any sequence of solutions $\{x_1, \ldots, x_T\}$ such that each $x_i \in \mathcal{X}$ for $i = 1, \ldots, T$. Under these two comparator classes we consider three types of regret in this thesis: fixed regret [66], shifting regret [70], and pseudo regret [24]. The fixed regret measures the algorithm's realized performance against the comparator class of fixed solutions and is formally defined as

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x^*) \ . \tag{2.13}$$

The shifting regret measures the algorithm's realized performance against the comparator class of time-varying solutions and is formally defined as

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{x_1^*, \ldots, x_T^* \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x_t^*) \ . \tag{2.14}$$

Finally, the pseudo regret measures the algorithm's expected performance against the expected performance of the comparator class of fixed solutions where the expectation is taken with respect to the internal randomness of the algorithm and the losses suffered and is formally defined as

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t)\right] - \min_{x^* \in \mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x^*)\right] . \tag{2.15}$$

We consider the fixed and shifting regret in the first half of this thesis under full information feedback and the pseudo regret in the second half of this thesis under bandit information feedback. The shifting regret is the strongest form of regret, the fixed regret follows, and the pseudo regret is the weakest since we only require the algorithm to be competitive with the best fixed solution *in expectation*. We require the algorithm to have sublinear regret with respect to the time horizon, i.e., $R_T = o(T)$, which implies the average regret goes to zero as the number of rounds T goes to infinity. A sublinear regret also implies the algorithm is performing almost as well as the best member from the comparator class which is all we can hope for when Nature is allowed to adversarially select the loss functions. Typical regret bounds are of the form $O(\sqrt{T})$ and $O(\log T)$.

2.4 Feedback

In addition to requirements on the algorithm's performance as measured by the regret, we also consider two types of feedback: full information and bandit information. With full information feedback, the algorithm is able to observe the loss function value for each action it could have chosen: $\mathfrak{F}_t := \{\ell_t(x)\ \forall x \in \mathcal{X}\}$ or, similarly, it has access to the gradient of the loss function at the action selected: $\mathfrak{F}_t := \{\nabla \ell_t(x_t)\}$. A typical real-world example when such feedback is available is a horse race. A gambler will place a bet on a horse, watch the race, and at the end can observe how each horse placed and the payoffs attributed to each horse. Full information feedback is the standard type of feedback considered in online learning. Algorithms which have access to such feedback tend to have sharp regret bounds [110] since learning in such settings can be considered easier than settings with less informative feedback. Full information feedback allows the algorithm to observe how it could have performed had it selected a different action and select a really good action in the next round.

A more difficult setting in which to learn is with bandit information feedback [24]. With bandit feedback, the algorithm is only able to observe the loss function value of the action selected: $\mathfrak{F}_t := \{\ell_t(x_t)\}$. For example, a medical practitioner can only observe how a patient's health changed for the treatment administered. Bandit information feedback provides the algorithm with less information with which to base its next action on compared to full information feedback. As such, learning with bandit feedback is more difficult and it has been shown that there is a price one pays when learning with bandit feedback in the form of a gap between the optimal regret achievable with bandit feedback compared to with full information feedback [38].

2.5 Structure

To compute efficient, high quality solutions we use the structure of the solutions and user models. We consider vectors and matrices to be structured if they have a small value according to some norm. In other words, we only consider types of structure which can be captured by a norm. Common types of structure include: sparse, group sparse, and low-rank which can be captured by the L_1, $L_{(1,2)}$, and nuclear norms defined as follows. Note, if a vector has a small L_2 norm we consider it to be unstructured.

Sparsity with L_1 Norm

The L_1 norm of a vector x is the sum of absolute values of the vector elements: $\|x\|_1 = \sum_{i=1}^{n} |x(i)|$. The L_1 norm is used often in machine learning to encourage sparse solutions, i.e., solutions with few non-zero elements[1]. For example, if we want to compute a sparse estimate of the coefficients of a linear function then we can solve the Lasso regression problem which uses the squared loss with the L_1 norm as a regularizer as

$$\hat{\theta} := \operatorname*{argmin}_{\theta \in \mathbb{R}^p} \frac{1}{n} \|y - X\theta\|_2^2 + \lambda \|\theta\|_1$$

where $y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n \times p}$ is the design matrix, $\theta \in \mathbb{R}^p$ is the sparse optimization parameter, and $\lambda > 0$ is the regularization parameter.

[1] Note, it would be more accurate to consider the L_0 norm which simply counts the number of non-zero elements. However, the L_0 norm is not convex and since we follow the OCO model we can only consider convex functions so we instead use the L_1 norm as a proxy since it is convex.

Group Sparsity with $L_{(1,2)}$ Norm

Let $\mathcal{G}$ be a set of K groups such that each $g_i \in \mathcal{G}$ is defined as $g_i \subseteq \{1, \ldots, p\}$ for $i = 1, \ldots, K$. The $L_{(1,2)}$ norm of a vector x is the sum of the L_2 norm of each sub-vector x_{g_i} which is equal to x for those indices in the set g_i and 0 otherwise, i.e., $\|x\|_{(1,2)} = \sum_{i=1}^{K} w_{g_i} \|x_{g_i}\|_2$ where $w_{g_i} \geq 0$ is the weight for group g_i. The $L_{(1,2)}$ norm is used to encourage sparsity over the groups rather than sparsity over the elements as in the L_1 norm. It is used in the group lasso regression problem

$$\hat{\theta} := \operatorname*{argmin}_{\theta \in \mathbb{R}^p} \frac{1}{n} \|y - X\theta\|_2^2 + \lambda \|\theta\|_{(1,2)} \ .$$

Low-rank with Nuclear Norm

Let $\Theta \in \mathbb{R}^{d \times p}$ be a matrix with rank $r \leq \min\{d, p\}$. The nuclear norm is the sum of the first r singular values σ_i (in descending order) of Θ: $\|\Theta\|_* = \sum_{i=1}^{r} \sigma_i$. The nuclear norm is used to compute low-rank approximations, for example, by solving a nuclear-norm regularized least squares problem as

$$\hat{\Theta} := \operatorname*{argmin}_{\Theta \in \mathbb{R}^{d \times p}} \frac{1}{n} \sum_{i=1}^{n} \Big(y_i - \operatorname{trace}(X_i^\top \Theta) \Big)^2 + \lambda \|\Theta\|_*$$

where $y_i \in \mathbb{R}$ is a response variable and $X_i \in \mathbb{R}^{d \times p}$ is a sample.

2.6 Algorithmic Frameworks

We will use two algorithmic frameworks on which we design our algorithms, one for the first half of the thesis where we consider full information feedback and another in the second half where we consider bandit information feedback.

Full Information Framework

The first framework on which we design our full information algorithms in Chapters 5-8 is motivated from the ideal scenario in the OCO setting where the algorithm would like to select an action x_{t+1} which minimizes the next loss function $\ell_{t+1}(\cdot)$, i.e.,

$$x_{t+1} := \operatorname*{argmin}_{x \in \mathcal{X}} \ell_{t+1}(x) \ .$$

However, given the online nature of the problems we consider, where the action x_t must be selected before the loss $\ell_t(x_t)$ is suffered, such minimization is not possible. Instead, a common approach is to minimize the previous loss instead

$$x_{t+1} := \operatorname*{argmin}_{x \in \mathcal{X}} \ell_t(x) \ .$$

It is reasonable to consider such a problem if we make some assumptions on the loss functions. One common assumption is to assume the losses are Lipschitz continuous

$$\|\ell_t(x_1) - \ell_t(x_2)\|_2 \leq G\|x_1 - x_2\|_2$$

for all $x_1, x_2 \in \mathcal{X}$ where $G > 0$ is the Lipschitz constant which limits how fast the loss function can change. Note, as we mentioned in Section 2.1.4, a function is G-Lipschitz if its (sub)gradients $g \in \partial\ell(x)$ are bounded as $\|g\| \leq G$ which implies the loss is Lipschitz continuous. When the losses are Lipschitz continuous, we can often show regret bounds when selecting an action by minimizing the previous loss function.

Further, it is common to linearize the loss function at the previous solution x_t to make it computationally easier to solve the optimization problem

$$x_{t+1} := \operatorname*{argmin}_{x \in \mathcal{X}} \ell_t(x_t) + \langle \nabla \ell_t(x_t), x - x_t \rangle + d_\psi(x, x_t) \ .$$

In doing so, we must add a proximal term (or distance function) $d_\psi(x, x_t)$ which penalizes for selecting an x which is far away from x_t so as to prevent following the linear approximation and getting too far away from the true loss function.

Finally, in this thesis we take advantage of a priori structure in the solution or user model as introduced in Section 2.5. We can encode such structure in the algorithm by adding a norm penalty function $R(\cdot)$ such as $R(x) = \|x\|_1$ to the optimization problem to induce such structured solutions or user models as

$$x_{t+1} := \operatorname*{argmin}_{x \in \mathcal{X}} \ell_t(x_t) + \langle \nabla \ell_t(x_t), x - x_t \rangle + \lambda R(x) + d_\psi(x, x_t) \ . \tag{2.16}$$

where $\lambda \geq 0$ is a parameter which balances minimizing the loss function and fitting to the structure. Note, (2.16) is know as the composite objective mirror descent (COMID) problem which we will discuss in Section 3.1.5.

Bandit Information Framework

The second framework on which we design our bandit information algorithms in Chapters 9 and 10 relies on the *optimism-in-the-face-of-uncertainty* (OFU) principle [24] which states that when one must select an action from a number of actions each with an uncertain reward (loss) that one should select the action which has the largest (smallest) potential reward (loss). The OFU principle is well-known in the bandit literature and one common approach is to compute the expected reward $\hat{\mu}(x_i)$ and a confidence interval $\sigma(x_i)$ for each action $x_i \in \mathcal{X}$ then select the action via

$$\operatorname*{argmin}_{x \in \mathcal{X}} \hat{\mu}(x) + \sigma(x) \ .$$

We will follow such a framework in Chapters 9 and 10 where we make assumptions about the expected loss, for example, that is a linear function parameterized by an unknown and structured variable, and construct confidence sets whose size depends on the structure of the unknown variable.

Chapter 3

Related Work

In this chapter, we will discuss the work related to the research presented in this thesis. We organize the related work into two sections: works with full information feedback and works with bandit information feedback. Within each section, we will present the work chronologically. We will discuss works which are most relevant to the research presented in this thesis however, this chapter is meant only as a brief review of the literature and to provide context to place our contributions.

3.1 Full Information

The standard online learning setting [28, 110] considers full information feedback and has roots that trace back to non-cooperative game theory [56] and the minimax theorem [122]. The minimax theorem was one of the first attempts at showing performance guarantees similar to regret. Such guarantees were made more clear by the work of James Hannan in 1957 when he introduced a modern measure of performance called Hannan consistency [66]. An algorithm is Hannan consistent if the average regret goes to zero in the limit. Outside of the game theory community, Frank Rosenblatt introduced the Perceptron algorithm [107] which is a linear classifier that sequentially classifies data points. The Perceptron algorithm was one of the first algorithms to consider online learning for classification.

3.1.1 Weighted Majority Algorithm and Expert Prediction

Online learning and regret were further considered in binary prediction settings where the algorithm has access to a set of N experts each which make a binary prediction of some event. The idea is to design an algorithm which will use the experts' predictions to form its own prediction such that the number of incorrect predictions, i.e., mistakes, is small. One of the simplest algorithms introduced in the 1970s and late 1980s for such a problem is the Halving algorithm [5, 92] which works by predicting according to a majority vote among the experts' predictions. After each round of prediction, the Halving algorithm removes those experts who made an incorrect prediction from the set of experts and the next vote is among only those left in the set. At most half of the experts will be removed from the set in each round, hence the name the Halving algorithm. If there exists at least one expert who never makes a mistake, this expert can be identified after at most $\log_2 N$ rounds so the regret will never grow above $O(\log N)$.

The Halving algorithm makes a strong assumption that there exists at least one expert who will never make a mistake which is unrealistic in most scenarios. For such scenarios, Littlestrong and Warmuth introduced the Weight Majority (WM) [93] algorithm in 1994. WM maintains a weight for each expert and if an expert makes a mistake then its weight is decreased by a factor $\beta \in (0, 1]$. In each round, WM calls a vote and each experts' prediction is adjusted by their weight therefore, badly performing experts' predictions contribute less to the overall prediction vote. WM then predicts by selecting the prediction which has highest aggregate weight. Therefore, WM will be able to identify good performing experts however, it never completely eliminates experts from the set of experts since their weight never reaches zero. It can be similarly shown that the regret of WM is of the order $\log N$ even without the assumption of a perfect expert.

The Halving algorithm and Weighted Majority algorithm fall under the framework of Prediction with Expert Advice (PWEA) [55, 56, 57, 82, 123, 27, 58, 60] which saw a surge of interest throughout the 1990s. One of the main ideas in PWEA is to maintain a weight vector $\mathbf{w}_t \in \mathbb{R}^N_{++}$ over the N experts and multiplicatively update the weights after each round: $w_{t+1}(i)=w_t(i)\beta^{\ell_t(w_t(i))}$ where $\beta>0$ and $\ell_t(w_t(i))$ is the loss for expert i. The multiplicative update decreases the weights of those experts which incur non-zero loss and does not change the weights for those experts that incur zero loss. Such multiplicative update algorithms tend to learn exponentially fast which experts are best.

3.1.2 Online Gradient Descent and Exponentiated Gradient

Instead of a multiplicative update, one may also consider an additive update where the weights change as $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta_t \mathbf{u}$. The additive update moves the weight vector in the direction $\mathbf{u}$ scaled by the step size η_t where $\mathbf{u}$ is designed to decrease the algorithm's loss at the next round. Often, the negative gradient of the loss function, i.e., $\mathbf{u} = -\nabla \ell_t(\mathbf{w}_t)$, is used since it points in the direction of steepest descent which gives

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell_t(\mathbf{w}_t) \tag{3.1}$$

which is the well-known online Gradient Descent algorithm.

The multiplicative and additive updates may seem different but it was shown by Kivinen and Warmuth [82] in 1997 that the two updates can be derived from a common framework. The framework introduced is the optimization problem

$$\mathbf{w}_{t+1} := \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^p} \ell_t(\mathbf{w}) + R(\mathbf{w}, \mathbf{w}_t) \tag{3.2}$$

where $\ell_t(\cdot)$ is a linear function and $R(\mathbf{w}, \mathbf{w}_t)$ is a function which measure the distance between a new weight vector $\mathbf{w}$ and the previous weight vector $\mathbf{w}_t$ such as squared Euclidean, relative entropy, etc.

They showed that one can derive online Gradient Descent in (3.1) by solving (3.2) with $R(\mathbf{w}, \mathbf{w}_t) = \frac{1}{2\eta_t} \|\mathbf{w} - \mathbf{w}_t\|_2^2$ where $\eta_t > 0$ is the learning rate as

$$\begin{aligned} \mathbf{w}_{t+1} &:= \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^p} \ell_t(\mathbf{w}) + \frac{1}{2\eta_t} \|\mathbf{w} - \mathbf{w}_t\|_2^2 \\ &= \mathbf{w}_t - \eta_t \nabla \ell_t(\mathbf{w}_t) \ . \end{aligned} \tag{3.3}$$

However, if we instead set $R(\mathbf{w}, \mathbf{w}_t) = \frac{1}{2\eta_t} \sum_{i=1}^p w(i) \log\left(\frac{w(i)}{w_t(i)}\right)$ and solve we get

$$\begin{aligned} \mathbf{w}_{t+1} &:= \operatorname*{argmin}_{w \in \mathbb{R}_+^p} \ell_t(\mathbf{w}) + \frac{1}{2\eta_t} \sum_{i=1}^p w(i) \log\left(\frac{w(i)}{w_t(i)}\right) \\ \Rightarrow w_{t+1}(i) &= \frac{w_t(i) \exp(-\eta_t \nabla \ell_t(\mathbf{w}_t)_i)}{\sum_{i=1}^p w_t(i) \exp(-\eta_t \nabla \ell_t(\mathbf{w}_t)_i)} \quad \text{for } i = 1, \ldots, p \end{aligned} \tag{3.4}$$

which is a multiplicative update similar to WM and other PWEA algorithms which they called the Exponentiated Gradient (EG). The EG algorithm has been used in the

context of online portfolio selection [69] which is the application domain we use for our algorithms and which we will formally introduce in Chapter 4.

Note, the framework in (3.2) can be used without a linear loss function if the proximal operator [105] can be computed efficiently which provides a more modern viewpoint. The proximal operator is defined as

$$\text{prox}_{\ell_t}(\mathbf{v}) := \underset{\mathbf{x}}{\text{argmin}}\ \ell_t(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 \ . \tag{3.5}$$

For example, if $\ell_t(\mathbf{x}) = \|\mathbf{x}\|_1$ then the proximal operator is a projection to the L_1 ball which can be efficiently computed with the soft thresholding operator [105]. Numerous other proximal operators have efficient computations however, as we will see in Chapter 7, many are still open problems.

3.1.3 Online Convex Optimization

Building on the work of Kivinen and Warmuth [82] a paper by Zinkevich [130] presented an analysis of online Gradient Descent for any general G-Lipschitz (bounded gradients) convex loss functions rather than only linear loss functions. Further, Zinkevich initialized the work in online optimization with convex functions named Online Convex Optimization (OCO) (previously introduced in Figure 2.1). Zinkevich showed for the OCO setting if the algorithm selects a $\mathbf{w}_t$ via online Gradient Descent (3.3) that the regret for any general convex loss function (with bounded gradients) incurs a regret at most $O(\sqrt{T})$. The general OCO setting is applicable in numerous applications and has gained significant popularity since Zinkevich introduced it in 2003 and is the setting we consider in the first half of this thesis.

3.1.4 Follow-The-Regularized-Leader

A common and intuitive framework often used within the OCO setting for computing the next weight vector is to find the vector which minimizes all previous losses

$$\mathbf{w}_{t+1} := \underset{\mathbf{w}\in\mathbb{R}^p}{\text{argmin}} \sum_{\tau=1}^{t} \ell_\tau(\mathbf{w}) \ . \tag{3.6}$$

Such an framework is called Fictitious Play [22] in economics and game theory and in machine learning it has been called Follow-The-Leader (FTL) [79]. It can FTL

suffers linear regret [110] for linear loss functions because the weight vector can change dramatically from one round to the next which Nature can take advantage of. One can avoid such volatile changes by introducing a scalar penalty function $R(\cdot)$ which penalizes changing the weight vector and instead solve

$$\mathbf{w}_{t+1} := \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^p} \sum_{\tau=1}^{t} \ell_\tau(\mathbf{w}) + R(\mathbf{w}) \ . \tag{3.7}$$

Such a problem was coined Follow-The-Regularized-Leader (FTRL) by Shalev-Shwartz and Singer in 2007 however, algorithms solving such a problem were first studied in the early 2000s by Grove et al. [64] and Kivinen and Warmuth [83]. It can be shown [110] that when one solves (3.7) with $R(\mathbf{w}) = \frac{1}{2\eta_t}\|\mathbf{w}\|_2^2$ one obtains

$$\mathbf{w}_{t+1} = \sum_{\tau=1}^{t} \eta_t \nabla \ell_t(\mathbf{w}) = \mathbf{w}_t - \eta_t \nabla \ell_t(\mathbf{w}_t)$$

which is the online Gradient Descent algorithm in (3.3). Similarly, if one sets $R(\mathbf{w}) = \frac{1}{2\eta_t}\sum_{i=1}^{p} w(i)\log(w(i))$ one obtains the EG algorithm as in (3.4). The interesting thing to note here is that there are two optimization problems from (3.2) and (3.7) respectively

$$\mathbf{w}_{t+1} := \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^p} \ell_t(\mathbf{w}) + R(\mathbf{w}, \mathbf{w}_t), \tag{3.8}$$

$$\mathbf{w}_{t+1} := \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^p} \sum_{\tau=1}^{t} \ell_\tau(\mathbf{w}) + R(\mathbf{w}) \tag{3.9}$$

and the solutions to such problems can be made equivalent by carefully selecting $R(\cdot)$ and $R(\cdot,\cdot)$. To make this clear, recall the Bregman divergence (Section 2.1.6) defined with respect to a strictly convex function ψ as

$$d_\psi(\mathbf{w}, \mathbf{w}_t) = \psi(\mathbf{w}) - \psi(\mathbf{w}_t) - \langle \nabla\psi(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle \ . \tag{3.10}$$

If we set the function $R(\mathbf{w})$ in (3.9) as $R(\mathbf{w}) = \|\mathbf{w}\|_2^2$ and solve we obtain the same solution (online Gradient Descent) as if we set the function $R(\mathbf{w}, \mathbf{w}_t)$ in (3.8) as a Bregman divergence with respect to $\psi(\mathbf{w}) = R(\mathbf{w}) = \|\mathbf{w}\|_2^2$ which gives

$$\begin{aligned} d_\psi(\mathbf{w}, \mathbf{w}_t) &= \|\mathbf{w}\|_2^2 - \|\mathbf{w}_t\|_2^2 - \langle 2\mathbf{w}_t, \mathbf{w} - \mathbf{w}_t \rangle \\ &= \|\mathbf{w} - \mathbf{w}_t\|_2^2 = R(\mathbf{w}, \mathbf{w}_t) \ . \end{aligned}$$

Similarly, setting $R(\mathbf{w}) = \sum_{i=1}^{p} w(i)\log(w(i))$ gives the same solution, the EG algorithm, as if we set $R(\mathbf{w}, \mathbf{w}_t)$ as a Bregman divergence with respect to $\psi(\mathbf{w}) = R(\mathbf{w}) = \sum_{i=1}^{p} w(i)\log(w(i))$ which gives

$$\begin{aligned} d_\psi(\mathbf{w}, \mathbf{w}_t) =& \sum_{i=1}^{p} w(i)\log(w(i)) - \sum_{i=1}^{p} w_t(i)\log(w_t(i)) - \sum_{i=1}^{p}(w(i) - w_t(i))(\log w_t(i) + \log e) \\ &= \sum_{i=1}^{p} w(i)\log\left(\frac{w(i)}{w_t(i)}\right) = R(\mathbf{w}, \mathbf{w}_t) \ . \end{aligned}$$

3.1.5 Online Mirror Descent and Composite Objective Mirror Descent

As we just saw, one can interpret FTRL (3.7) as a version of (3.2) when the regularizer is a Bregman divergence as

$$\mathbf{w}_{t+1} := \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}}\ \ell_t(\mathbf{w}) + d_\psi(\mathbf{w}, \mathbf{w}_t) \ . \tag{3.11}$$

Further, for linear loss functions it can be shown that the solutions of FTRL and (3.11) are equivalent [68, 110, 67]. Therefore, if we linearize the convex loss function we get

$$\mathbf{w}_{t+1} := \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}}\ \eta_t\langle\nabla\ell_t(\mathbf{w}), \mathbf{w}\rangle + d_\psi(\mathbf{w}, \mathbf{w}_t) \tag{3.12}$$

which is called Online Mirror Descent (OMD) [103, 12] which has the general solution

$$\mathbf{w}_{t+1} = \nabla\psi^*\left(\nabla\psi(\mathbf{w}_t) - \eta_t\nabla\ell_t(\mathbf{w}_t)\right) \tag{3.13}$$

where $\eta_t \geq 0$ is the step size and $\psi^*(\cdot)$ is the dual norm of $\psi(\cdot)$. For general convex loss function, OMD suffers a regret of $O(\sqrt{T})$ and for strongly convex loss functions suffers a regret of $O(\log T)$. In 2010, Duchi et al. generalized OMD to composite loss functions: $f_t = \ell_t + \Omega$ where ℓ_t is a convex loss function which can change with time and Ω is a fixed convex function. The Composite Objective Mirror Descent (COMID) [48] problem is

$$\mathbf{w}_{t+1} := \underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}}\ \eta_t\langle\nabla\ell_t(\mathbf{w}_t), \mathbf{w}\rangle + \lambda_t\Omega(\mathbf{w}) + d_\psi(\mathbf{w}, \mathbf{w}_t) \ . \tag{3.14}$$

Note, the loss function is again linearized however, the function Ω is not linearized. Often one wants to induce structure in the solutions and linearizing the objective function can lose such structure. Therefore, using COMID we can induce structure through the

function Ω where the structure is maintained since Ω is not linearized. When the losses ℓ_t are Lipschitz the regret is $O(\sqrt{T})$ and when $\ell_t + \Omega$ are strongly convex functions the regret is $O(\log T)$. Note, for the online setting COMID is known originally as the (Fast) Iterative Shrinkage-Thresholding Algorithm (ISTA/FISTA) [29, 53, 43].

Duchi et al. only showed the fixed regret (2.13) of COMID. In this thesis, we will make two contributions in Chapters 5 and 6 which will be two algorithms which build on COMID and extend the regret analysis to include bounds on shifting regret (2.14) when the composite objective includes a non-smooth term. Further, in Chapter 4 we will introduce the online portfolio selection problem and work related to research along this thread. In Chapters 5-8, we will again build on COMID now by inducing structured solutions by carefully selecting the function Ω. We make a number of practical contributions to the online portfolio selection literature using such structure.

3.2 Bandit Information

With bandit information, the feedback the algorithm receives is only the reward[1] function value for the action taken $r_t(x_t)$. There are many types of bandit problems which depend on the process generating the reward (stochastic or adversarial), the reward function model (probability distribution, linear, non-linear, general convex) and the type of feedback received (binary, real-valued) [24].

3.2.1 Stochastic K-Armed Bandits

The standard problem with bandit feedback is the stochastic K-armed bandit problem which consists of K arms (e.g. actions) each which have a fixed and unknown distribution. In each round t, the algorithm selects an arm $I_t \in \{1, \ldots, K\}$ and a reward $r_t(I_t)$ is drawn i.i.d. from the I_tth distribution. The algorithm observes $r_t(I_t)$ and receives a reward of $r_t(I_t)$. The goal of the algorithm is to maximize its cumulative reward.

The study of the problem began with William Thompson in 1933 [118] for application in experimental design. Thompson considered a Bayesian approach and developed an algorithm, later named Thompson Sampling (TS). TS works by selecting a likelihood

[1] The use of rewards or losses and arms or decisions depends on the community. We will use the terminology which follows from the community and it should be clear from the context what is intended.

function of the reward given a set of parameters, a prior distribution over the parameters, and computes a posterior distribution of the reward according to Bayes rule. Then, for each arm a sample is drawn i.i.d. from the posterior distribution and the arm with the largest sampled reward is selected to be played in the next round. After the reward is observed, a new posterior distribution is computed and another set of samples are drawn. TS can be considered a probability matching algorithm.

TS is one approach to the K-armed bandit problem. Another approached introduced by Lai and Robbins in 1985 [84] uses the *optimism-in-the-face-of-uncertainty* (OFU) principle which states one should select the action with largest potential reward. A general framework which uses the OFU principle was briefly discussed in Section 2.6 and we will follow such a framework in Chapters 9 and 10. More specifically, the framework uses the OFU principle by computing an upper confidence bound (UCB) for each arm which consists of the mean observed reward $\mu_k = \sum_{\tau=1}^{t} r_t(I_t) \cdot \mathbb{1}_{\{I_t=k\}}$, where $\mathbb{1}_{\{I_t=k\}}$ is the indicator function which returns 1 if the condition $\{I_t = k\}$ is true and 0 if not, and a confidence interval σ_k. The algorithm optimistically selects an arm via

$$I_{t+1} := \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} \ \mu_k + \sigma_k \ .$$

Lai and Robbins introduce the UCB1 algorithm in 1985 which computes the mean reward as above and computes the confidence interval using the Chernoff-Hoeffding inequality as $\sigma_k = \sqrt{2 \log t / n_k}$ where t is the current round and n_k is the number of times arm k has been selected. UCB1's regret is $O(K \log T)$.

3.2.2 Contextual and Stochastic Linear Bandits

Building off the standard K-armed bandit problem, one can consider scenarios when a p-dimensional feature vector is provided for each of the K decisions and the expected loss is a linear function of the feature vector and an unknown parameter. Such a problem is called the contextual linear bandit problem and has become popular for online advertising, news article recommendations, and medical treatment applications [24]. Several papers in the 2000s and early 2010s have studied the contextual linear bandit under the OFU principle and have taken a similar UCB approach [7, 90, 32] and shown the regret to be $O(\log^{3/2}(K)\sqrt{p}\sqrt{T})$. However, for some applications such as medical

treatments where a treatment may consist of an infinite number of drug mixtures, the dependence on the number of arms K can be problematic when it is large or infinite.

For such settings, Dani et al. in 2008 studied the stochastic linear bandit problem [38] where the decision set is an arbitrary compact set in $\mathbb{R}^p$. They presented the algorithm ConfidenceBall2 based on the OFU principle which computes an estimate $\hat{\theta}_t$ of the unknown parameter θ^* using ridge regression and constructs an confidence ellipsoid around $\hat{\theta}_t$ with radius $\sqrt{\beta_t}$. Then, x_t and $\widetilde{\theta}_t$ are selected optimistically from the decision set and confidence ellipsoid respectively, such that $\langle x_t, \widetilde{\theta}_t \rangle$ is minimized. They showed how to set β_t such that θ^* stays within the ellipsoid with high-probability for all t and showed a regret[2] of $\widetilde{O}(p\sqrt{T})$. Note, the regret depends on the ambient dimensionality p and not the number of decisions. Further, two papers [109, 1] around 2010 showed how to construct tighter confidence ellipsoids and, in Abbasi-Yadkori et al. [1], such confidence ellipsoids decreased the regret by a $\sqrt{\log T}$ multiplicative factor.

Building on such works which had considered the problem without structural assumptions on θ^*, two papers published simultaneously in 2012 [26, 2] considered the problem where θ^* is s-sparse. [2] followed the same problem setting as [38] and presented a method which can use the predictions of any full information online algorithm with an upper bound on its regret to construct confidence sets. When constructing confidence sets using the algorithm SeqSEW [61] they showed a regret of $\widetilde{O}(\sqrt{spT})$.

[26] does not consider the standard stochastic linear bandit problem. Instead, they define the loss as $\ell_t(x_t) = \langle x_t, \theta^* \rangle + \langle x_t, \eta_t \rangle$ where η_t is i.i.d. whitenoise and assume the decision set is the unit L_2 ball. Their algorithm performs random estimation then uses techniques from compressed sensing to identify the subspace where θ^* lives then runs ConfidenceBall2 where the decision set is a subspace. They show a regret of $\widetilde{O}(s\sqrt{T})$.

Such work shows that when it is know that θ^* is sparse, sharper regret bounds can be shown. This realization motivates the second half of this thesis which we present in Chapter 9 and Chapter 10. In Chapter 9, we consider the stochastic linear bandit problem under structural assumptions on the unknown parameter θ^*. We show sharper regret bounds for any norm structured θ^* not just sparse.

[2] The $\widetilde{O}(\cdot)$ notation selectively hides constant and log factors.

3.2.3 Stochastic Binary and Generalized Linear Model Bandits

Bandit problems have also been considered which do not assume a linear loss function as in the contextual or stochastic linear bandits. Such problems typically consider losses which are found in Generalized Linear Models (GLMs). GLMs have been considered in contextual bandits [89] where they show experimentally the GLM provides improvements over linear models specifically when the losses are binary. Special cases of the GLM setting such as binary losses have been analyze in the standard K-armed bandit problem [80], the finite vector [80] and infinite vector [129] stochastic linear bandit problem where they show matching regret bounds as in the linear setting. The GLM setting has been considered in the finite arm stochastic linear bandit problem [54] however, no previous works have considered GLMs for the standard stochastic linear bandit problem which allows for decision sets with an infinite number of vectors. In Chapter 10 we build on our results in Chapter 9 and consider non-linear loss functions which are used in Generalized Linear Models. We show regret bounds which match the linear loss function setting for any norm structured θ^* and any GLM loss function.

Chapter 4

Online Portfolio Selection

Our full information algorithms in Chapters 5-8 are designed for general resource allocation problems. One instance of such problems, which motivates many of the techniques we use, is the online portfolio selection where one must decide how to distribute wealth across a number of different assets. We use the problem to experimentally demonstrate the effectiveness of our algorithms and formally introduce the problem here.

4.1 Model

The online portfolio selection model is an instance of the Online Convex Optimization (OCO) setting discussed in Section 3.1.3. Specifically, it involves a stock market consisting of n stocks $\{s_1, \ldots, s_n\}$ over a span of T periods where a period can be any amount of time (minute, day, year) and for ease of exposition, we will consider a time period to be a day. Let $x_t(i) = \frac{\text{closing price}}{\text{opening price}}$ denote the *price relative* of stock s_i in day t, i.e., the multiplicative factor by which the price of s_i changes in day t. Hence, $x_t(i) > 1$ implies a gain, $x_t(i) < 1$ implies a loss, and $x_t(i) = 1$ implies the price remained unchanged. We assume all stocks are alive for all T days so that $\infty > x_t(i) > 0 \ \forall \ i, t$.

Let $\mathbf{x}_t = [x_t(1), \ldots, x_t(n)]^\top$ denote the vector of price relatives for day t, and let $\mathbf{x}_{1:t}$ denote the collection of such price relative vectors up to and including day t. A portfolio $\mathbf{p}_t = [p_t(1), \ldots, p_t(n)]^\top \in \mathcal{P}$ on day t must be selected from the convex set $\mathcal{P}$ which is a n-dimensional probability simplex $\mathcal{P} = \Delta_n := \{\mathbf{p} : p(i) \geq 0 \ \forall i, \sum_{i=1}^n p(i) = 1\}$. $\mathbf{p}_t$ is a probability distribution over n stocks which prescribes investing $p_t(i)$ fraction of the

current wealth in stock s_i which implies the portfolio is self-financing and all wealth must be invested in the portfolio at each time period. It also implies only long positions can be held so shorting or buying on margin is not allowed however, we will relax such constraints in Chapter 8. Note, the portfolio $\mathbf{p}_t$ has to be decided before knowing the price relatives $\mathbf{x}_t$ which are only revealed at the end of the day after the market closes.

Under such a model, the multiplicative gain in wealth at the end of day t is $\mathbf{p}_t^\top \mathbf{x}_t$. For a sequence of price relatives $\mathbf{x}_{1:t-1}$ up to day $(t-1)$, the sequential portfolio selection problem in day t is to determine a portfolio $\mathbf{p}_t$ based on past performance of the stocks. At the end of day t, $\mathbf{x}_t$ is revealed and the actual performance of $\mathbf{p}_t$ gets determined by $\mathbf{p}_t^\top \mathbf{x}_t$. Over T periods, for a sequence of portfolios $\mathbf{p}_{1:T}$, the multiplicative and logarithmic gain in wealth are respectively

$$S_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} \left(\mathbf{p}_t^\top \mathbf{x}_t \right) \ ,$$

$$LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \sum_{t=1}^{T} \log \left(\mathbf{p}_t^\top \mathbf{x}_t \right) \ .$$

Following Cover's work in portfolio theory and log-optimal portfolios [33], the goal is to maximize the doubling rate across all rounds of investing. The problem each day then is to compute a log-optimal portfolio $\mathbf{p}_{t+1} = \text{argmax}_{\mathbf{p}} \log(\mathbf{p}^\top \mathbf{x}_t)$. Therefore, in a costless environment, i.e., no transaction costs, we want to maximize $LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T})$ over $\mathbf{p}_{1:T}$. However, we cannot pose it as a batch optimization problem due to the temporal nature of the choices: $\mathbf{x}_t$ is not available when one has to decide on $\mathbf{p}_t$. Further, (statistical) assumptions regarding $\mathbf{x}_t$ can be difficult to make. As such, we will focus on minimizing the fixed regret (2.13) and (in some cases) the shifting regret (2.14) where the convex loss function is the negative logarithmic gain in wealth.

4.2 Algorithms

Algorithms for automatically designing portfolios based on historical stock market data have been extensively investigated in the literature for the past five decades [81, 97, 112, 88]. With the realization that any statistical assumptions regarding the stock market may be inappropriate and eventually counter-productive, over the past two decades,

new methods for portfolio selection have been designed which make no statistical assumptions regarding the movement of stocks [34, 35, 69]. In a well-defined technical sense, such methods are guaranteed to perform competitively with certain families of adaptive portfolios even in an adversarial market. From the theoretical perspective, algorithm design for portfolio selection has largely been a success story [28, 34, 69]. We build on such works and contribute to this line of literature by designing online portfolio selection algorithms which explicitly consider transaction costs in Chapter 5, trading using market sectors in Chapter 6, diversified trading in Chapter 7, and hedged trading in Chapter 8. No previous algorithms have considered such practical aspects and strategies in the online portfolio selection literature. In what follows, we describe a few of the popular and/or benchmark algorithms which we will compete against.

4.2.1 Single Stock and Buy-and-Hold

One of the simplest algorithms one can use is to invest the entire wealth each day into the same single stock s_i. For a portfolio $\mathbf{p}_t$ on day t the trading strategy is $p_t(i) = 1$ for the stock s_i the algorithm wants to invest in and $p_t(j) = 0$ for $j \neq i$. Investing in the same single stock for the entire trading period is not an sophisticated strategy. However, in our experiments, we will show results for the best single stock in terms of the stock which earns the most cumulative wealth over the trading period in hindsight.

A more common trading algorithm is the Buy-and-Hold strategy [28] where at the first day of the trading period, the algorithm distributes its wealth across n stocks according to an initial portfolio $\mathbf{q}$ (i.e., $\mathbf{p}_1 = \mathbf{q}$) after which the algorithm does not trade. The best Buy-and-Hold strategy is to select the best single stock in hindsight. One special case of the Buy-and-Hold strategy which we will compete against is the Uniform Buy-and-Hold (U-BAH) where the initial strategy is a uniform distribution across all stocks $q(i) = \frac{1}{n}$ for $i = 1, \ldots, n$. Such a strategy allows the wealth in the portfolio to naturally follow the performance of the market.

4.2.2 Constantly Rebalanced Portfolio

A Constantly Rebalanced Portfolio (CRP) [28, 36] is a portfolio which is set to a fixed distribution of wealth over the stocks each day. Specifically, given a fixed distribution $\mathbf{q}$,

each day t the portfolio is set as $\mathbf{p}_t = \mathbf{q}$. Compared to the Buy-and-Hold strategy, the CRP strategy requires frequent trading because throughout the day the distribution of wealth in a portfolio will change due price movements. Therefore, in the next day, in order to invest with a portfolio which has the same distribution of wealth $\mathbf{q}$, the investor must trade in order to rebalance the portfolio $\mathbf{p}_{t+1}$ back to the fixed distribution $\mathbf{q}$. One special case of the CRP strategies is the Uniform Constantly Rebalanced Portfolio (U-CRP) where the fixed distribution is the uniform distribution and the portfolio each day then is a uniform portfolio $p_t(i) = \frac{1}{n}\ i = 1, \ldots, n$. For our portfolio selection algorithms, we will measure their performance by the fixed regret (2.13) which is the difference in the total loss of the algorithm and the total loss of the best CRP in hindsight.

4.2.3 Universal Portfolio

Cover presented the Universal Portfolios (UP) algorithm in [34]. Let $\mathbf{q}$ be a CRP, UP maintains a distribution $\mu(\mathbf{q})$ over all CRPs in the n-dimensional probability simplex and computes a new portfolio by observing the price relative $\mathbf{x}_t$ and performing a Bayesian update. Let $S_{t-1}(\mathbf{q}, x_{1:t-1})$ be the cumulative wealth of CRP $\mathbf{q}$ over a period of $t-1$ days. UP computes a portfolio $\mathbf{p}_t$ which is a weighted average over all CRPs as

$$p_t(i) = \frac{\int_{\Delta_n} q(i) S_{t-1}(\mathbf{q}, \mathbf{x}_{1:t-1}) \mu(\mathbf{q}) \mathrm{d}\mathbf{q}}{\int_{\Delta_n} S_{t-1}(\mathbf{q}, \mathbf{x}_{1:t-1}) \mu(\mathbf{q}) \mathrm{d}\mathbf{q}} \quad \text{for } i = 1, \ldots, n\ .$$

UP has a fixed regret of $O(\log T)$ however, one downside of UP is that it is computationally demanding and often infeasible in practice.

4.2.4 Exponentiated Gradient

We have already introduced the EG algorithm [69] in (3.4) which was motivated from the problem:

$$\mathbf{p}_{t+1} = \operatorname*{argmin}_{\mathbf{p}} \ell_t(\mathbf{p}) + R(\mathbf{p}, \mathbf{p}_t)\ .$$

When setting the regularizer to the KL divergence $R(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2\eta_t} \sum_{i=1}^{n} p(i) \log\left(\frac{p(i)}{p_t(i)}\right)$ we get the multiplicative update

$$w_{t+1}(i) = \frac{w_t(i) \exp(-\eta_t \nabla \ell_t(\mathbf{p}_t) i)}{\sum_{i=1}^{n} w_t(i) \exp(-\eta_t \nabla \ell_t(\mathbf{p}_t) i)} \quad \text{for } i = 1, \ldots, n\ .$$

For application in the online portfolio selection problem, we use the negative logarithmic gain in wealth $\ell_t(\mathbf{p}_t) = -\log(\mathbf{p}_t^\top \mathbf{x}_t)$ as the loss function which gives

$$p_{t+1}(i) = \frac{p_t(i) \exp\left(\eta_t \frac{x_t(i)}{\mathbf{p}_t^\top \mathbf{x}_t}\right)}{\sum_{i=1}^n p_t(i) \exp\left(\eta \frac{x_t(i)}{\mathbf{p}_t^\top \mathbf{x}_t}\right)} \ .$$

In contrast to the UP algorithm, EG is computationally efficient with cost linear in the number of stocks n. Its regret can be shown to be $O(\sqrt{T})$ under the assumption that the price relatives are bounded away from zero $x_t(i) > 0 \ \forall i, t$ which implies it is competitive with the best CRP in hindsight however, it is worse than the $O(\log T)$ regret of UP.

4.3 Datasets

We perform experiments using 5 datasets consisting of price relatives from different markets, periods of time, and durations. The datasets were constructed by taking stock market prices and computing the price relatives from the following markets: Dow Jones Industrial Average (DJIA), New York Stock Exchange (NYSE), Standard & Poor's 500 (S&P 500), and the Toronto Stock Exchange (TSX). The datasets are different in nature where 25 out of the 30 stocks (83%) in the DJIA lost value, every stock in the NYSE increased in value, 6 of 263 stocks (2%) in the SP500 lost value, 7 of the 25 stocks (28%) in the SP500^a lost value, and 32 out of the 88 stocks (36%) in TSX lost value as detailed in Table 4.3. Note, the NYSE dataset captures the bear market that lasted between January 1973 and December 1974 and has been used extensively in the literature for demonstration of empirical results [4, 15, 34, 69] and the S&P500 dataset captures the bull and bear markets of recent times such as the dot-com bubble and burst between 1997-2000 and financial and housing bubble burst occurring between 2007-2009.

Dataset	Number of stocks	Number stock lost value	Trading days	Years
DJIA [15]	30	25	507	2001-2003
NYSE [69]	36	0	5651	1962-1984
S&P500 [39]	263	6	5162	1990-2010
S&P500^a [15]	25	7	1276	1998-2003
TSX [15]	88	32	1259	1994-1998

Table 4.1: Datasets with data taken from 4 different markets and trading periods.

Chapter 5

Online Lazy Updates

We start by considering problems which admit full information feedback. In particular, in this chapter we[1] will study how to design and analyze an algorithm which makes sparse or lazy updates to a resource allocation. In the context of portfolio selection, the algorithm makes sparse trades in order to control transaction costs. The work in this chapter first appeared as a peer-reviewed conference paper [41].

5.1 Introduction

With the ever increasing amount of data, particularly from search engines and social networks, online optimization algorithms have become desirable because of their efficiency and strong theoretical guarantees [3, 12, 16, 17, 111]. However, a major challenge is the cost of updating model parameters especially when there are billions of parameters. Often when parameters are updated, their values do not change significantly therefore, the cost of updating each parameter starts to outweigh the benefit. An important and relevant application where changing the model parameters might prove to be monetarily expensive is the domain of online portfolio selection. Every time an investor makes changes to the portfolio by buying or selling stocks, transactions costs are incurred. Hence, trading aggressively might hurt an investor instead of being beneficial. In such a situation, it might be helpful to make sparse or *lazy* updates to a portfolio.

In Section 4.2, we mentioned related work in portfolio selection and described a few

[1] The work in this chapter was done in collaboration with Puja Das.

algorithms. Although theoretical and empirical performance of such online portfolio selection algorithms have been encouraging, they have mostly ignored one crucial practical aspect of financial trading: transaction costs. These online algorithms [4, 15, 28, 34, 69] could be trading aggressively and a major concern is the cost they would incur in a real-world scenario. The need for considering transaction costs in the design and analysis of algorithms has been raised in [4, 15, 28, 35, 69, 104, 88]. However, only [14] has extended the analysis of [34] to include transaction costs. Their strategy first computes a target portfolio using the Universal Portfolio algorithm [34] and then pays for the transactions proportionally from each stock. Their analysis shows that the performance guarantee of the Universal Portfolio algorithm still holds (and gracefully degrades) in the case of proportional transaction costs. However, [34] is computationally demanding and has been shown to have sobering empirical performance [39, 69]. [14] and heuristics like Anticor [15] and OLMAR [87] do not account for transaction costs in their algorithm design. Anticor and OLMAR rely on empirical results to show scalability of their strategies to transaction costs only as a post-processing step.

In this chapter, we introduce an online portfolio selection algorithm *with transaction costs*. The algorithm is penalized by a fixed percentage of the amount of transactions it makes on a per day basis. We pose this as a non-smooth online convex optimization problem and propose an efficient algorithm called Online Lazy Updates (OLU) to make lazy updates to our online portfolio. Furthermore, we prove that our lazy portfolio is competitive with *fixed* and *shifting* strategies which have the benefit of hindsight.

We use the online portfolio selection model discussed in Chapter 4 for our experiments. Typically, there can be two types of transaction costs in real markets: (1) a *fixed percentage* of each transaction that the investor has to pay to a broker or (2) a *fixed amount* paid per transaction (sell or buy). In this work, we look at costs of the first type also known as *proportional transaction costs* in financial modeling [44, 95]. We conduct extensive experiments on the NYSE and S&P500 datasets (refer to Section 4.3 for more details on the two datasets). Our experiments show that our lazy portfolios are scalable with transaction costs and, interestingly, in some cases, can earn more wealth than their non-lazy counterparts which do not take transaction costs into account.

Overview of Contributions: We make two main contributions. First, we present a general framework and an efficient algorithm for lazy updates and instantiate it for the

problem of portfolio selection with transaction costs. We are one of the first to consider transaction costs in the algorithmic design and show that the algorithm out-performs existing methods in the literature with and without transaction costs.

Second, we provide theoretical analysis on both the fixed and shifting regret of our algorithm for general Lipschitz convex functions and strongly convex functions. Such a contribution extends the existing literature by showing a shifting regret bound when the loss function is composed of a smooth term and a non-smooth term (which is used to control the transaction costs). Previous work has only shown fixed regret bounds for smooth and/or non-smooth loss functions.

5.2 Problem Formulation

We present a general formulation for the online lazy updates problem and show how portfolio selection with transaction costs is a special case of such a setting. In an online lazy updates setting, the optimization proceeds in rounds where in round t the algorithm has to pick a solution $\mathbf{p}_t \in \mathcal{P} \subset \mathbb{R}^n$ from the feasible set such that it is close to the previous solution $\mathbf{p}_{t-1}$. Nature then selects a convex loss function $\phi_t : \mathcal{P} \to \mathbb{R}$, the feedback the algorithm receives is the entire loss function $\phi_t(\mathbf{p})\ \forall \mathbf{p} \in \mathcal{P}$, and the algorithms suffers a loss of $\phi_t(\mathbf{p}_t)$. Ideally, over T rounds we would like to minimize

$$\sum_{t=1}^{T} \phi_t(\mathbf{p}_t) + \gamma \sum_{t=2}^{T} \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1 \ . \tag{5.1}$$

The L_1 penalty term ensures that updates to the solution $\mathbf{p}_t$ are lazy. Absolute minimization of (5.1) is not feasible because we do not know the sequence of ϕ_t *a priori*. If the ϕ_ts are known, (5.1) reduces to a batch optimization problem: a special case is the fused lasso when ϕ_t is quadratic [120]. Hence, over T iterations we intend to get a sequence of $\mathbf{p}_t$ such that the following *regret bound* is sublinear in T

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}^*} \sum_{t=1}^{T} f_t(\mathbf{p}^*) = o(T) \tag{5.2}$$

where $f_t(\mathbf{p}) = \phi_t(\mathbf{p}) + \gamma||\mathbf{p} - \mathbf{p}_{t-1}||_1$ and $\mathbf{p}^*$ is the minimizer of $\sum_{t=1}^{T} f_t$ in hindsight. Note, while the $\mathbf{p}_t$s can change over time, $\mathbf{p}^*$ is fixed, i.e., $\mathbf{p}^* = \text{argmin}_{\mathbf{p}} \sum_{t=1}^{T} f_t(\mathbf{p}) =$

$\text{argmin}_{\mathbf{p}} \sum_{t=1}^{T} \phi_t(\mathbf{p})$, since it incurs zero L_1 penalty in every iteration. We will refer to this as the *fixed regret bound*.

Alternatively, over T rounds we intend to get a sequence of $\mathbf{p}_t$ such that the following *shifting regret bound* is sublinear in T

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*} \sum_{t=1}^{T} f_t(\mathbf{p}_t^*) = o(T) \tag{5.3}$$

where the sequence $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ is the minimizer of $\sum_{t=1}^{T} f_t$. The bound in (5.3) is a *shifting bound* [70] since it allows the $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ to also lazily change over time.

Following recent advances in online convex optimization, in order to accomplish a sublinear regret, we consider solving a linearized version of the problem obtained by a first-order Taylor expansion of ϕ_t at $\mathbf{p}_t$ along with a proximal term

$$\mathbf{p}_{t+1} = \underset{\mathbf{p} \in \mathcal{P}}{\text{argmin}} \underbrace{\phi_t(\mathbf{p}_t) + \langle \mathbf{p} - \mathbf{p}_t, \nabla_{\mathbf{p}_t} \phi_t(\mathbf{p}_t) \rangle}_{\text{Linear approximation}} + \underbrace{\gamma \|\mathbf{p} - \mathbf{p}_t\|_1}_{\substack{\text{Lazy update} \\ \text{penalty function}}} + \underbrace{d_\psi(\mathbf{p}, \mathbf{p}_t)}_{\text{Proximal term}} . \tag{5.4}$$

5.3 Algorithm

Online portfolio selection with transaction costs can now be viewed as a special case of our online lazy updates setting where $\mathcal{P} = \Delta_n = \{\mathbf{p} : p(i) \geq 0, \sum_{i=1}^{n} p(i) = 1\}$ is the n-dimensional probability simplex which forces the algorithm to invest all the wealth in each round of investing, $\phi_t(\mathbf{p}) = -\log(\mathbf{p}^\top \mathbf{x}_t)$ is the negative logarithmic gain in wealth used as the loss function, $\nabla_{\mathbf{p}_t} \phi_t(\mathbf{p}) = -\frac{\mathbf{x}_t}{\mathbf{p}_t^\top \mathbf{x}_t}$, and $d_\psi(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2\eta} \|\mathbf{p} - \mathbf{p}_t\|_2^2$. The portfolio for day $t+1$ is computed as

$$\mathbf{p}_{t+1} := \underset{\mathbf{p} \in \Delta_n}{\text{argmin}} - \log(\mathbf{p}_t^\top \mathbf{x}_t) - \frac{\mathbf{x}_t^\top (\mathbf{p} - \mathbf{p}_t)}{\mathbf{p}_t^\top \mathbf{x}_t} + \gamma \|\mathbf{p} - \mathbf{p}_t\|_1 + \frac{1}{2\eta} \|\mathbf{p} - \mathbf{p}_t\|_2^2 . \tag{5.5}$$

To simplifying the objective function, we drop the constant terms since it does not change the argument which minimizes the objective function. Then multiplying each term by η, (5.5) can be written as

$$\mathbf{p}_{t+1} := \underset{\mathbf{p} \in \Delta_n}{\text{argmin}} - \frac{\eta \mathbf{x}_t^\top}{\mathbf{p}_t^\top \mathbf{x}_t} \mathbf{p} + \alpha \|\mathbf{p} - \mathbf{p}_t\|_1 + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_t\|_2^2 \tag{5.6}$$

where $\alpha = \eta\gamma$. The L_1 penalty term on the difference of two consecutive portfolios measures the fraction of wealth traded. The parameter γ controls the amount that can be traded every day.

In the online lazy update framework, a new portfolio is computed as a function of $\mathbf{p}_t$, the price relatives $\mathbf{x}_t$, and lives in the n-dimensional probability simplex. The purpose of the first term is to maximize the logarithmic wealth if the current price relative $\mathbf{x}_t$ is replicated. The L_1 penalty term accounts for the amount of transaction that would take place to update the portfolio. The parameter $\alpha > 0$ determines how frequent trades are performed; high values of α lead to *lazy* updates of the portfolio with few transactions while low values allow the portfolio to change more frequently. Our framework for updating a portfolio vector is analogous to the framework of the EG algorithm [69]. We use $||\cdot||_2^2$ as the distance function instead of relative entropy. However, unlike EG, we solve a non-smooth problem.

We propose an ADMM (Alternating Direction Method of Multipliers [19]) based efficient primal-dual algorithm to obtain the lazy portfolio $\mathbf{p}_{t+1}$ by solving (5.6). We can rewrite (5.6) in the ADMM form by introducing an auxiliary variable $\mathbf{z}$ as

$$\mathbf{p}_{t+1} := \underset{\mathbf{p}\in\Delta_n,\, \mathbf{p}-\mathbf{p}_t=\mathbf{z}}{\operatorname{argmin}} \; -\frac{\eta \mathbf{x}_t^\top}{\mathbf{p}_t^\top \mathbf{x}_t}\mathbf{p} + \alpha\|\mathbf{z}\|_1 + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 \; . \tag{5.7}$$

The ADMM formulation allows decoupling the non-smooth L_1 term from the smooth terms which is computationally advantageous. The *augmented Lagrangian* for (5.7) is

$$L_\beta(\mathbf{p},\mathbf{z},\mathbf{u}) := \underset{\mathbf{p}\in\Delta_n}{\operatorname{argmin}} \; -\frac{\eta \mathbf{x}_t^\top}{\mathbf{p}_t^\top \mathbf{x}_t}\mathbf{p} + \alpha\|\mathbf{z}\|_1 + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}+\mathbf{u}\|_2^2 \tag{5.8}$$

where $\mathbf{u} = \frac{1}{\beta}\boldsymbol{\lambda}$ is the scaled dual variable and $\boldsymbol{\lambda}$ is the dual variable. ADMM consists of the following iterations for solving $\mathbf{p}_{t+1}$

$$\mathbf{p}_{t+1}^{(k+1)} := \underset{\mathbf{p}\in\Delta_n}{\operatorname{argmin}} \; -\frac{\eta \mathbf{x}_t^\top}{\mathbf{p}_t^\top \mathbf{x}_t}\mathbf{p} + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}^{(k)}+\mathbf{u}^{(k)}\|_2^2 \tag{5.9}$$

$$\mathbf{z}^{(k+1)} := \underset{\mathbf{z}}{\operatorname{argmin}} \; \alpha\|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{p}_{t+1}^{(k+1)}-\mathbf{p}_t-\mathbf{z}+\mathbf{u}^{(k)}\|_2^2 \tag{5.10}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (\mathbf{p}_{t+1}^{(k+1)} - \mathbf{p}_t - \mathbf{z}^{(k+1)}) \; . \tag{5.11}$$

Algorithm 1 shows the closed form updates derived for $\mathbf{p}_{t+1}^{k+1}$, $\mathbf{z}^{k+1}$, and $\mathbf{u}^{k+1}$. The update for $\mathbf{p}_{t+1}^{k+1}$ is derived by taking the derivative of (5.9), setting it to zero, and solving for $\mathbf{p}$. The projection to the simplex $\prod_{\Delta_n}$ is carried out as in [47]. The stopping criteria for the OLU algorithm is based on the primal and dual residuals from [19].

Algorithm 1 Online Lazy Update (OLU) with ADMM

1: Input $\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha, \beta$

2: Initialize $\mathbf{p}, \mathbf{z}, \mathbf{u} \in 0^n, k = 0$

3: ADMM iterations

$$\mathbf{p}^{(k+1)} = \prod_{\Delta_n} \Big\{ -\frac{\eta \mathbf{x}_t}{(\beta+1)\mathbf{p}_t^\top \mathbf{x}_t} + \mathbf{p}_t + \frac{\beta \mathbf{z}^{(k)}}{(\beta+1)} - \frac{\beta \mathbf{u}^{(k)}}{(\beta+1)} \Big\}$$
$$\mathbf{z}^{(k+1)} = S_{\alpha/\beta}(\mathbf{p}^{(k+1)} - \mathbf{p}_t + \mathbf{u}^{(k)})$$
$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (\mathbf{p}^{(k+1)} - \mathbf{p}_t - \mathbf{z}^{(k+1)})$$

where $\prod_{\Delta_n}$ is a projection to the simplex and S_ρ is the shrinkage operator.

4: Continue until **Stopping Criteria** is satisfied

Algorithm 2 Portfolio Selection with Transaction costs

1: Input η, γ, β; Compute $\alpha = \eta\gamma$

2: Initialize $p_{1,h} = \frac{1}{n}, h = 1, \ldots, n; \mathbf{p}_1 = \mathbf{p}_0; S_0^\gamma = 1$

3: For $t = 1, \ldots, T$

4: Receive $\mathbf{x}_t$ vector of price relatives

5: Compute cumulative wealth: $S_t^\gamma = S_{t-1}^\gamma \times (\mathbf{p}_t^\top \mathbf{x}_t) - \gamma \times S_{t-1}^\gamma \times ||\mathbf{p}_t - \mathbf{p}_{t-1}||_1$

6: Update portfolio: $\mathbf{p}_{t+1} = \text{OLU}(\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha, \beta)$

7: end for

Algorithm 2 is our online portfolio selection algorithm with transaction costs. It uses the OLU Algorithm to compute the lazy portfolio $\mathbf{p}_{t+1}$. It takes as input an additional parameter $\gamma \geq 0$ which is a fixed percentage charged for the total amount of transaction everyday. S_t^γ is the transaction cost-adjusted cumulative wealth at the end of t days.

5.4 Regret Analysis

We now present an analysis of our OLU algorithm. The updates we consider follow the general form:

$$\mathbf{p}_{t+1} = \underset{\mathbf{p} \in \mathcal{P}}{\text{argmin}}\ \eta \langle \nabla \phi_t(\mathbf{p}_t), \mathbf{p} \rangle + \eta\gamma \|\mathbf{p} - \mathbf{p}_t\|_1 + d_\psi(\mathbf{p}, \mathbf{p}_t)\ . \tag{5.12}$$

In general, a Bregman divergence with respect to a strictly convex function $\psi : \mathcal{P} \to \mathbb{R}$ is defined as $d_\psi(\mathbf{p}, \mathbf{p}_t) = \psi(\mathbf{p}) - \psi(\mathbf{p}_t) - \langle \nabla\psi(\mathbf{p}_t), \mathbf{p} - \mathbf{p}_t \rangle$. For our online lazy updates formulation, $\psi(\mathbf{p}) = \|\mathbf{p}\|^2$ and $d_\psi(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2}||\mathbf{p} - \mathbf{p}_t||_2^2$. For the portfolio

selection problem $\phi_t(\mathbf{p}_t) = -\log(\mathbf{p}_t^\top \mathbf{x}_t)$ but our analysis holds for any convex ϕ_t. From Algorithm 1, we obtain a sequence of lazy solutions $\{\mathbf{p}_1, \ldots, \mathbf{p}_T\}$ and on day t suffer a loss of $f_t(\mathbf{p}_t) = \phi_t(\mathbf{p}_t) + \gamma\|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1$. Our first goal is to minimize the *regret* with respect to the best fixed solution $\mathbf{p}^*$ in hindsight. Here, $\mathbf{p}^*$ is the optimal, fixed (constantly rebalanced) portfolio we could have computed if we had knowledge of the price relatives $\mathbf{x}_{1:T}$ beforehand. Our second goal is to minimize the regret with respect to the best sequence of shifting solutions $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*\}$ in hindsight. The sequence of solutions $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*\}$, is the optimal portfolio we could have computed each day with knowledge of the price relatives beforehand.

We first establish the standard fixed regret bounds for our lazy updates in Theorem 1 and Theorem 2. We then go on to establish the shifting bounds in Theorem 5.20. For both the fixed and shifting bounds, we focus on settings when the ϕ_ts are *general* convex functions. For the fixed regret bound, we additionally consider when the ϕ_ts are *strongly* convex functions. The proofs of all lemmas and theorems are in Appendix A.1.

5.4.1 Fixed Regret

We first focus on the case where the comparator class is fixed, i.e., $\mathbf{p}^*$ is the minimizer of $\sum_{t=1}^T \phi_t$ in hindsight as it incurs zero L_1 penalty in every iteration and we prove *regret bounds* as in (5.2). We show that for general convex functions the regret is $O(\sqrt{T})$ while for strongly convex functions the regret is $O(\log T)$.

General Convex Functions

We assume that ϕ_t are general convex functions with bounded (sub)gradients, i.e., for any $\hat{g} \in \partial\phi_t(\mathbf{p})$ we have $\|\hat{g}\|_2 \leq G$.

Lemma 1 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (5.12), with potentially time-varying η_t, γ_t. Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ and let $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. Then, for any $\mathbf{p}^* \in \mathcal{P}$,*

$$\begin{aligned} &\eta_t[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \\ &\leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\gamma_t L + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ . \end{aligned} \tag{5.13}$$

Based on the above result, we obtain the following fixed regret bound for Algorithm1:

Theorem 1 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (5.12). Let* ϕ_t *be a Lipschitz continuous function for which* $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$*. Then, by choosing* $\eta_t = \eta = \frac{c_1}{\sqrt{T}}$ *and* $\gamma_t = \frac{c_2}{\sqrt{t}}$ *for* $c_1, c_2 > 0$*, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \leq O(\sqrt{T}) \ . \tag{5.14}$$

Note, instead of an adaptive $\gamma_t = \frac{c_2}{\sqrt{t}}$, one can choose a fixed $\gamma = \frac{c_2}{\sqrt{T}}$ and obtain the same result, up to constants.

Strongly Convex Functions

Next, we assume that ϕ_t are all β-strongly convex functions so that for any $(\mathbf{p}, \mathbf{p}_t)$

$$\phi_t(\mathbf{p}) \geq \phi_t(\mathbf{p}_t) + \langle \mathbf{p} - \mathbf{p}_t, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{\beta}{2}\|\mathbf{p} - \mathbf{p}_t\|^2 \ . \tag{5.15}$$

Lemma 2 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (5.12) with potentially time-varying* η_t *and fixed* γ*. Let* $d_\psi(\cdot,\cdot)$ *be* λ*-strongly convex with respect to norm* $||\cdot||_2$*, i.e.* $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ *and* $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$*. Assuming* ϕ_t *are all* β*-strongly convex, for any* $\gamma < \frac{\beta}{4}$ *and any* $\mathbf{p}^* \in \mathcal{P}$*, we have*

$$\begin{aligned} &\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \gamma\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1] \\ &\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 - \eta_t\left(\frac{\beta}{2} - 2\gamma\right)\|\mathbf{p}^* - \mathbf{p}_t\|^2 \ . \end{aligned} \tag{5.16}$$

Theorem 2 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (5.12). Let* ϕ_t *be all* β*-strongly convex and* $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$*. Then, for any* $\gamma < \beta/4$*, choosing* $\eta_t = \frac{1}{\kappa t}$ *where* $\kappa \in (0, \beta - 4\gamma]$*, and with* $d_\psi(\mathbf{p}, \mathbf{p}') = \frac{1}{2}\|\mathbf{p} - \mathbf{p}'\|^2$*, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \gamma\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \leq O(\log T) \ . \tag{5.17}$$

5.4.2 Shifting Regret

We now focus on the case where the comparator class can also shift or change over time, i.e., time-varying sequences $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ is used as the comparator class on the cumulative loss $\sum_{t=1}^T f_t$. Such a time varying sequence will incur cumulative non-zero shifting penalty penalty of the form

$$\text{shift}_q(\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*) = \sum_{t=1}^{T} \|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ , \tag{5.18}$$

for some suitable q-norm for $q \geq 1$. The regret bounds in this section are in terms of such shifting penalties. A specific case of interest is when the comparator class consists of constant shifting penalties, i.e.,

$$\text{shift}_q(\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*) \leq c \ . \tag{5.19}$$

We show that for this important special case, the shifting regret bounds are of the same order as the regret bounds with a fixed comparator, i.e., $\mathbf{p}_t^* = \mathbf{p}^*$, as discussed earlier.

General Convex Functions

We assume that ϕ_t are general convex functions with bounded (sub)gradients, i.e., for any $\hat{g} \in \partial\phi_t(\mathbf{p})$ we have $\|\hat{g}\| \leq G$.

Theorem 3 *Let $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ be any sequence of portfolios serving as a comparator in (5.12). Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. For $\eta_t = \eta = \frac{c_1}{\sqrt{T}}$ and $\gamma_t = \frac{c_2}{\sqrt{t}}$ for $c_1, c_2 > 0$, $\frac{1}{r} + \frac{1}{q} = 1$, and $||\nabla\psi(\mathbf{p}_t)||_r \leq \zeta$, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}_t^*)] + \sum_{t=1}^{T}[\gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \tag{5.20}$$

$$\leq O(\sqrt{T}) + \frac{\sqrt{T}}{c_1}\left\{d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) + \zeta\sum_{t=1}^{T}\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q\right\}.$$

Assuming $d_\psi(\mathbf{p}, \mathbf{p}') \leq c_3, \psi(\mathbf{p}) \leq c_4$, and the shifting penalty $\sum_{t=1}^T \|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \leq c$, the shifting regret is $O(\sqrt{T})$, which is the same order as the fixed regret.

5.5 Experiments and Results

In this section, we present experimental results for our OLU algorithm for transaction cost-adjusted portfolio selection on two real-world datasets: the New York Stock Exchange (NYSE) [34] and the Standard & Poor's 500 (S&P 500) [39] (refer to Section 4.3 for more details).

5.5.1 Methodology and Parameter Setting

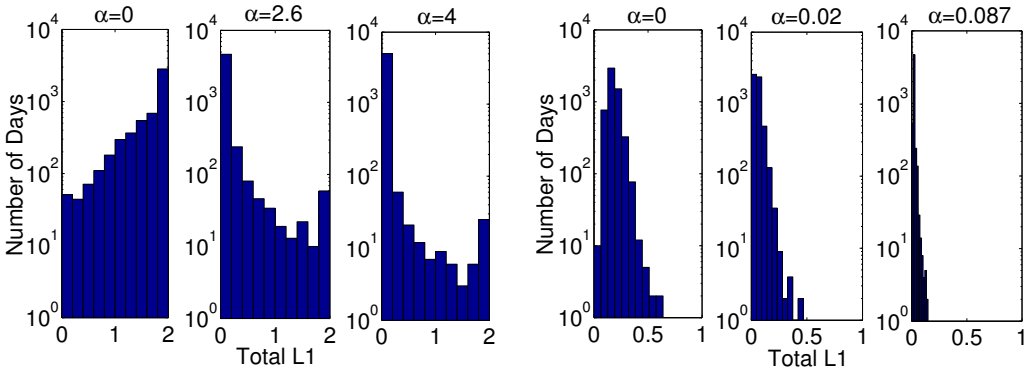
In all the experiments, an initial investment of \$1 and an initial portfolio uniformly distributed over all the stocks were used. Algorithm 2 was used to obtain the portfolios and compute the transaction cost-adjusted wealth each day. For all the experiments, $\beta = 0.1$ which was found to give reasonable and consistent accuracy. Table 5.1 contains the description of the various parameters.

Since the two datasets are different in nature (stock composition and duration), we experimented with a range of η and α values in $[10^{-6}, 10^{6}]$ to observe their effect on the lazy updates. Moreover, a range of γ values in $[0\%, 2\%]$ was used to compute the proportional transaction costs. Representative plots from either the NYSE or S&P500 datasets were used to illustrate the results.

We compared the wealth obtained from the EG algorithm, a uniform constant rebalanced portfolio (U-CRP), and a Buy-and-Hold strategy (without transaction costs) as benchmarks. EG has been shown to be competitive with U-CRP and, in some cases, outperform in terms of wealth earned [39, 69]. For the Buy-and-Hold strategy, we started with a uniformly distributed portfolio and performed a hold on the positions thereafter, i.e., no trades. Additionally, the S&P500 Index was used as a representative index for the US stock market to analyze the activity of our lazy update algorithm. We did not compare our method with Anticor or OLMAR because these heuristics do not account

η	Weight on logarithmic gain in wealth.
γ	Fixed transaction (pre-specified) cost expressed as a percentage.
α	Weight on L_1 penalty term: computed in Algorithm 2 as $\eta \times \gamma$.
β	Weight on the augmented Lagrangian: parameter for OLU algorithm with ADMM.

Table 5.1: Parameter descriptions as given in (5.6) and used in Algorithm 1 and 2.



(a) L_1 values for the S&P500 dataset.

(b) L_1 values for the NYSE dataset.

Figure 5.1: With a low $\alpha = 0$, we are effectively removing the lazy updates term and allowing the algorithm to trade aggressively to maximize returns. This is equivalent to what the EG algorithm is doing. With higher α values, we are penalizing the amount of transactions more severely and, as such, the total amount decreases as we increase α.

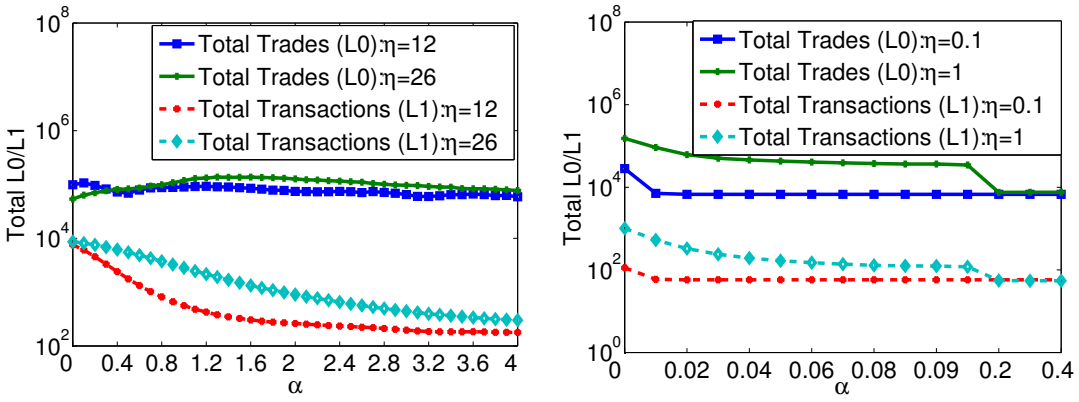
for transaction costs in their algorithmic framework and have no theoretical guarantees. We also did not compare with the Universal Portfolio algorithm because it has been shown to have sobering empirical performance even in the absence of transaction costs when compared to EG and Buy-and-Hold [39, 69].

5.5.2 Effect of α and the L_1 penalty

The parameter α is the weight on the L_1 penalty term and can influence (a) the total daily amount of transactions, (b) the total amount of trades and transactions, (c) the daily percentage of transactions, and (d) the stock activity.

(a) Total Daily Amount of Transactions: Let $\Upsilon_t = ||\mathbf{p}_{t+1} - \mathbf{p}_t||_1$, then $\sum_{t=1}^{T-1} \Upsilon_t$ is a measure of the total amount that a trader had to pay in transaction costs over T days (as a fraction of his wealth). Figure 5.1 plots histograms of Υ_t for varying α values. We observe that as α increases, the Υ_t value is small for most days. With $\alpha = 0$, Υ_t was 2 for most days denoting non-lazy portfolios which is how portfolios are expected to trade in a costless environment.

(b) Total Amount of Trades and Transactions: We now analyze the behavior of the total number of trades (L_0 norm) and the total amount of transactions ($\sum_{t=1}^{T-1} \Upsilon_t$)



(a) Number (L_0 norm) and amount (L_1 norm) of trades for the S&P500 dataset.

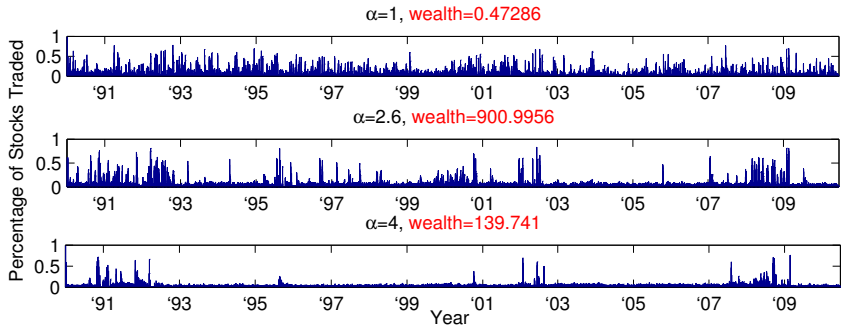
(b) Number (L_0 norm) and amount (L_1 norm) of trades for the NYSE dataset.

Figure 5.2: As α increases, the total amount of transaction and number of trades decrease for the NYSE dataset. However, for the S&P500 dataset, as we increase α the total amount of transaction decreases but the total number of trades does not decrease.

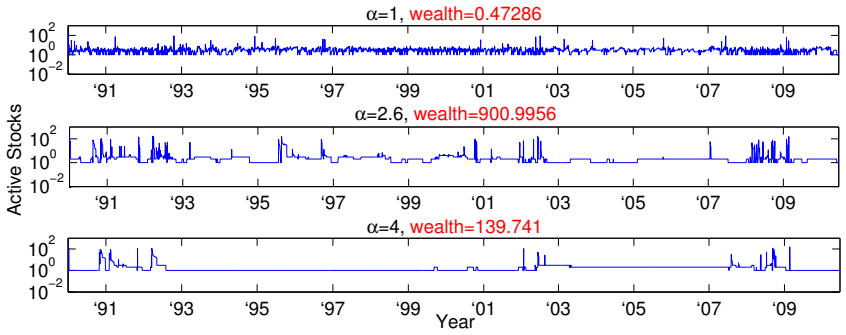
as we increase the value of α. Figure 5.2 gives an overview of the behavior of the aforementioned quantities as we increase α. Figure 5.2 confirms that the total L_1 norm decreases as we increase α. The total L_0 norm, however, does not always decrease as we increase α. Figure 5.2(a) shows such a situation for the S&P500 dataset. For the NYSE dataset in Figure 5.2(b), however we observe that the L_0 norm also decreases as we increase α.

(c) Daily Percentage of Transactions: Figure 5.3(a) plots the percentage of stocks traded per day for the S&P500 dataset for three values of α. We observe that as we increase the weight on the L_1 penalty term by tuning α, the number of transactions decreases. Whereas a large amount of the 263 stocks were traded everyday for $\alpha = 0$, with higher values of α the number of transactions reduces significantly. We observe a similar trend for the NYSE dataset.

(d) Active Stocks: Figure 5.3(b) plots the number of stocks which comprise 80% of the total wealth on a per day basis which we call the *active* stocks. As α increases, the lazy behavior of the portfolios becomes more apparent and the online portfolios change their composition only on a handful of days.



(a) Percentage of stocks traded each day.



(b) Stocks which comprise 80% of the wealth.

Figure 5.3: In (a) with $\alpha = 1$, the percentage of stocks frequently traded is high. As α increases, the percentage and frequency of stocks traded decreases. However, for each value of α we still see activity during periods of economic instability. In (b) with $\alpha = 1$, the number of active stocks changes often which shows the algorithm is making frequent trades. As α increases, the number of active stocks stabilizes which means the algorithm is not trading as often and instead investing and holding on to a few stocks.

Correlation with the US market: In Figure 5.3(b), we observe significant activity between years 2002-2003 and between years 2008-2009. On plotting the value of the S&P500 index for the US market between 1990 and 2010 in Figure 5.4, we realized that the increase in trading activity reflected two major market movements: the dot-com and housing bubble bursts. Figure 5.4 shows that the days of high stock activity coincides with major market movements. Similar trends were observed for the NYSE dataset.

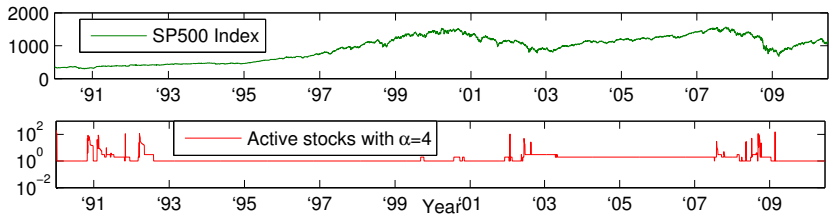
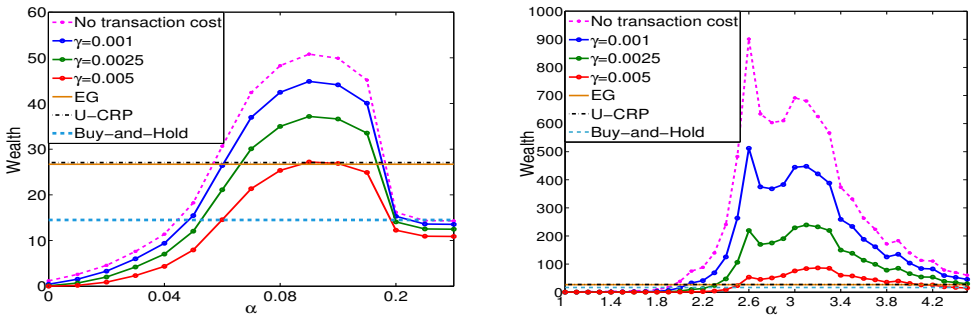


Figure 5.4: We compare active stocks with $\alpha = 4$ to the S&P500 index. When the index decreases between 2000-2003 and 2007-2009, the number of active stocks increases and OLU starts to make frequent trades. This implies during times of economic instability, e.g., the dot-com and financial crashes, OLU is trading frequently to find good stocks however, many stocks are performing badly with high volatility.

5.5.3 Wealth with Transaction Costs (S_T^γ)

To evaluate the practical application of our proposed algorithm, we now analyze its performance when calculating the transaction cost-adjusted cumulative wealth. Figure 5.5 shows how the choice of different α values affect the transaction cost-adjusted cumulative wealth for the two datasets (for a fixed η value). Figures 5.5(a) and 5.5(b) demonstrate that there exists a regime of $\alpha = \eta\gamma$ which makes an optimum choice between exploration and exploitation of stocks. Since γ can be fixed, the learning rate η can be adequately chosen to maximize wealth. Low values of α tend to aggressively change the portfolio too often. Whereas, with high values, the algorithm is too conservative and might not be able to take advantage of short trends in the market.

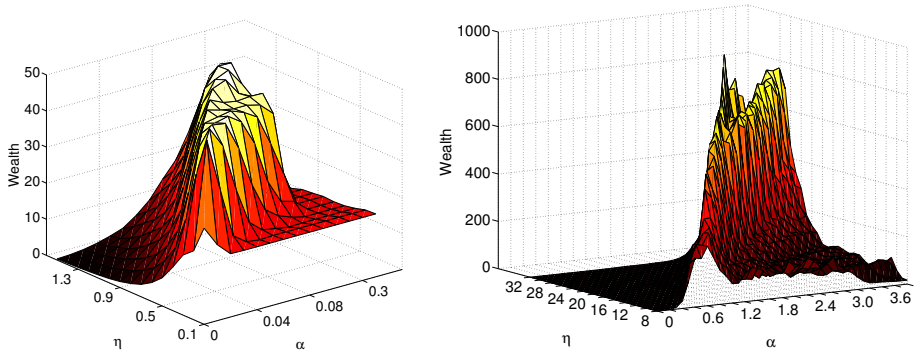
EG, U-CRP, and Buy-and-Hold: We compared the total wealth without transaction costs of OLU with that of EG, U-CRP, and a Buy-and-Hold strategy. EG, U-CRP, and Buy-and-Hold are plotted as horizontal lines and we can see that for the NYSE dataset U-CRP returns \$27.08, EG returns \$26.70, and Buy-and-Hold returns \$14.50. For the S&P500 dataset U-CRP returns \$27.29, EG returns \$26.78, and Buy-and-Hold returns \$16.65. In comparison, OLU returns \$50.80 and \$901.00 respectively without transaction costs. OLU returns almost 2x as much wealth for the NYSE dataset and 33x as much wealth for the S&P500 dataset as EG, U-CRP, or Buy-and-Hold do. Figures 5.5(a) and 5.5(b) also show that OLU is able to return more wealth than EG, U-CRP, and Buy-and-Hold with reasonable transaction costs (0.1%, 0.25%, and 0.5%).



(a) Wealth for the NYSE dataset.

(b) Wealth for the S&P500 dataset.

Figure 5.5: Transaction cost-adjusted cumulative wealth. In (a) with $\alpha = 0.087$ and in (b) with $\alpha = 2.6$, OLU earns more wealth ($50.80 and $901.00 respectively) than the competing algorithms even with transaction costs.



(a) Wealth for varying η and α values for the NYSE dataset.

(b) Wealth for varying η and α values for the S&P500 dataset.

Figure 5.6: Wealth as a function of η and α values. In (a) and (b) there is a range of parameter values that given significant wealth.

5.5.4 Parameter Sensitivity (η and α)

Figure 5.6 gives us more insight into how the transaction cost-adjusted wealth behaves as a function of η and α for the two datasets. We can see that the cumulative wealth looks like a hill or ridge and that on either sides of the ridge the wealth is small. This particularly occurs when either η or α are too high or too low. Only when both η and α are in relative balance are we able to obtain significant cumulative wealth.

Chapter 6

Online Lazy Updates with Group Sparsity

We saw in the previous chapter how to consider the cost of updating one's allocation of a resource in the algorithmic design. In particular, we saw how to effectively control transaction costs in online portfolio selection which is an important practical aspect mostly ignored in the literature but which is a real concern for traders. Another aspect which has not been considered is how to invest in groups of stock which perform similarly. In this chapter, we[1] will show how to design an algorithm which explicitly considers market sectors when investing and how such consideration leads to gains in wealth. The work in this chapter first appeared as a peer-reviewed conference paper [42].

6.1 Introduction

In financial trading, investors often follow a top down approach which usually involves group selection followed by identifying the most profitable stocks within a group. One of the ways investors group stocks is by the type of business. The idea is to put companies in similar sectors together. However, not all sectors can yield profit and not all stocks in a particular sector can be profitable. Moreover, sectors might react differently during different economic conditions [91, 6]. For example, defensive sectors like utilities and consumer staples are robust to economic downturns whereas cyclical

[1] The work in this chapter was done in collaboration with Puja Das.

sectors which include technology, financials, health care, etc., tend to react quickly to fluctuations in the market. We are particularly interested in exploiting any underlying structure amongst the stocks for the problem of online portfolio selection.

One aspect which is missing from the existing portfolio selection literature is taking advantage of the group structure that could exist amongst the stocks. In this chapter, we consider such an aspect and focus on using a group sparsity inducing regularizer to identify well performing groups of stocks in an online learning framework. In addition to considering groups of stocks, we also continue to consider transaction costs and use the L_1 regularizer introduced in Chapter 5 to induce lazy updates to the portfolio in order to control proportional transaction costs.

Overview of Contributions: We make two main contributions. First, we propose a general online lazy updates with group sparsity framework and go on to show that the online portfolio selection with sector information is a special case of such a framework. We pose the problem as a constrained non-smooth convex optimization problem at every iteration. We propose a novel alternating direction method of multipliers (ADMM) algorithm to efficiently solve such a problem. Using the algorithm, we conduct extensive experiments on two real-world datasets (NYSE and S&P500) and use the Global Industry Classification Standard to group the stocks into sectors. Our experiments show that our sparse group lazy portfolios can take advantage of the sector information to beat the market and are scalable with transaction costs. It shows an interesting group switching behavior and could be especially beneficial for individual investors who have expertise in select market sectors and are averse to changing their portfolio.

Second, we present an analysis for any convex composite function with lazy updates and show that our algorithm has $O(\sqrt{T})$ regret for a general convex functions and $O(\log T)$ regret for strongly convex functions. Additionally, we prove regret bounds with respect to a shifting solution which has the benefit of hindsight.

6.2 Problem Formulation

We present a general formulation for our online lazy algorithm with group sparsity and go on to show how the portfolio selection problem is a special case of our setting. In an online lazy setting, the optimization proceeds in rounds where in round t the algorithm

has to pick a solution from a feasible set, $\mathbf{p}_t \in \mathcal{P}$, such that it is sparse in the number of groups picked and close to the previous solution $\mathbf{p}_{t-1}$. Nature then selects a convex loss function $\phi_t : \mathcal{P} \to \mathbb{R}$, the algorithm observes the entire loss function $\phi_t(\mathbf{p})\ \forall \mathbf{p} \in \mathcal{P}$ as feedback, and suffers a loss of $\phi_t(\mathbf{p}_t)$. Ideally, over T rounds we would like to minimize

$$\sum_{t=1}^{T} \{\phi_t(\mathbf{p}_t) + \lambda_1 \Omega(\mathbf{p}_t)\} + \lambda_2 \sum_{t=1}^{T-1} \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 \ . \tag{6.1}$$

In (6.1), $\Omega(\cdot) : \mathcal{P} \to \mathbb{R}$ is a penalty function which we use to select groups of stocks and can be any group norm which will ensure group sparsity. We adopt the "groupwise" L_2-norm used in group lasso [128, 59] as our regularizer

$$\lambda_1 \Omega(\mathbf{p}) = \lambda_1 \sum_{g=1}^{\mathcal{G}} w_g \|\mathbf{p}_{|g}\| \ . \tag{6.2}$$

Let $\mathcal{G}$ be a set of groups and $\forall g \in \mathcal{G}, g \subseteq \{1, \ldots, n\}$. We define $\mathbf{p}_{|g}$ as the vector whose coordinates are equal to those of $\mathbf{p}$ for indices in the set g and 0 otherwise. We define weights $(w_g)_{g \in \mathcal{G}}$ where each $w_g \geq 0$ and $\|\cdot\|$ is the Euclidean norm. To introduce group sparsity, it is also possible to impose other joint regularization on the weight, e.g., the $L_{(1,\infty)}$-norm [106]. We consider the case where the groups are disjoint, i.e., $\mathcal{G}$ is separable over $\{1, \cdots, n\}$, however our framework and algorithm can be extended to overlapping group lasso [74]. The L_1 penalty term in (6.1) ensures lazy updates to $\mathbf{p}_t$.

Similar to what we considered in Chapter 5, absolute minimization of (6.1) is not possible because we do not know the sequence of ϕ_t *a priori*. If the ϕ_ts are known, (6.1) reduces to a batch optimization problem: a special case is the fused group lasso when ϕ_t is quadratic [59, 120] or TV regularization [108]. Alternatively, over T iterations we again intend to select a sequence of $\mathbf{p}_t$ such that the fixed regret is sublinear in T,

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}^* \in \mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{p}^*) = o(T) \tag{6.3}$$

where $f_t(\mathbf{p}) = \phi_t(\mathbf{p}) + \lambda_1 \Omega(\mathbf{p}) + \lambda_2 \|\mathbf{p} - \mathbf{p}_{t-1}\|_1$ is non-smooth and $\mathbf{p}^*$ is the minimizer of $\sum_{t=1}^{T} f_t$ in hindsight.

Additionally, we again examine the case where the comparator class can also change over time. In particular, we consider the sequence $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ which has the power

of hindsight. Then, over T iterations we ensure that the following shifting regret is sublinear in T

$$\sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}_1^*,\cdots,\mathbf{p}_T^*} \sum_{t=1}^{T} f_t(\mathbf{p}_t^*) = o(T) + c \text{ size}(\langle \mathbf{p}_1^*, \cdots, \mathbf{p}_T^* \rangle) \ , \tag{6.4}$$

where size$(\langle \mathbf{p}_1^*, \cdots, \mathbf{p}_T^* \rangle)$ measures the amount of shifting that occurs in the best sequence of solutions in hindsight and c is a constant. Online portfolio selection with group sparsity can now be viewed as a special case of the above setting where $\phi_t(\mathbf{p}) = -\log(\mathbf{p}^\top \mathbf{x}_t)$ and the L_1 penalty term on the difference of two consecutive portfolios measures the fraction of wealth traded. The parameters λ_1 controls how many groups are selected (setting $\lambda_1 = 0$ reduces to the OLU problem considered in Chapter 5) and λ_2 controls the amount that can be traded every day.

To show a sublinear regret bound, we solve a linearized version of the problem as we did in Chapter 5 by taking a first-order Taylor expansion of ϕ_t at $\mathbf{p}_t$ with a proximal term

$$\mathbf{p}_{t+1} = \text{argmin}_{\mathbf{p}\in\mathcal{P}} \ \langle \nabla\phi_t(\mathbf{p}_t), \mathbf{p} \rangle + \lambda_1 \Omega(\mathbf{p}) + \lambda_2 \|\mathbf{p} - \mathbf{p}_t\|_1 + \frac{1}{2\eta}\|\mathbf{p} - \mathbf{p}_t\|_2^2 \tag{6.5}$$

where we use the squared Euclidean distance as the proximal term and ignore the constants since they do not change which argument optimizes the objective function.

6.3 Algorithm

We now present our Online Lazy Updates with Group Sparsity (OLU-GS) algorithm. In the sequel, we show that using the solutions generated by OLU-GS, we can achieve sublinear regret for the fixed (6.3) and shifting case (6.4). At the beginning of day $t+1$, we find a new solution $\mathbf{p}_{t+1}$ by minimizing (6.5). The objective function in (6.5) is composite with smooth and non-smooth terms with the probability simplex as a constraint set. Although there is literature on solving composite functions [48, 127], composite functions with linear constraints have not been adequately investigated. We propose an Alternating Direction Method of Multipliers (ADMM) [19] based efficient primal-dual algorithm to solve (6.5). We rewrite (6.5) in ADMM form by introducing auxiliary variables $\mathbf{y}$ and $\mathbf{z}$

$$\underset{\mathbf{p}\in\Delta_n, \mathbf{p}=\mathbf{y}, \mathbf{p}-\mathbf{p}_t=\mathbf{z}}{\text{argmin}} \ \langle \nabla\phi_t(\mathbf{p}_t), \mathbf{p} \rangle + \lambda_1 \Omega(\mathbf{y}) + \lambda_2 \|\mathbf{z}\|_1 + \frac{1}{2\eta}\|\mathbf{p} - \mathbf{p}_t\|_2^2 \ . \tag{6.6}$$

Using variable splitting, we write the *augmented lagrangian* as,

$$L(\mathbf{p},\mathbf{y},\mathbf{z},\mathbf{w},\mathbf{v}) = \langle \nabla\phi_t(\mathbf{p}_t),\mathbf{p}\rangle + \lambda_1\Omega(\mathbf{y}) + \lambda_2\|\mathbf{z}\|_1 + \frac{1}{2\eta}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{y}+\mathbf{w}\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}+\mathbf{v}\|_2^2$$

where $\mathbf{w}$ and $\mathbf{v}$ are the scaled dual variables and $\mathbf{p} \in \Delta_n$. Splitting the variables as we do in (6.6) has two advantages. Firstly, we will show there is a closed form solution for each update. Secondly, the updates for $\mathbf{y}$ and $\mathbf{z}$ can be done in parallel and the same is true for the scaled dual variables $\mathbf{w}$ and $\mathbf{v}$. ADMM consists of the following iterations for solving $\mathbf{p}_{t+1}$,

$$\mathbf{p}_{t+1}^{(k+1)} = \underset{\mathbf{p}\in\Delta_n}{\operatorname{argmin}} \langle \nabla\phi_t(\mathbf{p}_t),\mathbf{p}\rangle + \frac{1}{2\eta}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{y}^{(k)}+\mathbf{w}^{(k)}\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}^{(k)}+\mathbf{v}^{(k)}\|_2^2 \tag{6.7}$$

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}}{\operatorname{argmin}}\ \lambda_1\Omega(\mathbf{y}) + \frac{\beta}{2}\|\mathbf{p}_{t+1}^{(k+1)}-\mathbf{y}+\mathbf{w}^{(k)}\|_2^2 \tag{6.8}$$

$$\mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\operatorname{argmin}}\ \lambda_2\|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{p}_{t+1}^{(k+1)}-\mathbf{p}_t-\mathbf{z}+\mathbf{v}^{(k)}\|_2^2 \tag{6.9}$$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + (\mathbf{p}_{t+1}^{(k+1)} - \mathbf{y}^{(k+1)}) \tag{6.10}$$

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + (\mathbf{p}_{t+1}^{(k+1)} - \mathbf{p}_t - \mathbf{z}^{(k+1)})\ . \tag{6.11}$$

p-update: We take the derivative of (6.7) with respect to $\mathbf{p}$, set it to zero, and solve for $\mathbf{p}$ to get a closed form update. The projection $\prod_{\mathbf{p}\in\Delta_n}$ is carried out as in [47].

y-update: We can rewrite (6.8) as

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{p}^{(k+1)}+\mathbf{w}^{(k)}-\mathbf{y}\|_2^2 + \frac{\lambda_1}{\beta}\Omega(\mathbf{y})\ . \tag{6.17}$$

(6.17) is *separable* in every group when $\Omega(\cdot)$ is a group lasso penalty with L_2 norm and $\mathcal{G}$ is a partitioning of $\{1,\cdots,n\}$ and the solution is a generalization of the soft-thresholding operator to groups of variables [75]:

$$\forall g \in \mathcal{G}, \mathbf{y}_{|g} = \begin{cases} 0 & \text{if } \|\mathbf{q}_{|g}\|_2 \le \tilde{\lambda} \\ \frac{\|\mathbf{q}_{|g}\|_2 - \tilde{\lambda}}{\|\mathbf{q}_{|g}\|_2}\mathbf{q}_{|g} & \text{otherwise} \end{cases} \tag{6.18}$$

Algorithm 3 OLU-GS Algorithm with ADMM

1: Input $\mathbf{p}_t, \mathbf{x}_t, \nabla\phi_t(\mathbf{p}_t), \mathcal{G}, \lambda_1, \lambda_2, \eta, \beta$

2: Initialize $\mathbf{p}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \mathbf{v} \in 0^n, k = 0$

3: Set $\hat{a} = \frac{1+\eta\beta}{1+2\eta\beta}$, $\hat{b} = \frac{\eta\beta}{1+2\eta\beta}$, and $\hat{c} = \frac{\eta}{1+2\eta\beta}$

4: ADMM iterations

$$\mathbf{p}_{t+1}^{k+1} = \prod_{\mathbf{p}\in\Delta_n} \left\{ \hat{a}\mathbf{p}_t - \hat{c}\nabla\phi_t(\mathbf{p}_t) + \hat{b}(\mathbf{y}^{(k)} + \mathbf{z}^{(k)} - \mathbf{w}^{(k)} - \mathbf{v}^{(k)}) \right\} \tag{6.12}$$

$$\mathbf{y}_{|g}^{(k+1)} = S_{\lambda_1/\beta}(\mathbf{p}_{|g}^{(k+1)} - \mathbf{p}_{t|g} + \mathbf{w}_{|g}^{(k)}), \quad \forall g \in \mathcal{G} \tag{6.13}$$

$$\mathbf{z}^{(k+1)} = S_{\lambda_2/\beta}(\mathbf{p}_{t+1}^{k+1} - \mathbf{p}_t + \mathbf{v}^k) \tag{6.14}$$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + (\mathbf{p}^{(k+1)} - \mathbf{y}^{(k+1)} + \mathbf{w}^{(k)}) \tag{6.15}$$

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + (\mathbf{p}^{(k+1)} - \mathbf{p}_t - \mathbf{z}^{(k+1)} + \mathbf{v}^{(k)}) \tag{6.16}$$

where $\prod_{\Delta_n}$ is the projection to the simplex and S_ρ is the shrinkage operator.

5: Continue until **Stopping Criteria** is satisfied

Algorithm 4 Portfolio Selection with Group Sparsity

1: Input $\mathcal{G}, \lambda_1, \lambda_2, \eta, \beta$; Transaction cost γ

2: Initialize $p_{1,g} = \frac{1}{|\mathcal{G}|}, g = 1, \ldots, |\mathcal{G}|; p_0 = p_1; S_0^\gamma = 1$

3: For $t = 1, \ldots, T$

4: Receive $\mathbf{x}_t$, the vector of price relatives

5: Compute cumulative wealth: $S_t^\gamma = S_{t-1}^\gamma \times (\mathbf{p}_t^\top \mathbf{x}_t) - \gamma \times S_{t-1}^\gamma \times \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1$

6: Update portfolio: $\mathbf{p}_{t+1} = \text{OLU-GS}(\mathbf{p}_t, \mathbf{x}_t, -\frac{\mathbf{x}_t}{\mathbf{p}^\top \mathbf{x}_t}, \mathcal{G}, \lambda_1, \lambda_2, \eta, \beta)$

7: end for

where $\mathbf{q} = \mathbf{p}^{(k+1)} + \mathbf{w}^{(k)}$, $\tilde{\lambda} = \frac{\lambda_1}{\beta}$ and $\mathbf{y}_{|g}$ is a vector of length $n \times 1$ whose coordinates are equal to those of $\mathbf{y}$ for indices in the set g and 0 otherwise.

z-update: We obtain a closed form solution for $\mathbf{z}^{k+1}$ by using the soft-thresholding operator $S_\rho(a)$ [19].

w and v updates: The updates for $\mathbf{w}^{(k+1)}$ and $\mathbf{v}^{(k+1)}$ are already in closed form.

We iterate over the updates until convergence according to the stopping criteria in [19]. Algorithm 3 summarizes the ADMM updates for OLU-GS. Algorithm 4 outlines our algorithm for computing the transaction cost-adjusted wealth S_T^γ, where γ is a proportional transaction cost [41].

6.4 Regret Analysis

We consider updates of the following form:

$$\mathbf{p}_{t+1} := \operatorname*{argmin}_{\mathbf{p}\in\mathcal{P}} \eta\langle\nabla\phi_t(\mathbf{p}_t), \mathbf{p}\rangle + \eta r(\mathbf{p}) + \eta\lambda_2\|\mathbf{p}-\mathbf{p}_t\|_1 + d_\psi(\mathbf{p}, \mathbf{p}_t) \ , \tag{6.19}$$

where $r(\cdot)$ is any non-smooth regularizer and d_ψ is a Bregman divergence. Note, that our analysis is different from [48], because of the presence of the $\|\mathbf{p}-\mathbf{p}_t\|_1$ term in our updates and is different from the update we considered in Section 5.4 because of the additional $r(\cdot)$ term. Our OLU-GS updates in Section 6.3 are a special case of (6.19) by setting $r(\mathbf{p}) = \lambda_1\Omega(\mathbf{p})$ and $d_\psi(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2$. In this section, we prove fixed regret bounds for two cases: when the (1) ϕ_ts are general convex functions and (2) ϕ_ts are strongly convex. Moreover, we prove shifting bounds, where the comparator class can itself change over time when the ϕ_ts are general convex functions. All proofs for the following lemmas and theorems are in Appendix B.1.

6.4.1 Fixed Regret

We first focus on the case where the comparator class is fixed, i.e., $\mathbf{p}^*$ is the minimizer of $\sum_{t=1}^T \phi_t$ in hindsight as it incurs zero L_1 penalty in every iteration and we prove fixed regret bounds. We show that for general convex functions the fixed regret is $O(\sqrt{T})$ while for strongly convex functions the fixed regret is $O(\log T)$.

General Convex Functions

We assume that ϕ_t are general convex functions with bounded (sub)gradients, i.e., for any $\hat{g} \in \partial\phi_t(\mathbf{p})$ we have $\|\hat{g}\| \leq G$.

Lemma 3 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (6.19), with potentially time-varying η_t. Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ and let $||\mathbf{p}-\hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. Then, for any $\mathbf{p}^* \in \mathcal{P}$,*

$$\begin{aligned} &\eta_t[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_{t+1}) + \lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*) - \lambda_2 r(\mathbf{p}^*)] \\ &\leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\lambda_2 L + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ . \end{aligned} \tag{6.20}$$

Based on the above result, we obtain the following fixed regret bound

Theorem 4 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by (6.19). Let* ϕ_t *be a Lipschitz continuous function for which* $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$. *Then, by choosing* $\eta \propto \frac{1}{\sqrt{T}}$ *and* $\lambda_2 \propto \frac{1}{\sqrt{T}}$, *we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi(\mathbf{p}^*) - r(\mathbf{p}^*)] \leq O(\sqrt{T}) \ . \tag{6.21}$$

Strongly Convex Functions

We assume that ϕ_t are all β-strongly convex functions. For any $(\mathbf{p}, \mathbf{p}_t)$, $\phi_t(\mathbf{p}) \geq \phi_t(\mathbf{p}_t) + \langle \mathbf{p} - \mathbf{p}_t, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{\beta}{2}\|\mathbf{p} - \mathbf{p}_t\|^2$.

Lemma 4 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (6.19) with potentially time-varying* η_t *and* λ_2. *Let* $d_\psi(\cdot,\cdot)$ *be* λ*-strongly convex with respect to norm* $||\cdot||_2$, *i.e.* $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ *and* $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. *Assuming* ϕ_t *are all* β*-strongly convex, for any* $\lambda_2 < \frac{\beta}{4}$ *and any* $\mathbf{p}^* \in \mathcal{P}$, *we have*

$$\begin{aligned}&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_{t+1}) - \lambda_1 r(\mathbf{p}^*) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1] \\ &\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 - \eta_t\left(\frac{\beta}{2} - 2\lambda_2\right)\|\mathbf{p}^* - \mathbf{p}_t\|^2 \ .\end{aligned} \tag{6.22}$$

Theorem 5 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by (6.19). Let* ϕ_t *be all* β*-strongly convex and* $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$. *Then, for any* $\lambda_2 < \beta/4$, *choosing* $\eta_t = \frac{1}{\kappa t}$ *where* $\kappa \in (0, \beta - \lambda_2]$ *and with* $d_\psi(\mathbf{p}, \mathbf{p}') = \frac{1}{2}\|\mathbf{p} - \mathbf{p}'\|_2^2$, *we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi(\mathbf{p}^*) - r(\mathbf{p}^*)] \leq O(\log T) \ . \tag{6.23}$$

6.4.2 Shifting Regret

We now focus on the case where the comparator class can also shift or change over time, i.e., time-varying sequences $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ is used as the comparator class on the cumulative loss $\sum_{t=1}^T f_t$. Such a time varying sequence will incur cumulative non-zero shifting penalty penalty of the form

$$\text{shift}_q(\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*) = \sum_{t=1}^T \|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ , \tag{6.24}$$

for some suitable q-norm for $q \geq 1$. The regret bounds in this section are in terms of such shifting penalties. A specific case of interest is when the comparator class consists of constant shifting penalties, i.e.,

$$\text{shift}_q(\mathbf{p}_1^*, \ldots, \mathbf{p}_T^*) \leq c \ . \tag{6.25}$$

We show that for this important special case, the shifting regret bounds are of the same order as the regret bounds with a fixed comparator, i.e., $\mathbf{p}_t^* = \mathbf{p}^*$, as discussed earlier.

General Convex Functions

We assume that ϕ_t are general convex functions with bounded (sub)gradients, i.e., for any $\hat{g} \in \partial\phi_t(\mathbf{p})$ we have $\|\hat{g}\| \leq G$.

Theorem 6 *Let $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ be any sequence of portfolios serving as a comparator in (6.19). Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. For $\eta_t = \eta = \frac{c_1}{\sqrt{T}}$ and $\lambda_2 = \frac{c_2}{\sqrt{T}}$ for $c_1, c_2 > 0$, $\frac{1}{r} + \frac{1}{q} = 1$, and $||\nabla\psi(\mathbf{p}_t)||_r \leq \zeta$, we have*

$$\sum_{t=1}^T [\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) - \phi_t(\mathbf{p}_t^*) - \lambda_1 r(\mathbf{p}^*)] + \sum_{t=1}^T [\lambda_2 \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \lambda_2 \|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \tag{6.26}$$

$$\leq O(\sqrt{T}) + \frac{\sqrt{T}}{c_1}\left\{ d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) + \zeta \sum_{t=1}^T \|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \right\}.$$

Assuming $d_\psi(\mathbf{p}, \mathbf{p}') \leq c_3, \psi(\mathbf{p}) \leq c_4$, and the shifting penalty $\sum_{t=1}^T \|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \leq c$, the shifting regret is $O(\sqrt{T})$, which is the same order as the fixed regret.

6.5 Experiments and Results

The experiments were conducted on data taken from the NYSE and S&P 500 stock markets (refer to Section 4.3 for details on the two datasets). We used the Global Industry Classification Standard (GICS) to group the stocks in the datasets into their designated sectors. This resulted in 8 sectors and 30 stocks in the NYSE dataset and 9 sectors and 243 stocks in the S&P500 dataset (some sectors were removed if they were empty and some stocks were removed if they were the only member in a sector). Table 6.1 shows the sectors represented in the two datasets, and a couple representative companies from each sector.

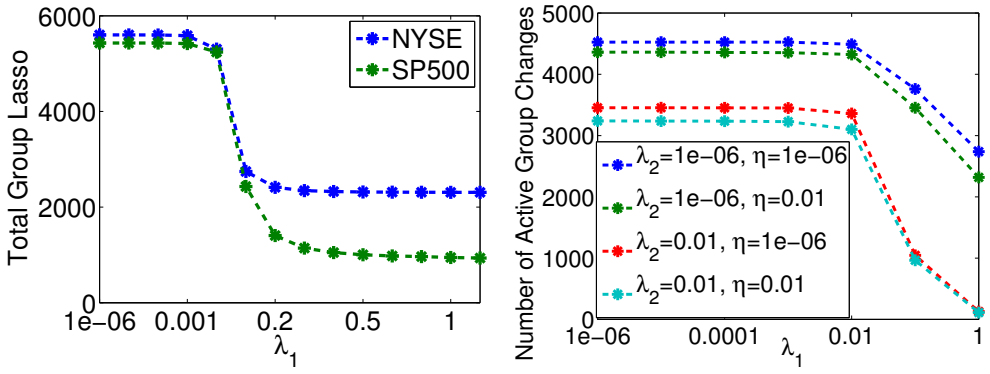
Sector	**Example Companies**
Consumer Discretionary	Nike Inc., Target Corp.
Consumer Staples	Costco Co., Beam Inc.
Energy	Chevron Corp., Noble Corp.
Financials	Equifax Inc., AFLAC Inc.
Health Care	Cerner, Pfizer Inc.
Industrials	Raytheon Co., 3M Co.
Information Tech	Apple Inc., Dell Inc.
Materials	Alcoa Inc., Ecolab Inc.
Utilities	AGL Resources, AES Corp.

Table 6.1: Overview of GICS sectors used in our datasets.

6.5.1 Methodology and Parameter Setting

In all experiments we started with \$1 as our initial investment and an initial portfolio uniformly distributed over the groups to avoid group bias. We use OLU-GS to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth for each day. The parameters consist of $\boldsymbol{\lambda}_1$: weight on group sparsity norm, $\boldsymbol{\lambda}_2$: lazy updates weight, $\boldsymbol{\eta}$: weight on the L_2 norm, and $\boldsymbol{\beta}$: the parameter for the augmentation term. For all our experiments, we set $\beta = 2$ which we found to give reasonable accuracy and use group lasso for group sparsity.

Since the two datasets are very different in nature (stock composition and duration),



(a) Total Group Lasso. (b) Number Active Group Changes.

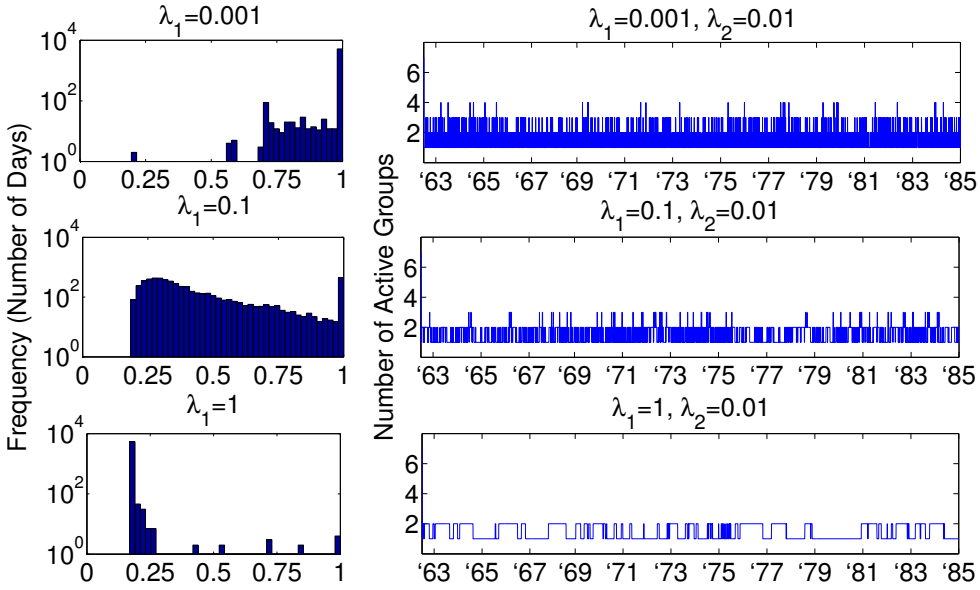
Figure 6.1: As λ_1 increases the (a) total group lasso value and (b) number of active group changes decrease.

we experimented extensively with a values of λ_1, λ_2, and η values ranging in $[1e{-}9, 1]$ to observe their effect on group sparsity and lazy updates to our portfolio. Moreover, we chose a reasonable range of γ values ranging in $[0\%, 2\%]$ to compute the proportional transaction costs incurred due to the portfolio update every day. We have illustrated some of our results with representative plots from either the NYSE or S&P500 dataset.

We use the wealth obtained (without transaction costs) from the EG algorithm with experimentally tuned parameters, a Buy-and-Hold strategy, and the best single stock as benchmarks for our experiments with initial investments of \$1. EG has been shown to outperform a uniform constantly rebalanced portfolio [69, 39]. For the Buy-and-Hold case we start with a uniformly distributed portfolio and do a hold on the positions thereafter (i.e. no trades). For the best single stock case we observe how the market performs and select the stock that has accumulated the most wealth at the end of the period. Note, in a real world situation, this strategy is infeasible since it is not possible to know the best stock *a priori*.

6.5.2 Effect of λ_1 for Group Sparsity ($\Omega(\mathbf{p})$)

The regularization parameter λ_1 for the group lasso term ($\Omega(\mathbf{p})$) is varied from $[1e{-}9, 1]$ to obtain different levels of group sparsity. The value of λ_1 has a strong effect on (a) the total group lasso penalty value, (b) the number of active groups, and (c) active groups.

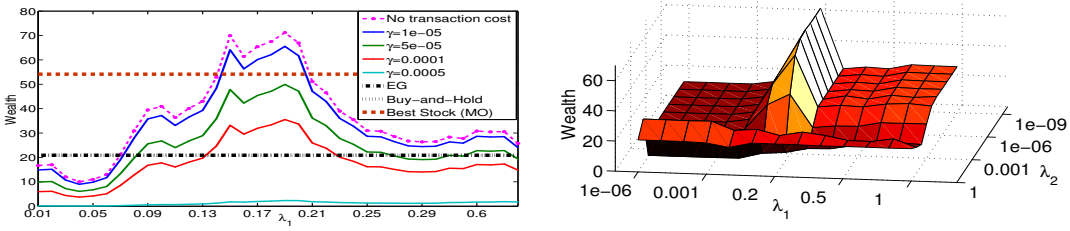


(a) Group Lasso per day. (b) Number of Active Groups.

Figure 6.2: As λ_1 increases the number of days with high group lasso value and the number of active groups decrease.

(a) Total Group Lasso penalty: Figure 6.1(a) plots the value of the total group lasso penalty ($\sum_{t=1}^{T} \Omega(\mathbf{p}_t)$) as we increase λ_1, keeping λ_2 and η fixed. For both the NYSE and S&P500 datasets, we observe that $\sum_{t=1}^{T} \Omega(\mathbf{p}_t)$ decreases as we increase λ_1, which is in conformance with our objective. Since the two datasets are different in terms of the total number of stocks and the number of stocks composing each sector, Figure 6.1(a) specifically illustrates how to choose λ_1 to attain a desired level of sparsity for each of the datasets. Figure 6.2(a) plots a histogram of the total per day group lasso penalty with increasing λ_1 values for the S&P500. It is fairly evident that there is a decrease in the number of days with high group lasso penalty as λ_1 increases.

(b) Active Groups: We compute the *active groups* each day by selecting the groups in which the majority (80%) of the wealth is invested. Figure 6.2(b) plots the number of active groups per day for the NYSE dataset. With $\lambda_1 = 1\text{e}{-3}$, OLU-GS picks up to 4 groups on a particular day. For a higher value of $\lambda_1 = 1$ a maximum of 2 groups are selected to invest in. In particular, the two sectors picked are Basic Materials and Consumer Discretionary.



(a) NYSE transaction cost-adjusted wealth.

(b) NYSE Wealth as a function of λ_1 and λ_2.

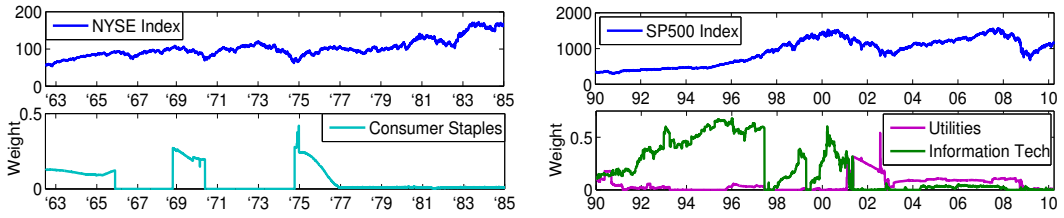
Figure 6.3: (a) OLU-GS returns more than competing algorithms even with transaction costs. (b) There exists a parameter range that gives good wealth performance.

(c) Active Groups Changes: Figure 6.1(b) plots the total number of times the *active groups* change for the NYSE dataset (over 22 years) as λ_1 increases. We consider an active group change as anytime the group composition changes. The individual line plots indicate different values of λ_2 and η. For λ_1 between 1e$-$6 to 1e$-$2: with low values for λ_2, the total number of changes in the active groups are quite high but for a higher value of $\lambda_2 = 1\text{e}{-2}$ we see a decrease in the number of active group changes illustrating the portfolio laziness. With values of $\lambda_1 \geq 1e-2$ we see a dramatic drop in the number of active group changes and high λ_2 values reemphasize this behavior.

6.5.3 Wealth and Group Sparsity

To evaluate the practical application of our proposed algorithm, we now analyze its performance when calculating the transaction cost-adjusted cumulative wealth. Figure 6.3(a) shows how the choice of different λ_1 values affect the transaction cost-adjusted cumulative wealth for the NYSE dataset (for a fixed λ_2 and η value). Figure 6.3(b) demonstrates that there exists a combination of λ_1 and λ_2 values which make an optimal choice between group sparsity and lazy updates.

EG, Buy-and-Hold, Best Single Stock: We compared the total wealth without transaction costs of OLU-GS with that of EG, a Buy-and-Hold strategy, and the best performing single stock. These strategies are plotted as horizontal lines and we can see that for the NYSE dataset EG returns \$20.89, Buy-and-Hold returns \$20.88, and the best single stock, Phillip Morris (MO), returns \$54.14. In comparison, OLU-GS returns \$71.18 without transaction costs. OLU-GS returns over 3x as much wealth for



(a) Consumer staples picked during bear markets.

(b) Utilities/Info Tech selected during bear/bull markets.

Figure 6.4: Picking non-cyclic sectors, sectors that tend to perform well during economic downturns, during the 1970s and dot-com bear markets and cyclic sectors, sectors that perform well during economic booms, during the dot-com bull market.

the NYSE dataset as EG or Buy-and-Hold do and about $15 more than the best stock. Figure 6.3(a) also shows that OLU-GS is able to return more wealth than EG and Buy-and-Hold with reasonable transaction costs (0.001%, 0.005%, and 0.01%).

6.5.4 Switching Sectors

We desire that OLU-GS is able to identify the best sectors automatically. We illustrate the strength of OLU-GS in selecting the best sectors with two examples. A recurring trend that we observe from our experiments with both the NYSE and S&P500 datasets is that OLU-GS selects stocks in Consumer Staples during the bear markets. Figure 6.4(a) clearly shows that OLU-GS selects and invests in this defensive sector during the historical bear markets of 1969-1971 and 1975-1977. Another example of a defensive or non-cyclic sector is Utilities. Figure 6.4(b) shows that the weight on the Utilities sector sees a considerable increase during the dot-com crash. This is interesting because unlike other areas of the economy, even during bear markets, the demand for Consumer Staples and Utilities do not slow down. These sectors consist of stocks which are defensive in nature and usually outperform the S&P500 Index during bearish markets and under-perform during bullish markets. Sectors like Information Technology and Financials comprise of cyclical stocks which are sensitive to market movements and can take advantage of the bullish markets. In Figure 6.4(b), the Information Technology sector is picked up during the bullish markets which preceded the dot-com bubble.

Chapter 7

Online Structured Diversification

So far, we have designed and analyzed algorithms which control transaction costs in Chapter 5 and take advantage of market sectors in Chapter 6. However, one key aspect of many resource allocation problems especially in algorithmic trading is risk and ways to control risk. In this chapter, we introduce a framework and algorithm which alleviates various measure of risk through diversifying one's resource allocation. The work in this chapter first appeared as a peer-reviewed conference paper [76].

7.1 Introduction

In sequential decision making problems and specifically resource allocation problems, one must consider some notion of risk, and suitable ways to alleviate risk. One common approach in financial trading is to diversify the asset portfolio. Specifically in portfolio selection, putting all of one's money in one or a few stocks is considered risky, since those few stocks may not perform well over time. In several such settings, the risk is often structured and calls for structured diversification strategies. For example, in portfolio selection, one considers investing in market sectors, such as energy, technology, utilities, etc., which gives a structure beyond individual stocks and may lead to more returns as we saw in Chapter 6. Often, stocks within a sector move together in response to external influences so that investing in just one sector can be risky. A structurally diverse portfolio would invest in multiple sectors to alleviate risk.

In addition to the literature on portfolio selection we discussed in Section 4.2, there is

also a large literature on general resource allocation where several papers pose the problem as an auction [45], mechanism design problem [8], Markov Decision Process [46, 99], matching problem [62], and planning problem [63, 126]. However, few have considered risk when computing resource allocations in an online fashion [50, 71, 72]. Additionally, not much work [40] has been done to consider risk in the common online convex optimization framework used for most online portfolio selection algorithms.

Overview of Contributions: We make two main contributions. First, we introduce a novel formulation for online resource allocation with structured diversification (ORASD). The formulation considers groups over the assets of interest, such as stocks, where the groups can be of different sizes and they can overlap. We also outline a way in which these groups can be inferred from the data, e.g., based on their correlation structures. The key novel component of our formulation is the $L_{(\infty,1)}$ group norm, which is rather different from $L_{(1,p)}$ type group norms typically used for overlapping group Lasso and related problems. We pose the ORASD problem as a suitable online convex optimization problem with the constraint that the $L_{(\infty,1)}$ group norm of the resource allocation vector has to be bounded within a pre-specified limit. Such a constraint ensures that no single group gets a large share of the resource. Further, unlike overlapping group Lasso, there is no sparsity restriction on the resource allocation vector, so that one can invest in all assets if that helps the resource allocation objective.

Second, we instantiate the problem in the context of portfolio selection and propose an efficient algorithm based on the alternating direction method of multipliers (ADMM). We illustrate the effectiveness through extensive experiments on two benchmark datasets and several baselines from the existing literature.

7.2 Problem Formulation

In this section, we introduce a framework for resource allocation with structured diversity, and consider the online resource allocation problem under such structured diversity. In Section 7.3, we consider the online portfolio selection problem as an instance of such diversified resource allocation.

7.2.1 Structured Diversity with $L_{(\infty,1)}$ Norm

We consider an online resource allocation problem over n objects, where at each round t the goal is find a probability distribution $\mathbf{p}_t \in \Delta_n$, the n-dimensional probability simplex, which determines how to split up a resource over the n objects such that a certain (convex) objective $f_t : \Delta_n \to \mathbb{R}$ is minimized. For example, in the context of online portfolio selection, the n objects can be different stocks, and $\mathbf{p}_t$ is an investment strategy on day t, i.e., what fraction of one's money should one invest in a stock. The basic idea of diversification is to avoid putting all of the resources on one asset. A simple way to accomplish this is to put a cap or upper bound on the amount of resources which can be put on any asset, i.e., $p(i) \leq \kappa$ for some $\kappa \in [0, 1]$. A potential issue with such an approach is that there may be structural dependencies between the assets. For example, in the context of portfolio selection, selling some Google stocks to buy Apple stocks may not accomplish the goals of diversification when the entire tech sector goes down. In this example, Google and Apple as assets can be considered structurally related, both being part of the tech sector. The goal of structured diversification is to develop strategies which explicitly consider such structurally related groups and diversify across them.

For the development, we assume knowledge of such structurally related groups, and outline approaches for inferring such groups directly from the data in Section 7.3 in the context of portfolio selection. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be the set of groups, where g_i is the set of indices for the n_i assets in the group. The groups can be of different sizes, $|g_i| = n_i$, and the groups may overlap, i.e., $g_i \cap g_j \neq \emptyset$.

Given such group structure, we introduce the $L_{(\infty,1)}$ group norm which plays a key role in structured diversity. In general, for any $\mathbf{x} \in \mathbb{R}^n$ and a set of (possibly overlapping, different sized) groups $\mathcal{G}$, the $L_{(\infty,1)}$ group norm is defined as:

$$\|\mathbf{x}\|_{(\infty,1)}^{\mathcal{G}} = \left\| \left[\|\mathbf{x}_{g_1}\|_1 \ldots \|\mathbf{x}_{g_m}\|_1 \right]^\top \right\|_\infty \tag{7.1}$$

where $\mathbf{x}_{g_i}$ is a vector of length n with value equal to $\mathbf{x}$ for indices in g_i and 0 otherwise. The $L_{(\infty,1)}$ group norm is determined by the largest L_1 norm over all groups in $\mathcal{G}$. This norm is rather different in character compared to the $L_{(1,2)}$ norm [75], used in the context of overlapping group Lasso and multi-task learning, as well as the $L_{(1,\infty)}$ group norm [96]. The $L_{(1,p)}$ group norms accomplish sparsity over the groups, i.e., certain groups can become all zeros. Such a structure is not desirable in the context

of diversified portfolio selection, since one wants to distribute the resource to multiple groups for diversification. In fact, for $\|\mathbf{x}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$, each individual group g_i has the flexibility of increasing their L_1 norm to κ without changing the norm ball (penalty). Similarly, one could also consider any $L_{(\infty,p)}$ group norm, which looks at the maximum over L_p norms over groups. In the context of resource allocation, since the key object of interest is a probability distribution, we work with the $L_{(\infty,1)}$ group norm and use it as a constraint in our problem framework.

7.2.2 Online Resource Allocation Framework

We continue to consider the online learning framework in this chapter. Specifically, the problem proceeds in rounds where in round t the algorithm has to pick a solution from a feasible set, $\mathbf{p}_t \in \Delta_n$, Nature selects a convex loss functions $\phi_t(\cdot)$, the algorithm observes the loss function, and incurs a loss of $\phi_t(\mathbf{p}_t)$. In addition to being feasible, we add two additional requirements on $\mathbf{p}_t$: (1) $\mathbf{p}_t$ needs to stay close to $\mathbf{p}_{t-1}$ since we do not want the resource allocation to change drastically in every time step, which may have a cost associated with it as in the previous two chapters, and (2) $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$ for $\kappa \in [0, 1]$ so that structural diversity is maintained over time. Thus, the sub-problem at time t takes the form

$$\min_{\mathbf{p}\in\Delta_n} \ \eta\, \phi_t(\mathbf{p}) + \Omega(\mathbf{p}, \mathbf{p}_{t-1}) \quad \text{s.t. } \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa \ , \tag{7.2}$$

where ϕ_t denotes a suitable loss function and $\Omega(\cdot, \cdot)$ is a convex penalty function which measures the change in $\mathbf{p}_t$. Again, over T rounds we would like to minimize the constrained cumulative loss

$$\sum_{t=1}^{T} \eta\, \phi_t(\mathbf{p}_t) + \Omega(\mathbf{p}_t, \mathbf{p}_{t-1}) \quad \text{s.t. } \|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa \ . \tag{7.3}$$

However, in the online setting, absolute minimization of (7.3) is not feasible since we do not know the sequence of ϕ_t a priori. Again, we instead consider the fixed regret such that over T rounds we compute a sequence of $\mathbf{p}_t$ such that the regret is sublinear in T, i.e.,

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}^*} \sum_{t=1}^{T} f_t(\mathbf{p}^*) = o(T) \tag{7.4}$$

where $f_t(\mathbf{p}) = \eta\phi_t(\mathbf{p}) + \Omega(\mathbf{p}, \mathbf{p}_{t-1})$. The regret is measured with respect to the best fixed minimizer in hindsight $\mathbf{p}^*$.

As in the previous two chapters, we consider solving a linearized version of the problem obtained by a first-order Taylor expansion of f_t at $\mathbf{p}_t$ along with a proximal term, so that

$$\mathbf{p}_{t+1} := \underset{\substack{\mathbf{p}\in\Delta_n \\ \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)}\leq\kappa}}{\operatorname{argmin}} \quad \eta\langle\mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \lambda\Omega(\mathbf{p}, \mathbf{p}_t) + d(\mathbf{p}, \mathbf{p}_t) \ , \tag{7.5}$$

where $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2$ is the proximal term and $\eta, \lambda \geq 0$ are tunable parameters.

7.3 Algorithm

Given the resource allocation with structured diversity framework, we can now view the online portfolio selection problem (Chapter 4) as a special case of our framework by setting $\phi_t(\mathbf{p}_t) = -\log(\mathbf{p}_t^\top \mathbf{x}_t)$, and $\Omega = \|\mathbf{p} - \mathbf{p}_t\|_1$. As discussed in the previous two chapters, the L_1 penalty term on the difference of two consecutive portfolios measures the fraction of wealth traded and encourages lazy updates to the portfolio to limit transaction costs. The parameter λ controls the amount that can be traded every day. The $L_{(\infty,1)}$ penalty term forces the portfolio to spread out the investment amongst the groups. The level of diversification depends on the value of κ.

We propose a primal-dual algorithm for solving (7.5) in this context. Additionally, since we allow overlapping groups, we will solve (7.5) via lifting by adding a consensus constraint $\mathbf{p}_{g_i} = \mathbf{S}_{g_i}\mathbf{p}\ \forall i$ where $\mathbf{p}_{g_i}$ is a vector of length n with value equal to $\mathbf{p}$ for indices in g_i and 0 otherwise and $\mathbf{S}_{g_i}$ is a known diagonal matrix with $\mathbf{S}_{g_i}(j,j) = 1$ if element j is in group g_i and 0 otherwise. The online portfolio selection with structured diversification problem is now

$$\mathbf{p}_{t+1} := \underset{\substack{\mathbf{p}\in\Delta_n \\ \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)}\leq\kappa \\ \mathbf{S}_{g_i}\mathbf{p}=\mathbf{p}_{g_i}\ \forall i}}{\operatorname{argmin}} \quad \eta\langle\mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2 + \lambda\|\mathbf{p} - \mathbf{p}_t\|_1 \ . \tag{7.6}$$

(7.6) consists of smooth and non-smooth terms in the objective with $L_{(\infty,1)}$ and linear constraints. We can use ADMM to solve this problem by introducing auxiliary variable

Algorithm 5 ORASD Algorithm with ADMM

1: Input $\mathbf{p}_t, \mathbf{x}_t, \mathbf{S}_{g_1,\ldots,g_m}, \eta, \lambda, \beta$

2: Initialize $\mathbf{p}, \mathbf{p}_{g_i}, \mathbf{z}, \mathbf{u}_i, \mathbf{v} \in \mathbf{0}^n, k = 0$

3: Set $\hat{\mathbf{a}} = \left((1+\beta)I + \beta\left(\mathbf{S}_{g_1} + \cdots + \mathbf{S}_{g_m}\right)\right)^{-1}$

4: ADMM iterations

$$\mathbf{p}^{k+1} = \prod_{\Delta_n}\left(\hat{\mathbf{a}}\frac{\eta\mathbf{x}_t}{\mathbf{p}_t^\top\mathbf{x}_t} + \hat{\mathbf{a}}(1+\beta)\mathbf{p}_t + \hat{\mathbf{a}}\beta\Big(\mathbf{S}_{g_1}(\mathbf{p}_{g_1}^k - \mathbf{u}_1^k) + \cdots + \mathbf{S}_{g_m}(\mathbf{p}_{g_m}^k - \mathbf{u}_m^k) + \mathbf{z}^k - \mathbf{v}^k\Big)\right) \tag{7.8}$$

$$\mathbf{p}_{g_i}^{k+1} = \prod_{\|\cdot\|_1 \le \kappa}\Big(\mathbf{S}_{g_i}\mathbf{p}^{k+1} + \mathbf{u}_i^k\Big)\,\forall i \tag{7.9}$$

$$\mathbf{z}^{k+1} = S_{\lambda/\beta}(\mathbf{p}^{k+1} - \mathbf{p}_t + \mathbf{v}^k) \tag{7.10}$$

$$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i}^{k+1}\,\forall i \tag{7.11}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z}^{k+1} \tag{7.12}$$

where $\prod_{\Delta_n}$ is the projection to the simplex and S_ρ is the shrinkage operator.

5: Continue until **Stopping Criteria** is satisfied

$\mathbf{z} = \mathbf{p} - \mathbf{p}_t$ and moving the inequality constraint into the objective function if we let $h(\mathbf{p}_{g_i}) = \mathbb{1}_{(\|\mathbf{p}_{g_i}\|_1 \le \kappa)}$ where $\|\mathbf{p}\|_{(\infty,1)}^{\mathcal{G}} \le \kappa \equiv \|\mathbf{p}_{g_i}\|_1 \le \kappa\ \forall i$

$$\mathbf{p}_{t+1} := \underset{\substack{\mathbf{p}\in\Delta_n \\ \mathbf{S}_{g_i}\mathbf{p}=\mathbf{p}_{g_i}\ \forall i \\ \mathbf{p}-\mathbf{p}_t=\mathbf{z}}}{\operatorname{argmin}} \eta\langle\mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\|\mathbf{z}\|_1 + \sum_{i=1}^m h(\mathbf{p}_{g_i}). \tag{7.7}$$

The ADMM formulation in (7.7) naturally lets us decouple the non-smooth terms from the smooth terms, which is computationally advantageous. The augmented Lagrangian for (7.7) is

$$\begin{aligned} L(\mathbf{p}, \mathbf{p}_{g_1\ldots g_m}, \mathbf{z}, \mathbf{u}_{1\ldots m}, \mathbf{v}) = \eta\langle\mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\|\mathbf{z}\|_1 + \sum_{i=1}^m h(\mathbf{p}_{g_i}) \\ + \frac{\beta}{2}\sum_{i=1}^m \|\mathbf{S}_{g_i}\mathbf{p} - \mathbf{p}_{g_i} + \mathbf{u}_i\|_2^2 + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}+\mathbf{v}\|_2^2 \end{aligned} \tag{7.13}$$

where $\mathbf{u}$ and $\mathbf{v}$ are scaled dual variables. ADMM consists of the following iterations

$$\begin{aligned} \mathbf{p}^{k+1} := \underset{\mathbf{p}\in\Delta_n}{\operatorname{argmin}}\ \eta\langle\mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \frac{\beta}{2}\sum_{i=1}^m \|\mathbf{S}_{g_i}\mathbf{p} - \mathbf{p}_{g_i}^k + \mathbf{u}_i^k\|_2^2 \\ + \frac{\beta}{2}\|\mathbf{p}-\mathbf{p}_t-\mathbf{z}^k+\mathbf{v}^k\|_2^2 \end{aligned} \tag{7.14}$$

Algorithm 6 Diversified Online Portfolio Selection

Input η, λ, β; Transaction cost γ, Days lag num_{lag}
Initialize $\mathbf{p}_0(i) = \frac{1}{n}\ i = 1, \ldots, n, S_0^\gamma = \$1$
For $t = 1, \ldots, T$
  Receive $\mathbf{x}_t$, the vector of price relatives
  Compute wealth: $S_t^\gamma = S_{t-1}^\gamma \left(\mathbf{p}_t^\top \mathbf{x}_t - \gamma \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1\right)$
  If $t \leq num_{lag}$
    $\mathbf{p}_{t+1}(i) = \frac{1}{n}\ i = 1, \ldots, n$ (uniform portfolio)
  Else
    Compute groups: $\mathcal{G}$
    Update: $\mathbf{p}_{t+1} = \text{ORASD}(\mathbf{p}_t, \mathbf{x}_t, \mathcal{G}, \eta, \lambda, \beta)$
end for

$$\mathbf{p}_{g_i}^{k+1} := \text{argmin}_{\mathbf{p}_{g_i}}\ h(\mathbf{p}_{g_i}) + \frac{\beta}{2}\|\mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i} + \mathbf{u}_i^k\|_2^2\ \forall i \tag{7.15}$$

$$\mathbf{z}^{k+1} := \text{argmin}_{\mathbf{z}}\ \lambda\|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z} + \mathbf{v}^k\|_2^2 \tag{7.16}$$

$$\mathbf{u}_i^{k+1} := \mathbf{u}_i^k + \mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i}^{k+1}\ \forall i \tag{7.17}$$

$$\mathbf{v}^{k+1} := \mathbf{v}^k + \mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z}^{k+1}\ . \tag{7.18}$$

p-update: We solve for $\mathbf{p}$ be taking the gradient of (7.14) with respect to $\mathbf{p}$, setting it to zero, and solving to get the closed form update of $\mathbf{p}$ as

$$\mathbf{p} = \prod_{\Delta_n}\left(\hat{\mathbf{a}}\frac{\eta\mathbf{x}_t}{\mathbf{p}_t^\top\mathbf{x}_t} + \hat{\mathbf{a}}(1+\beta)\mathbf{p}_t + \hat{\mathbf{a}}\beta\big(\mathbf{S}_{g_1}(\mathbf{p}_{g_1}^k - \mathbf{u}_1^k) + \cdots + \mathbf{S}_{g_m}(\mathbf{p}_{g_m}^k - \mathbf{u}_m^k) + \mathbf{z}^k - \mathbf{v}^k\big)\right) \tag{7.19}$$

for $\hat{\mathbf{a}} = ((1+\beta)I + \beta\left(\mathbf{S}_{g_1} + \ldots + \mathbf{S}_{g_m}\right))^{-1}$ and $\prod_{\Delta_n}$ is a projection to Δ_n following [47].

$\mathbf{p}_{g_i}$-updates: We solve for each $\mathbf{p}_{g_i}$ in parallel by projecting to the $L_1 1$ ball of radius κ

$$\mathbf{p}_{g_i} = \prod_{\|\cdot\|_1 \leq \kappa}\left(\mathbf{S}_{g_i}\mathbf{p}^{k+1} + \mathbf{u}_i^k\right)\ . \tag{7.20}$$

z-update: We solve for $\mathbf{z}$ by using the soft-thresholding operator $S_\rho(a)$ [75]

$$\mathbf{z} = S_{\lambda/\beta}(\mathbf{p}^{k+1} - \mathbf{p}_t + \mathbf{v}^k)\ . \tag{7.21}$$

$\mathbf{u}_i$ and $\mathbf{v}$ updates: The $\mathbf{u}_i$ and $\mathbf{v}$ updates are in closed form and the $\mathbf{u}_i$s can be computed in parallel.

Algorithm 5 shows the complete ADMM based algorithm with the closed form updates. The stopping criteria for the ORASD algorithm is based on the primal and dual residuals from [19]. Algorithm 6 is our diversified online portfolio selection algorithm. It uses the ORASD algorithm to compute $\mathbf{p}_{t+1}$. It takes in an additional parameter γ which is a fixed percentage charged for the total amount of transaction every day. S_t^{γ} is the transaction cost-adjusted cumulative wealth gain at the end of t days.

7.4 Regret Bound

We sequentially invest with the diverse portfolios $\mathbf{p}_1, \cdots, \mathbf{p}_T$ obtained from Algorithm 6 and on day t suffer a loss $f_t(\mathbf{p}_t) = \eta\phi_t + \lambda\|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1$, where $\phi_t = -\log(\mathbf{p}_t^{\top}\mathbf{x}_t)$. Our goal is to minimize the *regret* with respect to the best fixed portfolio $\mathbf{p}^*$ in hindsight.

Theorem 7 *Let $\mathbf{p}^* \in \mathcal{P}$ be the fixed portfolio obtained from $\min_{\mathbf{p}} \sum_{t=1}^{T} \phi_t(\mathbf{p})$. For $\eta = \frac{1}{\sqrt{T}}$, $\lambda = \frac{1}{t}$, and $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$, the regret can be bounded as,*

$$\sum_{t=1}^{T} \phi_t(\mathbf{p}_t) + \sum_{t=2}^{T} \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1 - \sum_{t=1}^{T} \phi_t(\mathbf{p}^*) \leq O(\sqrt{T}), \tag{7.22}$$

where ϕ_t is a Lipschitz continuous convex function and the sequence $\mathbf{p}_t$ and the fixed optimal portfolio $\mathbf{p}^*$ all lie in $\mathcal{P} := \{\mathbf{p} \,|\, \mathbf{p} \in \Delta_n, \|\mathbf{p}\|_{(\infty,1)}^{\mathcal{G}} \leq \kappa\}$.

The theorem follows directly from Theorem 1.

7.5 Experiments and Results

The experiments were conducted on data taken from the NYSE and S&P 500 datasets (refer to Section 4.3 for more details).

7.5.1 Methodology and Parameter Setting

In all our experiments we start with \$1 as our initial investment and an initial portfolio which is uniformly distributed over all the stocks. We use Algorithm 6 to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth each day.

For our experiments, we utilize the follow method for computing the groups. For the previous num_{lag} days, we compute the correlation graph C over the stocks. To compute structurally related groups, we set a correlation threshold ϵ on the edges of the graph and if an edge has weight $< \epsilon$ then it is removed resulting in C_ϵ. For each stock in C_ϵ, we construct a group around it by including its neighbors within k hops away. For example, if $k = 1$ then we construct group g_i by including stock s_i and only its directly connected neighbors in the group. As such, there will be exactly n groups but the size of each group may vary. For our experiments, we allow the groups to change each day.

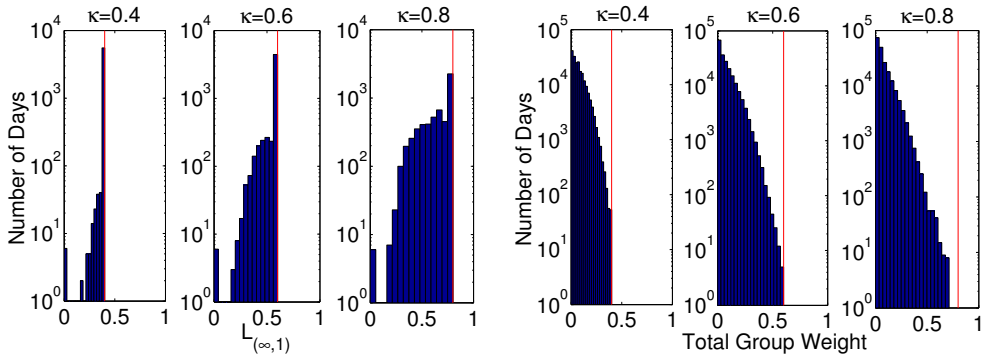
Since the two datasets are very different in nature (stock composition and duration), we experimented with various parameter values. We found stable behavior across the following range of parameters: $num_{lag} \in \{5, 10, 15\}$, $\epsilon = \{0.90, 0.95\}$, and $\beta = 2$. Additionally, we experimented extensively with a large range of values for η and λ in $[10^{-6}, 10^{3}]$ and values for κ in $[0.1, 1.0]$ to observe their effect on our portfolio. Moreover, we chose a reasonable range of γ values in $[0\%, 2\%]$ to compute the proportional transaction costs incurred due to the portfolio update every day. We have illustrated some of our results with representative plots from either the NYSE or S&P500 dataset.

We use the wealth obtained from OLU [41], EG [69], a uniform constant rebalanced portfolio (U-CRP), and a buy-and-hold strategy as benchmarks for our experiments. For U-CRP, we make trades to rebalance the portfolio at the end of each day after the market movement has driven it away from the uniform distribution. For the buy-and-hold case we start with a uniformly distributed portfolio and do a hold on the positions thereafter, i.e., no trades. We do not consider algorithms such as Anticor [15] or OLMAR [87], which are good heuristics but without regret bounds.

7.5.2 Effect of the $L_{(\infty,1)}$ group norm and κ

The $L_{(\infty,1)}$ group norm encourages diversity between the groups and sparsity within the groups. This structure is further effected by the value of κ which has direct control over the level of diversity. κ has an effect on (a) the value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ each day, (b) the group weight $\|\mathbf{p}_{g_i}\|_1 \; \forall i$, and (c) the number of active groups.

(a) Value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ With the diversity inducing constraint $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$ we are encouraging different levels of diversity depending on the value of κ. From Figure 7.1(a),



(a) Total value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ with vertical red lines marking the value of κ.

(b) Total group weight $\|\mathbf{p}_{g_i}\|_1 \; \forall i$ with vertical red lines marking the value of κ.

Figure 7.1: (a) With an aggressive trading strategy, the total value of the $L_{(\infty,1)}$ penalty follows the increasing κ. (b) With low κ the algorithm is forced to diversify but less so as κ increases.

we can see the effect κ has on $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ with $\eta = 100$ and $\lambda = 10^{-6}$. With a low κ value of 0.4, $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ is small with many days seeing the value exactly equal to κ. As we increase κ, we see that the value moves along with κ with many days seeing the value equal κ. This behavior is consistent with aggressive trading due to the high η and low λ values. ORASD tries to invest as much wealth as allowed into groups that performed well in the past. With higher λ this behavior is not as aggressive and the value more slowly moves with κ and becomes more spread out.

(b) Group Weight $\|\mathbf{p}_{g_i}\|_1$ Not only is the value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ affected by κ, but so are the group weights $\|\mathbf{p}_{g_i}\|_1 \; \forall i$. Each of these are constrained to be less than or equal to κ and how close each group's weight gets to κ is of interest. If there are many days which see many groups with weight close to κ and many groups with little weight then this suggests ORASD is focusing on a few groups to invest the majority of wealth in at each day. If, however, there are many days with small group weight and few with large this suggests ORASD is investing a small amount of wealth in many groups and has a more diversified portfolio.

From Figure 7.1(b) we can see with $\eta = 10^{-3}$ and $\lambda = 10^{-6}$ ORASD utilizes a more conservative trading strategy. We can see this from how slowly the distribution moves with κ and from how many days see a small amount of wealth invested in each group.

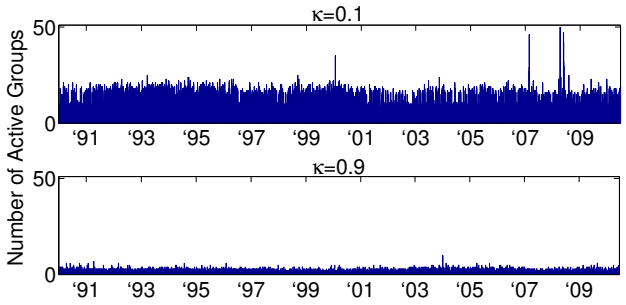


Figure 7.2: Number of active groups for the S&P500 dataset. For low κ we trade with a more conservative, diversified portfolio while with high κ we are more aggressive and risk seeking.

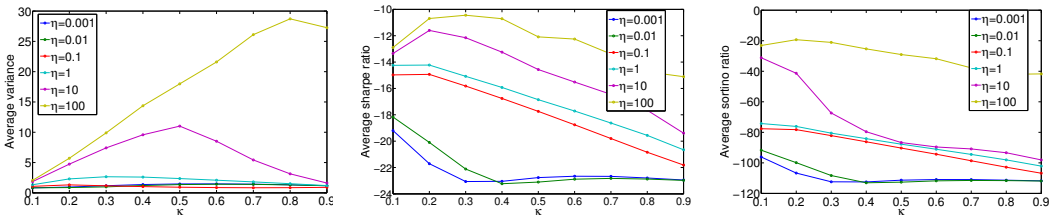
With low η and λ, ORASD does not trade aggressively and is further limited by κ. The portfolio is diversified with small κ and becomes slightly less so as we increase κ.

(c) Number of Active Groups We define an active group as a group which has a significant percent of wealth invested in it. The number of active groups can be a measure of how diverse a portfolio is. If there are many active groups this implies a diverse portfolio where few active groups does not. From Figure 7.2 we can see that for low $\kappa = 0.1$, the number of active groups is reasonably high with around 20 groups active out of 263 (about 8%). With low κ we are forcing a diverse portfolio therefore many groups have a significant amount of wealth invested in them. For high $\kappa = 0.9$, we see that the number of active groups drops to around 3 groups (about 1%). This implies that ORASD is focusing on a handful of well performing groups. This trading strategy has a potentially high reward but it also carries high risk. We can see how adjusting the value of κ can provide us the flexibility to use different trading strategies.

7.5.3 Risk and κ

Even though our online structured diversification framework (7.5) does not explicitly take risk into account, we can effectively control risk using κ as a proxy. We observe how the value of κ affects the amount of risk our portfolio is exposed to using three measures of risk: (a) covariance, (b) Sharpe ratio, and (c) Sortino ratio.

(a) Covariance We compute the covariance $\mathbf{\Sigma}_t$ using the previous num_{lag} days of price relatives. We measure the risk of a portfolio $\mathbf{p}_t$ with respect to a uniform constant



(a) Covariance risk, α_{cov} (b) Sharpe ratio, α_{sharpe} (c) Sortino ratio, $\alpha_{sortino}$

Figure 7.3: For α_{cov} with low η, the risk stays low for varying κ. For large η, the risk increases with an increasing κ. For both α_{sharpe} and $\alpha_{sortino}$, the risk-adjusted return is higher for higher η and deceases as we increase κ. From this figure, we can see that if we want to control the risk exposure, we can effectively do it by controlling κ.

rebalanced portfolio $\mathbf{u}$ as α_{cov}=$\mathbf{p}_t^\top \mathbf{\Sigma}_t \mathbf{p}_t / \mathbf{u}^\top \mathbf{\Sigma}_t \mathbf{u}$. High α_{cov} implies high risk and low α_{cov} implies low risk.

(b) Sharpe ratio The Sharpe ratio [113] measures how much the return (percent gain or loss on investment) of a portfolio compensates for the level of risk taken. It computes what can be considered as a risk-adjusted return for a given portfolio and benchmark return. It does this by measuring both the downwards and upwards volatility. A higher Sharpe ratio implies better compensation for the risk exposure. We compute the Sharpe ratio of a portfolio as α_{sharpe}=$(R-R_b)/\sqrt{var(R-R_b)}$ where R is the return for the portfolio and R_b is the benchmark return which is typically a large index such as the S&P500.

(c) Sortino ratio The Sortino ratio [114] is similar to the Sharpe ratio however it only measures the downwards volatility. Typically, upwards volatility is encouraged as we would gladly accept the price of a stock we have invested in to go up. However, the Sharpe ratio penalizes this type of volatility where the Sortino ratio does not. We compute the Sortino ratio as $\alpha_{sortino} = (R-R_b)/DR$ where DR is the standard deviation of negative returns (losses).

From Figure 7.3, we can see the behavior of the three measures of risk for varying values of η and κ with $\lambda = 10^{-6}$. For α_{cov} in 7.3(a), with low values of η the risk is low and stays low as we increase κ. However, once we have higher η values, the risk starts to increase up to a point before it starts to show a decreasing trend as we

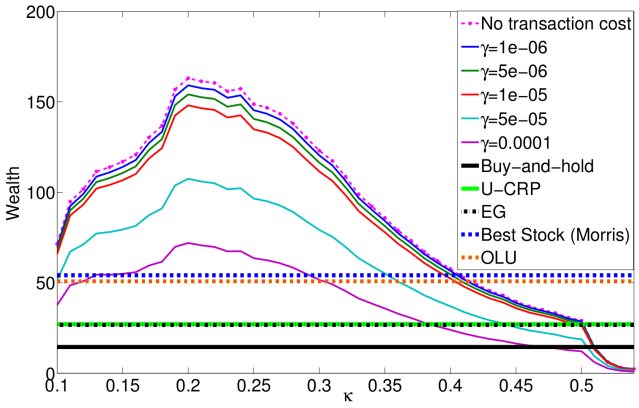


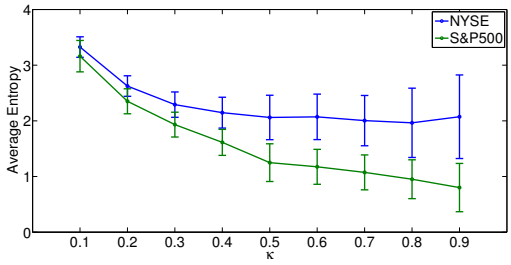
Figure 7.4: Transaction cost-adjusted wealth for the NYSE dataset. ORASD returns more wealth than competing algorithms even with transaction costs. Note, ORASD earns more wealth than OLU from Chapter 5 which was tuned for optimal wealth and without transactions costs.

increase κ. With higher η, the portfolio is focusing more on maximizing returns and is only investing in stocks that performed well in the past. This behavior does afford the portfolio the ability to earn huge amounts of wealth, however, as we can see it also exposes the investor to huge amounts of risk. In 7.3(b) and 7.3(c), we see that the risk-adjusted return increases as we increase η however, the curve trends downwards as we increase κ. This again implies that as κ increases so does the risk. From these plots we can see that κ is a good proxy to risk and we can therefore control risk by setting κ.

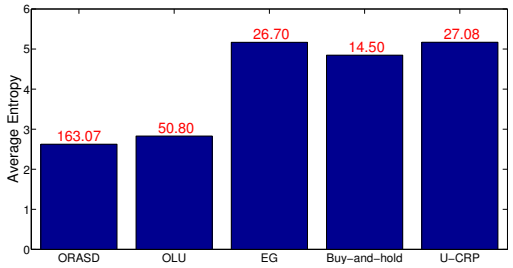
7.5.4 Transaction Cost-Adjusted Wealth

To evaluate the practical application of ORASD we analyze the performance by calculating the transaction cost-adjusted cumulative wealth. We do this for varying values of κ to get a sense of the tradeoff between risk and return. We compare the performance of ORASD to the state-of-the-art algorithms OLU, EG, U-CRP, buy-and-hold, and the best stock in hindsight (without transaction costs) with empirically determined parameters.

From Figure 7.4 we can see that for optimal η and λ values and varying values of κ, ORASD returns more wealth than all the other competing algorithms for the NYSE dataset. ORASD earns \$163.07 compared to \$54.14 for the single best stock



(a) Entropy of ORASD for η=100, λ=10^{-6}. High entropy implies the investment is more spread out. As κ increases, the entropy decreases which implies that the portfolio is concentrated on a single group of stocks.



(b) Entropy of ORASD and competing algorithms with cumulative wealth. ORASD has around the same level of diversification as OLU but returns more wealth.

Figure 7.5: Average entropy for ORASD and competing algorithms. We can see that as κ decreases so does the entropy. This indicates that the portfolio is becoming less diverse and as such may be exposed to more risk.

(Morris), \$50.80 for OLU, \$27.08 for U-CRP, \$26.70 for EG, and \$14.50 for buy-and-hold. ORASD returns over 3x as much as OLU and the best single stock we could have chosen in hindsight. We can see that ORASD also returns more wealth than the competing algorithms even with transaction costs.

7.5.5 Diversification

We saw from Figure 7.2 that one way to measure the diversity of a portfolio is by the number of active groups. Another measure commonly used is the entropy of the portfolio. Entropy measures how spread out a distribution is. If we have a portfolio with high entropy this implies that the portfolio is invested in many groups and is more diversified where with low entropy the portfolio only has investments in a few groups and is not as diversified.

From Figure 7.5(a), we can see that as κ increases the entropy decreases. This shows that with high κ the portfolio is focusing on a few groups to invest in and as such may be exposed to more risk. This again gives evidence that κ is a proxy to risk.

Finally, we want to compare how diversified ORASD is against competing algorithms. We can see from Figure 7.5(b) that ORASD has the lowest entropy but is close

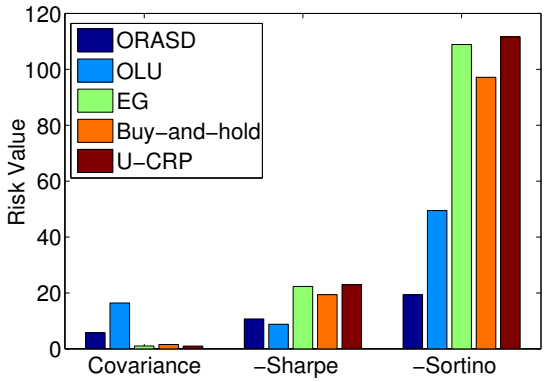


Figure 7.6: Risk comparison on the NYSE dataset. We plot the negative Sharpe and Sortino ratios therefore, a higher bar implies higher risk.

to that of OLU. However, ORASD returns more than 3x as much wealth as OLU for the same level of diversity and risk exposure.

7.5.6 Risk Comparison

From Section 7.5.4, we saw that ORASD is able to return more wealth than all of the state-of-the-art algorithms. However, as Markowitz postulated, we should seek low risk in addition to high returns. As such, we compare the risk exposure for each of the competing algorithms with optimal parameters with respect to wealth. To be consistent in the plot and have each bar represent the level of risk exposure, we have plotted the negative Sharpe and Sortino ratios since a low ratio implies a high risk relative to the return. Therefore, for each of the bar plots, a higher bar height implies higher risk.

In Figure 7.6 we can see that ORASD has lower α_{cov} than OLU but higher than the others with $\kappa = 0.2$, however, the other algorithms returned far less wealth. There is a balance between risk taken and potential returns. For α_{sharpe}, we can see that ORASD and OLU are reasonably close and both have smaller risk than the other algorithms. For $\alpha_{sortino}$, ORASD has the smallest risk. This can be explained by the fact that both the Sharpe and Sortino ratios compute the risk-adjusted return and the competing algorithms do not return much wealth so the compensation for the risk is low.

Chapter 8

Online Structured Hedging

This is the last chapter in which we consider problems which admit full information feedback. In Chapters 5-7, we have considering the online portfolio selection problem where we required the algorithm to invest all its wealth into the portfolio each day. Moreover, we required the algorithm to only invest by using money to purchase shares of a stock after which the algorithm earned money if the stock share price increased and lost if it decreased. Such a requirement will fail when the market crashes and all stocks lose value. To address such a practical issue, in this chapter we allow the algorithm to borrow money and/or shares of stock from the bank and sell such shares. After, the algorithm is can purchase the shares back and return what is owed to the bank therefore, if the market crashes and the share price falls then the algorithm earns wealth. Further, we take advantage of such an ability to reduce various measures of risk. The work in this chapter first appeared as a peer-reviewed conference paper [77].

8.1 Introduction

Existing algorithms for online resource allocation problems, in particular, online portfolio selection problems focused on budgeting resources that are currently in possession [45, 50, 63, 99, 126]. For example, financial trading algorithms only invest in the stock market using the current cash on hand. However, in several settings there are opportunities to borrow additional resources at cost of a certain interest rate to use as leverage to increase returns. For example, a trader can invest with borrowed cash from

a bank with the obligation to pay back the loan plus interest at a later date. However, such leveraged allocations are risky since gains and losses are magnified.

Moreover, several online resource allocation problems allow different allocation types or positions. For instance, when investing in the stock market, one can take a long position by purchasing shares in a company with cash on hand. Profit is made when the price of the shares increases. Additionally, one can take a short position by borrowing shares and selling them on the market at a price X and later purchasing them at a price Y and returning them. Profit is made when the price of purchasing them back is less than the price of selling them, $Y < X$, i.e., the price of the shares decreases. When such online resource allocation problems allow opposing allocation positions, one can strategically invest in both positions to reduce risk. This is often referred to as hedging and can be done by mining and utilizing correlation structures between assets to hold opposing positions in similar assets to withstand market crashes.

In this chapter, we consider the problem of hedging resource allocations using leverage. In the context of online portfolio selection, a few papers have considered leverage [4, 34, 69] through buying on margin. However, this was simulated after-the-fact and not considered in their problem setting. One attempt at including leverage and long/short positions in the problem setting was recently presented in [65]. A few preliminary experiments on one dataset were presented to illustrate their formulation worked as expected. However, they did not consider the structural dependencies between positions or utilize such structure to control risk. We build off of the setting in [65] by introducing a framework with key changes to the loss function and constraint set to allow more flexible long and short portfolios and to consider structured hedging to alleviate risk in addition to thoroughly exploring the affect similar features have on existing algorithms in the literature.

Overview of Contributions: We make two main contributions. First, we present a novel framework and algorithm for hedging online resource allocations with leverage (SHERAL). The formulation considers opposing long and short positions over assets such as stocks. The key novel components of our formulation are (1) a loss function for general leveraging and structurally dependent allocation positions and (2) a penalty function which encourages hedging between structurally dependent assets to control risk. We pose the problem as a constrained online convex optimization problem and

instantiate it in the context of online portfolio selection where one has to compute a portfolio of stocks to invest in each day.

Second, we experiment extensively with five datasets and various measures of risk. Further, we present leveraged variants of the EG algorithm [69] and show experimentally that adding leveraged long-only positions leads to orders of magnitude improvements in wealth gain, and adding both long and short positions with leverage leads to even more improvements. The results are consistent across multiple datasets, and SHERAL outperforms several existing methods for the portfolio selection problem.

8.2 Problem Formulation

In this section, we introduce a framework for structured hedging with leverage, and consider the online resource allocation problem under such structured hedging. In Section 8.4, we consider the online portfolio selection problem as an instance of such hedged resource allocation.

8.2.1 Hedging and Leverage

In many resource allocation problems, one may benefit from borrowing additional resources to increase allocation power. For example, in finance, one often buys on margin which is the act of buying shares of stock using cash that is borrowed. This is achieved by putting some amount of cash into a margin account, which acts as collateral c, after which the bank loans additional cash. When one borrows cash, interest is charged on the loan which is often represented as a percentage of the loan size, e.g., a daily interest rate of $r = 0.000245$ which is equivalent to an annual rate of 0.063. In finance, the borrowed cash used to increase potential return is often referred to as leverage.

In addition to leverage, there are often opposing allocation types or positions one may take. For instance, one can budget a fraction of the resource to an object, e.g., using cash to buy shares of stock. In finance, this is what is often referred as holding a long position in an asset. One will profit in a long position if the value of the allocation increases, i.e., the stock price increases.

In contrast, one can choose an opposing type of allocation where one will profit if the value of the allocation decreases. In finance, one such type of allocation is called a short

position, which is when one allocates a borrowed resource. One obtains a short position by placing collateral c, such as cash, into a margin account after which a resource, such as shares of a stock, is borrowed with the obligation to return it later at its current value. Often interest is charged on the borrowed resource r_b and may be earned on the collateral r_c, however in this chapter we assume the interest on the borrowed resource, collateral, and cash leverage are the same, i.e., $r_b = r_c = r$. We also assume that the value of the collateral c is equal to the value of the resource borrowed.

More specifically, a trader places X amount of cash as collateral into a margin account and then the bank borrows the trader shares of stock worth X. The trader then sells the shares at the current market price (presumably X). After the market moves, the trader purchases the shares back at the new market price Y and returns them to the bank. The trader profits if the purchase price is less than the selling price, i.e., $Y < X$.

Long and short positions act in opposition and are most frequently utilized to offset the risk of a particular allocation. For example, if two stocks s_1 and s_2 are positively correlated but highly volatile, then a trader may hold a long position in s_1 and a short position in s_2. The trader is exposed to less risk with this allocation because if both stocks crash then the trader will not lose as much since there will be a loss in the long position s_1 but a gain in the short position s_2. This is often referred to as hedging and the difficulty is in taking advantage of the structural dependencies between assets to balance high returns and exposure to risk.

8.2.2 Structured Hedging

We consider structure in resource allocation problems in the form of a graph over the assets. For n objects, the goal is to find an allocation $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{2n}$ which determines how to split up a resource amongst long and short positions over the n objects such that a certain (convex) objective $f(\mathbf{p})$ is minimized. We denote the set of indices $\ell = \{1, \ldots, n\}$ and $s = \{n+1, \ldots, 2n\}$ as the long and short positions in $\mathbf{p}$ and let $\mathbf{D}_\ell$ and $\mathbf{D}_s$ be $2n \times 2n$ diagonal matrices with $D_\ell(i,i) = 1$ for $i \in \ell$ and $D_s(i,i) = 1$ for $i \in s$ and 0 otherwise. Let $\mathbf{q}_\ell = \mathbf{D}_\ell \mathbf{p} \geq 0$ and $\mathbf{q}_s = \mathbf{D}_s \mathbf{p} \leq 0$ where the inequalities are taken element-wise. $\mathbf{q}_\ell$ and $\mathbf{q}_s$ are the long-only and short-only vectors of size $2n \times 1$ with value equal to $\mathbf{p}$ for indices in ℓ and s respectively and 0 otherwise.

For example, in the context of portfolio selection, the n objects are stocks and $\mathbf{p}$ is an investment strategy, i.e., what fraction of one's money should one put on each stock. The basic idea of hedging is to place resources in opposing positions and different assets. A simple way to accomplish this is to select a set of assets to hold long positions and another set of assets to hold short positions. A potential issue with such an approach is that there may be structural dependencies between the assets, such as being negatively correlated, which may result in losing every position held.

For example, Apple and Costco may be negatively correlated because Apple sells luxury items and Costco sells consumer staples. Since companies that sell luxury items are cyclical as we saw in Chapter 6, i.e., share price is positively correlated with economic conditions, the purchases of luxury items often slows during market crashes. In contrast, companies that sell consumer staples are non-cyclical and the purchases of consumer staple items does not slow during market crashes. Therefore, if we hold a long position in Apple and a short position in Costco during this time, we will lose in both positions.

Such structure is often hard to determine a priori and represent simply such as in groups of assets, i.e., market sectors. We can more easily capture relationships between assets via a graph where the value of the edges determines how similar the assets are. Similarity can be any suitable measure of correlation such as linear correlation, Rank correlation, etc. The goal of structured hedging is to develop hedged strategies which explicitly consider such graph structured assets.

Hedging Penalty Function

For the development, we assume knowledge of the structured graph $\mathcal{G} = (V, E)$ where $V = \{v(1), \ldots, v(n)\}$ are the nodes and $E = \{e(1), \ldots, e(m)\}$ are the edges where $e(k) = (v(i), v(j))$ if there is an edge between nodes i and j. Let the weight on edge $e(k)$ be $W_{ij} \in \mathbb{R}$, then $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the graph's weighted adjacency matrix with $W_{ij} = 0$ if there is no edge between nodes i and j. We outline approaches for constructing such a graph directly from the data in Section 8.4 in the context of portfolio selection.

Given such a graph and long/short position vectors $\mathbf{q}_\ell, \mathbf{q}_s \in \mathbb{R}^{2n}$ where the long position of asset i is contained in index i of $\mathbf{q}_\ell$ and the short position of asset i is

contained in the index $i+n$ of $\mathbf{q}_s$, we introduce a hedging penalty function

$$\Omega_h = \sum_{i=1}^{n} \sum_{\substack{j=1+n \\ j\neq i+n}}^{2n} W_{ij} \left(q_\ell(i) + q_s(j)\right)^2 \tag{8.1}$$

where W_{ij} is a measure of the similarity between assets i and j and $q_\ell(i) \geq 0$ and $q_s(j) \leq 0$ are the value of the asset positions. We seek to minimize Ω_h, so when W_{ij} is large, we minimize Ω_h by making the value of the assets' opposing positions close, effectively encouraging hedging. When W_{ij} is small, the assets are not similar so the assets' position values do not need to be close and we do not hedge.

One key aspect of hedging is that we want to hold opposite positions in different but structurally related assets rather than the same asset since this is similar to holding only the difference between the positions. (8.1) is designed to do this by (i) taking into account the structural dependencies between assets in $\mathcal{G}$ and (ii) only considering edges between opposing positions in different assets. In other words, we do not consider the difference between $q_\ell(i)$ and $q_\ell(j)$ (same position type, different assets) or the difference between $q_\ell(i)$ and $q_s(i)$ (different position type, same assets).

One benefit of our quadratic hedging function (8.1) is that we are able to capture the interaction between different asset *positions* which provides more flexibility in responding to asset fluctuations in addition to considering just the correlation between assets. We can consider other hedging penalty functions, e.g., $\sum_{i=1}^{n} \sum_{\substack{j=1+n \\ j\neq i+n}}^{2n} W_{ij} \left(\frac{1}{q_\ell(i)-q_s(j)}\right)$, however such a function does not capture the position interactions. Additionally, such functions may not reflect a true hedging strategy since the function can be minimized by making one position large while leaving the other small. One limitation of (8.1) is that when there are two identical assets, this form will encourage both long positions and short positions to be similar across assets.

We can represent (8.1) in a more compact form by considering a graph over the $2n$ positions instead of the n assets. Let $\mathcal{G}_p = (V_p, E_p)$ be a graph over the positions where $V_p = \{v_\ell(1), v_s(1), \ldots, v_\ell(n), v_s(n)\}$ are the nodes, i.e., $v_\ell(i)$ is the node for the long position of asset i and $v_s(i)$ is the node for the short position of asset i. We can construct the set of edges E_p and their corresponding weights U_{ij} from $\mathcal{G}$ by considering assets i and j that are connected by edge $e(k)$ with weight W_{ij}. We add an edge to E_p between nodes $v_\ell(i)$ and $v_s(j)$ with weight W_{ij} and an edge between nodes $v_s(i)$ and

$v_\ell(j)$ with weight W_{ij}. In other words, we only connect the opposing position nodes between the different assets and the corresponding edges will have the same weight as the weight between the assets.

Under this construction we can similarly define $\mathcal{G}_p$'s weighted adjacency matrix $\mathbf{U} = \begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{W} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$, which is symmetric since $\mathbf{W}$ is symmetric. Let $\mathbf{D} \in \mathbb{R}^{2n \times 2n}$ be a diagonal matrix with $D(i,i) = \sum_j U_{ij}$ and 0 otherwise, then

$$\mathbf{L} = \mathbf{D} + \mathbf{U} \tag{8.2}$$

which is also symmetric and positive semi-definite by design.
Then, by construction

$$\mathbf{p}^\top \mathbf{L} \mathbf{p} = \sum_{i=1}^{n} \sum_{\substack{j=1+n \\ j \neq i+n}}^{2n} W_{ij} \left(q_\ell(i) + q_s(j) \right)^2 \ . \tag{8.3}$$

8.2.3 Online Resource Allocation Framework

We consider the online learning setting which proceeds in rounds where at each round t the algorithms selects a solution $\mathbf{p}_t \in \mathcal{P}$, Nature selects a convex loss function $\phi_t : \mathcal{P} \to \mathbb{R}$, the algorithm observes the entire loss function as feedback, and suffers a loss of $\phi_t(\mathbf{p}_t)$. Using $\Omega_h(\mathbf{p}) = \mathbf{p}^\top \mathbf{L} \mathbf{p}$ to induce hedged solutions, we would like to minimize the constrained cumulative loss

$$\sum_{t=1}^{T} \phi_t(\mathbf{p}_t) + \beta \Omega_h(\mathbf{p}_t) \ . \tag{8.4}$$

However, in the online setting, minimization of (8.4) is not feasible since we do not know the sequence of ϕ_t a priori. Alternatively, over T rounds we intend to get a sequence of $\mathbf{p}_t$ such that the following fixed regret is sublinear in T

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}^*} \sum_{t=1}^{T} f_t(\mathbf{p}^*) = o(T) \tag{8.5}$$

where $f_t(\mathbf{p}) = \phi_t(\mathbf{p}) + \beta \Omega_h(\mathbf{p})$. The regret is measured with respect to the best fixed minimizer in hindsight $\mathbf{p}^*$.

As in previous chapters, each day we consider solving a linearized version by taking a first-order Taylor expansion of ϕ_t at $\mathbf{p}_t$ along with a proximal term, so we end up solving

$$\mathbf{p}_{t+1} = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{P}} \langle \mathbf{p}, \nabla\phi_t(\mathbf{p}_t)\rangle + \beta\Omega_h(\mathbf{p}) + d(\mathbf{p}, \mathbf{p}_t) \ , \tag{8.6}$$

where $d(\mathbf{p}, \mathbf{p}_t)=\frac{1}{2\eta_t}\|\mathbf{p} - \mathbf{p}_t\|_2^2$ and parameters $\eta_t, \beta \geq 0$.

8.3 Online Portfolio Selection

We follow a similar online portfolio selection setting as discussed in Chapter 4 with some key changes to allow leverage and long/short positions. Recall for a stock market consisting of n stocks we consider price relatives $x_t(i) > 0$ for $i = 1, \ldots, n$ at day t which is the closing price over the opening price of stock i. Let $\hat{\mathbf{x}}_t = [x_t(1), \ldots, x_t(n)]^\top$ denote the vector of price relatives for day t, let $\mathbf{x}_t = [\hat{\mathbf{x}}_t; \hat{\mathbf{x}}_t]$ be the $2n \times 1$ length double stacked vector of price relatives, and let $\mathbf{x}_{1:t}$ denote the collection of such price relative vectors up to and including day t. A portfolio on day t is $\mathbf{p}_t = [p_t(1), \ldots, p_t(2n)]^\top \in \mathcal{P}$ where the first $|\ell|$ elements are long-only positions, i.e., $p_t(i) \geq 0$ and the last $|s|$ elements are short-only positions, i.e., $p_t(i) \leq 0$ which prescribes investing $p_t(i)$ fraction of the total wealth, including leverage, in stock s_i.

8.3.1 Long-Only Portfolios

For long-only portfolios $\mathbf{q}_\ell \geq 0$ without leverage, the multiplicative gain in wealth at the end of day t is $\mathbf{q}_\ell^\top \mathbf{x}_t$. When we allow borrowing cash from the bank as leverage we have

$$\underbrace{\mathbf{q}_\ell^\top \mathbf{x}_t}_{\substack{\text{market change}\\ \text{in wealth}}} + \underbrace{(1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)}_{\substack{\text{cash borrowed}\\ \text{or not invested}}} \ . \tag{8.7}$$

The last term accounts for the percentage of our current wealth we borrowed from the bank or wealth not invested. If we did not borrow any cash then $0 \leq (1 - \mathbf{q}_\ell^\top \mathbf{1}) \leq 1$ and any cash not invested is considered as being held in a savings bank account which earns interest at of rate of r. If we did borrow cash, then this term is negative and is the amount we have to pay back to the bank plus interest r.

When we allow borrowing cash, we have to be careful about owing more money than we have left at the end of any day in order to avoid financial ruin. For instance, if we invest long with leverage and the market crashes, we may have no money left to pay back the bank loan. In order to guarantee no-ruin, we make an assumption on the price relatives similar to [65] such that $0 < 1 - B_\ell < \mathbf{x}_t$ where B_ℓ is a parameter that can be set based on historical stock data. Then the maximum amount we can invest is $\frac{1+r}{B_\ell+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Definition 1 For a long-only portfolio $\mathbf{q}_\ell$ such that $\mathbf{q}_\ell \geq 0$, $\|\mathbf{q}_\ell\|_1 \leq \frac{1+r}{B_\ell+r}$, and with bounded price relatives $0 < 1-B_\ell < \mathbf{x}_t$, the multiplicative gain $\mathbf{q}_\ell^\top \mathbf{x}_t + (1-\mathbf{q}_\ell^\top \mathbf{1})(1+r) \geq 0$.

Proof: For ease of exposition, let us consider only a single investment so $\mathbf{q}_\ell \in \mathbb{R}_+$. Then in the worst case we have

$$\mathbf{q}_\ell^\top \mathbf{x}_t + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r) \geq \frac{1+r}{B_\ell + r}(1 - B_\ell) + \left(1 - \frac{1+r}{B_\ell + r}\right)(1+r) = 0 \ .$$

8.3.2 Short-Only Portfolios

For short-only portfolios $\mathbf{q}_s \leq 0$ without cash leverage, the multiplicative gain in wealth at the end of day t consists of the difference between the price value of shares borrowed $\mathbf{q}_s^\top \mathbf{1}$ and the price the shares are now worth $\mathbf{q}_s^\top \mathbf{x}_t$ minus the interest owed on borrowing the shares plus the collateral c and interest on collateral r. In this chapter, we assume the interest rate for borrowing cash/shares is the same as the interest rate earned for keeping cash and collateral in a margin account. Additionally, the total value of shares borrowed is equal to the collateral put down, i.e., $\|\mathbf{q}_s\|_1 = c$. Therefore, the multiplicative gain in wealth is $\mathbf{q}_s^\top(\mathbf{x}_t - 1) + \mathbf{q}_s^\top \mathbf{1} r - \mathbf{q}_s^\top \mathbf{1}(1+r)$. When we allow borrowing cash from the bank as leverage we have

$$\underbrace{\mathbf{q}_s^\top(\mathbf{x}_t - 1)}_{\substack{\text{market change} \\ \text{in wealth}}} + \underbrace{\mathbf{q}_s^\top \mathbf{1} r}_{\substack{\text{interest owed on} \\ \text{borrowed shares}}} + \underbrace{(1+\mathbf{q}_s^\top \mathbf{1})(1+r)}_{\substack{\text{cash borrowed} \\ \text{or not invested}}} - \underbrace{\mathbf{q}_s^\top \mathbf{1}(1+r)}_{\substack{\text{collateral and} \\ \text{interest earned}}} \ .$$

Re-arranging the terms we get

$$\begin{aligned}&\mathbf{q}_s^\top(\mathbf{x}_t-1)+\mathbf{q}_s^\top\mathbf{1}r+(1+\mathbf{q}_s^\top\mathbf{1})(1+r)-\mathbf{q}_s^\top\mathbf{1}(1+r)\\&=\mathbf{q}_s^\top(\mathbf{x}_t-1+r)+(1+r)\ .\end{aligned}\tag{8.8}$$

Similar to long-only portfolios, in order to guarantee no-ruin we assume $\mathbf{x}_t < 1+B_s < \infty$. Then the maximum amount we can invest is $\frac{1+r}{B_s+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Definition 2 For a short-only portfolio $\mathbf{q}_s$ such that $\mathbf{q}_s \leq 0$, $\|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_s+r}$, and with bounded price relatives $\mathbf{x}_t < 1+B_s < \infty$, the multiplicative gain $\mathbf{q}_s^T(\mathbf{x}_t-1+r)+(1+r) \geq 0$.

Proof: For ease of exposition, let us consider only a single investment so $\mathbf{q}_s \in \mathbb{R}_-$. Then in the worst case we have

$$\mathbf{q}_s^T(\mathbf{x}_t-1+r)+(1+r) \geq -\frac{1+r}{B_s+r}((1+B_s)-1+r)+(1+r)=0\ .$$

8.3.3 Long and Short Portfolios

For portfolios that allow both long and short positions with leverage, the multiplicative gain in wealth at the end of the day t will combine (8.7) and (8.8) to get

$$\mathbf{q}_\ell^\top\mathbf{x}_t+(1-\mathbf{q}_\ell^\top\mathbf{1})(1+r)+\mathbf{q}_s^\top(\mathbf{x}_t-1+r)+(1+r)\ .$$

However, we have counted the amount of cash borrowed or not invested plus interest twice since both were included individually in (8.7) and (8.8) so we subtract $(1+r)$ to get

$$\mathbf{q}_\ell^\top\mathbf{x}_t+\mathbf{q}_s^\top(\mathbf{x}_t-1+r)+(1-\mathbf{q}_\ell^\top\mathbf{1})(1+r)\ .\tag{8.9}$$

Unlike in [65], we do not assume $B_\ell = B_s$ since bear and bull markets tend to not be symmetrical. Therefore, we define a hyperplane that guarantees no-ruin for a portfolio with a combination of long and short positions as $\|\mathbf{q}_\ell\|_1 + \frac{B_s+r}{B_\ell+r}\|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_\ell+r}$. Since $\mathbf{q}_s \leq 0$, we can define a vector $\mathbf{a}$ such that the first $|\ell|$ elements are equal to 1 and the

last $|s|$ elements are equal to $-\frac{B_s+r}{B_\ell+r}$, so the constraint on maximum investment as a combination of long and short positions is $\mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Definition 3 For a long and short portfolio $\mathbf{p} = \mathbf{q}_\ell + \mathbf{q}_s$ such that $\mathbf{q}_\ell \geq 0$, $\mathbf{q}_s \leq 0$, $\mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}$, and with bounded price relatives $0 < 1 - B_\ell < \mathbf{x}_t < 1 + B_s < \infty$, the multiplicative gain $\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \geq 0$.

Proof: Setting $u = \|\mathbf{q}_\ell\|_1$ implies $\|\mathbf{q}_s\| \leq \left(\frac{1+r}{B_s+r} - \frac{B_\ell+r}{B_s+r}u\right)$. Then in the worst case

$$\begin{aligned}
&\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \\
&\geq u(1 - B_\ell) - \left(\frac{1+r}{B_s+r} - \frac{B_\ell+r}{B_s+r}u\right)((1 + B_s) - 1 + r) + (1 - u)(1 + r) \\
&= 0 \ .
\end{aligned}$$

Therefore, the multiplicative gain in wealth at the end of day t for a leveraged portfolio with long and short positions is (8.9) which, in the worst case, is non-negative. This ensures we have enough money left over after the market moves to pay back all loans.

Given this, and a sequence of price relatives $\mathbf{x}_{1:t-1} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}\}$ up to day $(t-1)$, the sequential portfolio selection problem in day t is to determine a portfolio $\mathbf{p}_t$ based on past performance of the stocks. At the end of day t, $\mathbf{x}_t$ is revealed and the actual performance of $\mathbf{p}_t$ gets determined by (8.9). Over T periods, for a sequence of portfolios $\mathbf{p}_{1:T} = \{\mathbf{p}_1, \ldots, \mathbf{p}_T\}$, the multiplicative and logarithmic gain in wealth are

$$\begin{aligned}
S_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) &= \prod_{t=1}^{T} \left(\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)\right) \\
LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) &= \sum_{t=1}^{T} \log \left(\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)\right) .
\end{aligned}$$

8.4 Algorithm

Online portfolio selection can now be viewed as a special case of our online resource allocation with structured hedging setting with

$$\phi_t(\mathbf{p}_t) = -\log \left(\alpha_1 \mathbf{q}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)\right)$$

Algorithm 7 SHERAL Algorithm

1: Input $\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s, r, \mathbf{D}_\ell, \mathbf{D}_s, \mathcal{G}$.

2: Compute weighted adjacency matrix $\mathbf{L}$ via (8.2).

3: Compute portfolio for day $t+1$:

$$\mathbf{p}_{t+1} = \prod_{\mathcal{P}} \left(\eta \nabla \phi_t(\mathbf{p}_t) + \mathbf{p}_t\right) \left(\lambda(\mathbf{L} + \mathbf{L}^\top) + \mathbf{I}\right)^{-1}$$

where $\nabla \phi_t(\mathbf{p}_t)$ is as defined in (8.12) and $\prod_{\mathbf{p}\in\mathcal{P}}$ is a projection to the convex set $\mathcal{P} = \left\{\mathbf{p} \,|\, \mathbf{D}_\ell \mathbf{p} \geq 0, \mathbf{D}_s \mathbf{p} \leq 0, \mathbf{a}^T \mathbf{p} \leq \frac{1+r}{B_\ell + r}\right\}$ via alternating projections.

Algorithm 8 Hedging for Online Portfolio Selection

1: Input $\eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s$, Interest rate r, Transaction cost γ, Days lag δ.

2: Set $S_0^\gamma = \$1$ and

3: $p_{0:1}(i) = \frac{(1+r)/(B_\ell+r)}{1+(B_s+r)/(B_\ell+r)}$ $\forall i$ (uniform over positions).

4: For $t = 1, \ldots, T$

5: Receive the vector of price relatives: $\mathbf{x}_t$.

6: Compute multiplicative gain in wealth via (8.9) as Ψ_t.

7: Compute wealth: $S_t^\gamma = S_{t-1}^\gamma \left(\Psi_t - \gamma \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1\right)$.

8: If $t \leq \delta$

9: $\mathbf{p}_{t+1} = \mathbf{p}_t$ (uniform over positions).

10: Else

11: Compute graph: $\mathcal{G}$.

12: $\mathbf{p}_{t+1} = \text{SHERAL}(\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s, r, \mathbf{D}_\ell, \mathbf{D}_s, \mathcal{G})$.

13: end for

where $\alpha_1, \alpha_2 \geq 0$ are parameters that control the importance of long and short positions respectively. Note, if both long and short positions are valued equally, $\alpha_2 > \alpha_1$ since the scale of the position returns differ by a factor (Refer to 8.6.3 **(c)** for an example). Letting $\eta_t = \eta$ and multiplying each term in (8.6) by η so that $\lambda = \eta\beta$, the online portfolio selection with structured hedging problem is

$$\begin{aligned} \min_{\substack{\mathbf{q}_\ell \geq 0 \\ \mathbf{q}_s \leq 0 \\ \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell + r}}} \quad & \eta \langle \mathbf{p}, \nabla \phi_t(\mathbf{p}_t) \rangle + \lambda \mathbf{p}^\top \mathbf{L} \mathbf{p} + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_t\|_2^2 \,. \end{aligned} \tag{8.10}$$

This is a strongly convex optimization problem with the linear constraint set $\mathcal{P} =$

$\left\{\mathbf{p} \,|\, \mathbf{q}_\ell \geq 0, \mathbf{q}_s \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell + r}\right\}$ where $\mathbf{q}_\ell = \mathbf{D}_\ell \mathbf{p}$ and $\mathbf{q}_s = \mathbf{D}_s \mathbf{p}$ (Section 8.2.2).

We propose an efficient projected gradient descent algorithm for solving (8.10). Since the objective in (8.10) is strongly convex we can find the minimum by taking the gradient, setting it to zero, and solving for $\mathbf{p}$ to get

$$\mathbf{p}_{t+1} = \prod_{\mathcal{P}} \left(\eta \nabla \phi_t(\mathbf{p}_t) + \mathbf{p}_t\right) \left(\lambda(\mathbf{L} + \mathbf{L}^\top) + \mathbf{I}\right)^{-1} \tag{8.11}$$

where

$$\nabla \phi_t(\mathbf{p}_t) = \frac{\alpha_1 \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) - \mathbf{D}_\ell^\top \mathbf{1}(1+r)}{\alpha_1 \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{p}_t^\top \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{1})(1+r)} \tag{8.12}$$

and $\prod_{\mathcal{P}}$ is a projection onto the convex constraint set $\mathcal{P}$.

Algorithm 7 shows the complete details for computing the hedged portfolio. Algorithm 8 is our hedged online portfolio selection with leverage algorithm which includes computing the transaction-cost adjusted wealth using a fixed percentage transaction cost γ.

8.5 Regret Bound

We sequentially invest with the hedged portfolios $\mathbf{p}_1, \cdots, \mathbf{p}_T$ obtained from Algorithm 8 and on day t suffer a loss of $f_t(\mathbf{p}_t) = \eta \phi_t + \lambda \mathbf{p}_t^\top \mathbf{L} \mathbf{p}_t$. Our goal is to minimize the *regret* with respect to the best fixed portfolio $\mathbf{p}^*$ in hindsight. We establish the standard regret bound in portfolio selection literature [4, 34, 69] and omit the proof since it follows from existing results.

Theorem 8 *Let $\mathbf{p}^* \in \mathcal{P}$ be the fixed portfolio obtained from $\min_{\mathbf{p}} \sum_{t=1}^T \phi_t(\mathbf{p})$. For $\eta = \frac{1}{\sqrt{T}}$, $\lambda = \frac{1}{t}$, and $\|\nabla \phi_t(\mathbf{p}_t)\|_2^2 \leq G$, the regret can be bounded as,*

$$\sum_{t=1}^T \phi_t(\mathbf{p}_t) + \mathbf{p}_t^\top \mathbf{L} \mathbf{p}_t - \sum_{t=1}^T \phi_t(\mathbf{p}^*) \leq O(\sqrt{T}), \tag{8.13}$$

where ϕ_t is a strongly convex function and the sequence $\mathbf{p}_t$ and the fixed optimal portfolio $\mathbf{p}^*$ all lie in the constraint set $\mathcal{P} = \left\{\mathbf{p} \,|\, \mathbf{D}_\ell \mathbf{p} \geq 0, \mathbf{D}_s \mathbf{p} \leq 0, \mathbf{a}^T \mathbf{p} \leq \frac{1+r}{B_\ell + r}\right\}$.

8.6 Experiments and Results

The experiments were conducted on 5 datasets with data taken from: Dow Jones Industrial Average (DJIA), New York Stock Exchange (NYSE), Standard & Poor's 500 (S&P 500), and the Toronto Stock Exchange (TSX) (refer to Section 4.3 for details). Note, we use a subset of the S&P500 dataset which consists of the same 263 stocks but over a shortened time period of 505 trading days over a period of 2 years from 2007 to 2009 and we denote this dataset[1] S&P500^b. The reason we consider a subset is to focus specifically on the financial and housing crash where 253 of the 263 stocks (96%) lost value to illustrate the effectiveness of short positions and structured hedging.

8.6.1 Methodology and Parameter Setting

In all our experiments we start with \$1 as our initial investment and an initial portfolio with maximum leverage uniformly distributed over all the positions. We use Algorithm 8 to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth each day.

Since the five datasets are very different in nature, we experimented with various parameter values for all algorithms using a grid search in the following ranges: $\delta \in \{5, 10, \ldots, 50\}$, and each of η, α_1, α_2, and λ following a log-scale in $[10^{-6}, 10^3]$ to observe their affect on our portfolio and found stable behavior in these ranges. Moreover, we chose a reasonable range of transaction costs $\gamma \in [0\%, 2\%]$ to observed their affect on the transaction-cost adjusted wealth.

Typical yearly margin interest rates are between 5% and 8%. For all experiments we set the daily interest rate $r = 0.000245$ which is equivalent to a yearly interest rate of 6.3%. Rates around 6.3% have been used before in the literature [34, 65, 69]. For each dataset, we computed the parameters B_ℓ and B_s to the nearest hundredth decimal place in hindsight (Table 8.1). In practice, one must use historical data to compute such values, thus, no-ruin guarantees can only be made probabilistically. Alternatively, one could develop online generalizations of Bayesian Optimization or suitable variants of parameter free online learning to avoid parameter setting in hindsight.

In all our experiments, we construct the graph $\mathcal{G}$ by fully connecting each stock so

[1] http://www-users.cs.umn.edu/∼njohnson/port_sel.html

every node has degree $n-1$. We compute their similarity by calculating the linear correlation coefficient w_{ij} between each pair of stocks i and j using the previous δ days of price relatives to get the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. The positional graph $\mathcal{G}_p$ was then constructed as in (8.2).

8.6.2 Cumulative Wealth

To evaluate the practical application of SHERAL, we analyze the performance and compare several benchmark algorithms with parameters tuned for maximum cumulative wealth (without transaction costs).

Leveraged Long and Short Effect on EG

First, we take the well-known algorithm EG [69] and observe its performance on the 5 datasets. To observe the affect that long and short positions with leverage have, we further experiment with variants of EG to allow: (1) long-only positions with leverage (LO), (2) short-only positions with leverage (SO), and (3) long and short positions with leverage (LS).

EG Variants Formulation: EG allows long-only positions $\mathbf{p} \geq 0$ and uses the relative entropy function as the proximal term, i.e., $d(\mathbf{p}, \mathbf{p}_t) = \sum_{i=1}^{n} p(i) \log\left(\frac{p(i)}{p_t(i)}\right)$ where $\mathbf{p} \in \Delta_n = \{p(i) \geq 0 \, \forall i, \sum_i p(i) = 1\}$. Since we need to allow $p(i) \leq 0$ for short positions, we instead use $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2$. Additionally, we utilize the loss function defined in (8.9). Essentially, the variant EG problem is

$$\min_{\mathbf{p} \in \mathcal{P}} \eta \langle \mathbf{p}, \nabla \phi_t(\mathbf{p}_t) \rangle + \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2 \tag{8.14}$$

where $\mathcal{P} = \{\mathbf{p} \,|\, \mathbf{q}_\ell \geq 0, \|\mathbf{q}_\ell\|_1 \leq \frac{1+r}{B_\ell + r}\}$ for LO, $\mathcal{P} = \{\mathbf{p} \,|\, \mathbf{q}_s \leq 0, \|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_s + r}\}$ for SO, and $\mathcal{P} = \{\mathbf{p} \,|\, \mathbf{q}_\ell \geq 0, \mathbf{q}_s \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell + r}\}$ for LS. We can see that this is a special case

	DJIA	NYSE	SP500^a	SP500^b	TSX
B_ℓ	0.60	0.26	0.31	0.61	0.64
B_s	0.21	0.36	0.25	0.67	0.94

Table 8.1: Table of B_ℓ and B_s values for each dataset.

	DJIA	NYSE	SP500^a	SP500^b	TSX
EG	0.81	26.70	1.64	0.68	1.59
EG* (LO)	1.55	6.9×10^{14}	**20.90**	2.21	$\mathbf{1.0\times10^{3}}$
EG* (SO)	0.63	0.04	0.34	1.10	1.07
EG* (LS)	2.00	6.6×10^{14}	20.65	2.26	1.62
SHERAL (λ>0)	**2.47**	$\mathbf{1.8\times10^{15}}$	19.89	**7.84**	8.74

Table 8.2: Cumulative wealth for EG, leveraged long-only (LO), short-only (SO), and long/short (LS) variants of EG*, and SHERAL with $\lambda > 0$.

of (8.10) where $\lambda = 0$. Therefore, SHERAL is able to perform at least as good as the EG* variants.

Results: The results are presented in Table 8.2. From this table, we can see that the original EG is outperformed by at least one of the EG* variants in each dataset. Interestingly, when only adding leverage to EG, i.e. EG* (LO), the performance improves substantially for the NYSE, SP500^a, and TSX datasets. It seems that EG* (LO) is able to select the correct stocks and that increasing the investment power dramatically enhances the multiplicative gain in wealth.

Additionally, we see that EG* (SO) is able to earn more wealth for the SP500^b dataset than EG. This is because almost all stocks in this dataset lose value and EG is not able to hold short positions thereby limiting it to invest in stocks that are performing poorly. However, we also see that EG* (LO) earns more wealth than EG* (SO) even though it is limited by only holding long positions. It seems that for the few stocks that have increases in value, the leverage is enough to allow it to earn more wealth. Finally, we see that the EG* (LS) is able to outperform EG on each datasets even though they are significantly different in nature. It is also able to compete with EG* (LO) in 4 of the 5 datasets. From these results, we can see that adding leverage to EG significantly increases cumulative wealth. Moreover, it seems that short positions do not have a strong impact on cumulative wealth but they do allow for more flexibility and reasonable performance on datasets with both bull and bear markets.

We can observe the affect that structured hedging has on wealth by comparing the cumulative wealth of SHERAL to that of EG and EG* variants. We can especially see its affect by looking at EG* (LS) since it is equivalent to setting $\lambda = 0$ in (8.10). We

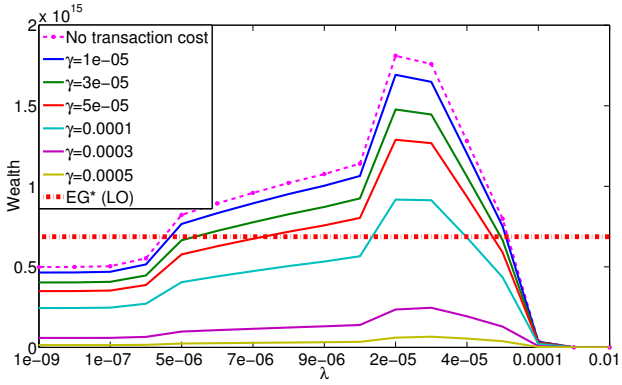


Figure 8.1: Transaction cost-adjusted wealth for the NYSE dataset with varying values of λ. SHERAL returns more wealth than the best competing algorithm even with transaction costs.

can see that SHERAL is able to earn more wealth than EG and all EG* variants on 3 of the 5 datasets. This implies that there is some volatility in these datasets and the structured hedging helps alleviate this volatility. The structured hedging has a positive affect on most datasets and, as we will see later, it is able to reduce the risk and earn similar amounts of wealth. When $\lambda = 0$, SHERAL can earn as much as EG and all EG* variants so for datasets where hedging is not beneficial, we can still earn the same amount of wealth.

In Figure 8.1, we can see how the transaction-cost adjusted cumulative wealth changes as we vary the structured hedging weight λ in comparison with the best performing variant EG* (LO) for the NYSE dataset. SHERAL is able to earn more wealth even with reasonable transaction costs than the best EG variant (which does not include transaction costs).

Comparison with Benchmark Algorithms

In addition to a comparison between EG and our EG* variants, we compared the performance between several benchmark algorithms: a buy-and-hold strategy (U-BAH), a uniform constant rebalanced portfolio (U-CRP), Universal Portfolios (UP) [34], Online Lazy Updates (OLU) from Chapter 5, and the best single stock in hindsight. These algorithms were designed to only invest in long positions and without leverage.

	DJIA	NYSE	SP500^a	SP500^b	TSX
U-BAH	0.76	14.49	1.34	0.63	1.61
U-CRP	0.81	26.78	1.64	0.69	1.59
UP	0.80	26.99	1.62	0.69	1.59
OLU	0.84	50.80	2.45	3.02	2.24
Best Stock (LO)	1.18	54.14	3.77	1.74	6.27
U-BAH* (LO)	0.55	38.44	0.44	0.38	1.71
U-BAH* (SO)	0.43	3.68×10^{-6}	0.01	1.21	0.54
U-BAH* (LS)	0.97	0.43	0.78	1.03	1.12
U-CRP* (LO)	0.61	695.59	1.04	0.43	1.68
U-CRP* (SO)	0.28	1.04×10^{-6}	1.00×10^{-2}	0.97	0.54
U-CRP* (LS)	0.83	0.05	0.45	0.93	0.92
Best Stock* (LO)	1.09 (P&G)	65.71 (PM)	0.56 (WMT)	1.15 (SWN)	7.84 (GTA)
Best Stock* (SO)	1.13 (MCD)	1.45×10^{-5} (DD)	0.02 (KO)	4.71 (GCI)	2.11 (IFP)
SHERAL ($\lambda > 0$)	**2.47**	$\mathbf{1.81\times10^{15}}$	**19.89**	**7.84**	**8.74**

Table 8.3: Cumulative wealth (without transaction costs) of SHERAL, benchmark algorithms, and several variants for each of the five datasets.

Further, we again compared against several variants of these algorithms: a long-only, short-only, and long/short with leverage variant of both U-BAH and U-CRP, and a long-only and short-only with leverage variant of the best single stock. The results of these algorithms and variants on the 5 datasets are presented in Table 8.3.

We can see that out of those algorithms that only invest in long positions and without leverage, the best single stock tends to outperform the best with OLU being competitive in most cases. We can see how the market performs by looking at the U-BAH algorithm. If U-BAH returns < 1 then the market was down for that dataset. For such datasets (DJIA, SP500^b), we see that both U-CRP and UP perform about the same and lose money. However, the best stock earns money for these datasets. We can see that there is at least one stock that performs well even if the majority of the stocks do not. OLU is able to identify this stock, or the few similar stocks, for SP500^b and earn money but is not able to do the same for DJIA. Comparing the U-BAH and U-CRP variants on the SP500^b dataset, we see that only U-BAH* (SO) and (LS) are able to earn money. Since for this dataset, the vast majority of the stocks decrease in value, these variants are able to hold short positions and take advantage of this. U-CRP* (SO) and (LS) lose

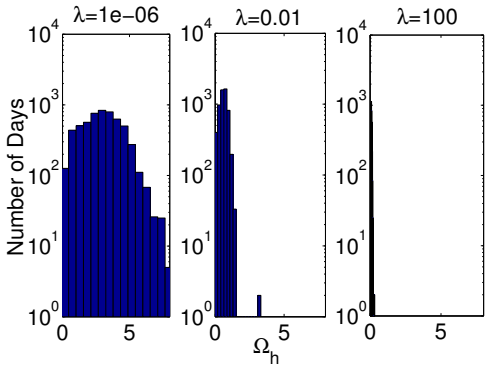


Figure 8.2: Total value of the hedging penalty Ω_h with varying λ for the NYSE dataset. As λ increases, the value Ω_h decreases.

money but still perform better than the (LO) variant for this dataset.

For the NYSE dataset, we can see that U-CRP* (LO) earns significantly more wealth than all the other algorithms (except SHERAL). Since all the stocks in the NYSE dataset increase in value, holding long positions is beneficial and adding leveraging provides even more earning power. However, comparing the leveraged (LO) best stock with the non-leveraged (LO) best stock, we can see that for each dataset, the leverage does not appear to help that much and even in some cases hurts the cumulative wealth. This is because leverage magnifies not only gains but also losses. We can see the benefits of leverage but also the drawbacks. If we have a portfolio that is allowed to switch between stocks with leverage, there can be huge gains. However, if we are stuck investing in a single stock with leverage, there are few gains and even some losses.

Looking at the last row of Table 8.3, we observe the results for our SHERAL algorithm. We can see that SHERAL is able to outperform all other algorithms for each of the datasets even though they are very different in nature. For example, SHERAL earns \$7.84 on the SP500^b dataset where 96% of the stocks decrease in value and is also able to earn $\mathbf{\$1.81 \times 10^{15}}$ on the NYSE dataset where all stocks increase in value. This shows the flexibility and power of being able to invest in both long and short positions with leverage.

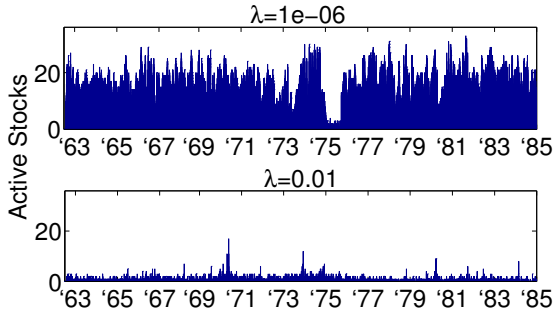


Figure 8.3: Active stocks with varying λ. As λ increases the number of active stocks decreases which indicates either more hedging or less total investing.

8.6.3 Effect of Hedging Function Ω_h and λ

The hedging penalty function Ω_h encourages structural hedging between assets. The amount of hedging is further controlled by the value of λ which has an affect on (a) the value of Ω_h each day, (b) the number of active stocks, and (c) the number of active positions.

(a) Value of Ω_h With the hedging penalty function Ω_h, we are encouraging different levels of asset hedging depending on the value of λ. From Figure 8.2, we can see the affect λ has on $\Omega_h = \mathbf{p}^\top \mathbf{L}\mathbf{p}$ with $\eta = 0.1, \alpha_1 = 0.1$, and $\alpha_2 = 0.1$ for the NYSE dataset with $\delta = 20$ days. With a low λ value of 10^{-6}, there are many days with large values of Ω_h. As we increase λ to 100, we see that the value of Ω_h becomes very small with most days having a value of 0.

(b) Number of Active Stocks An active stock is a stock which has a significant percent of the wealth, e.g, 1%, 10%, etc., invested in it between both long and short positions. For instance, if there is 1% of the wealth invested in the long position and 1% invested in the short position, the total *effective wealth* invested is 0% since the amounts are equal and cancel out. An active stock has a significant amount of total effective wealth invested. It is a measure of the number of non-hedged assets where a higher number indicates less hedging and a lower number indicates more hedging or less total investing, i.e., more wealth held in the bank.

From Figure 8.3, we can see that with $\eta = 0.1$, $\alpha_1 = 1$, $\alpha_2 = 10$, and $\lambda = 10^{-6}$, the number of active stocks for the NYSE dataset is high, with around 25 out of 36 stocks

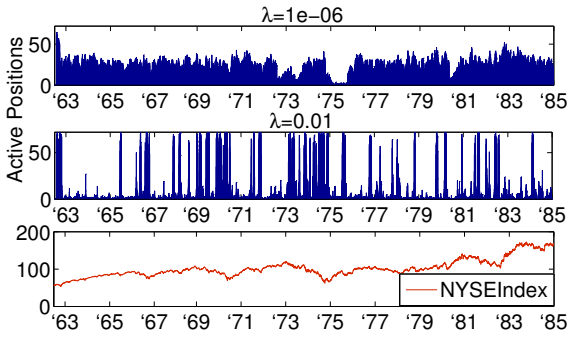


Figure 8.4: Active positions with varying λ. When the NYSE Index decreases, hedging increases with $\lambda = 0.01$.

active (about 70%) on average. With the other parameters fixed and only increasing λ to 0.01, the number of active stocks decreases to around 4 (about 11%). This could be due to two reasons: (1) the amount invested in both long and short positions increases and the total effective wealth invested decreases, or (2) the total amount invested decreases with the rest kept in the bank.

(c) Number of Active Positions We define an active position in a similar way to that of an active stock, however we consider the positions independent of one another. If both the long and short positions have the same and significant amount invested in them, then both are considered active. From Figure 8.4, we can see the affect λ has on the number of active positions for the same parameters values as that of Figure 8.3. With a low λ value of 10^{-6}, we can see that the number of active positions is reasonably high, with around the same amount as the number of active stocks in Figure 8.3 (note the scale differences). This shows that when λ is small for these set of parameters, SHERAL is not hedging much. When λ increases to 0.01, the number of active positions increases some days and decreases other days.

Comparing this to the NYSE index value in the bottom plot, we can see that the days where SHERAL is hedging more corresponds to days that the index decreases, e.g. 1969-1970, 1973-1975, and 1981-1983. This is explained by the fact that with this set of parameters, we are emphasizing long positions more even though $\alpha_2 > \alpha_1$ because of the difference in scale. For example, the mean price relative for the NYSE dataset is 1.0006, and for a long position the gain is $\mathbf{q}_\ell^\top \mathbf{x}_t$ which on average is of the order $1 \times \mathbf{q}_\ell$

whereas the gain for short positions is $\mathbf{q}_s^\top(\mathbf{x}_t - 1 + r)$ which on average is of the order $(10^{-4}+r)\times\mathbf{q}_s$. With $r = 0.000245$, to weight the positions equally we would have to set α_2 to be in the range $[10^3, 10^4]$. However, for these results $\alpha_2 = 10$ so the long positions are emphasized more. Since there is more hedging on days the market decreases, this indicates that SHERAL is trying to take advantage of increasing stocks on some days whereas other days, when the market is crashing, it is not quite sure which stocks will perform well so it is hedging more between them.

8.6.4 Risk Comparison

From Section 8.6.2, we saw that SHERAL is able to return more wealth than all of the state-of-the-art algorithms on all of the datasets, and more wealth than EG and our EG* variants in three out of the five datasets. However, as Markowitz postulated, we should seek low risk in addition to high returns. As such, we compare the risk exposure for each of the competing algorithms with optimal parameters with respect to wealth using three common measures of risk: (a) covariance, (b) Sharpe ratio, and (c) Sortino ratio.

(a) Covariance We compute the covariance $\mathbf{\Sigma}_t$ using the previous δ days of price relatives. We measure the risk of a portfolio $\mathbf{p}_t$ *w.r.t.* a uniform constant rebalanced portfolio $\mathbf{u}$ as α_{var}=$\mathbf{p}_t^\top\mathbf{\Sigma}_t\mathbf{p}_t/\mathbf{u}^\top\mathbf{\Sigma}_t\mathbf{u}$. High α_{var} implies high risk and low α_{var} implies low risk.

(b) Sharpe ratio The Sharpe ratio [113] measures how much the return (percent gain or loss on investment) of a portfolio compensates for the level of risk taken. It computes what can be considered as a risk-adjusted return for a given portfolio and benchmark return. It does this by measuring both the downwards and upwards volatility. A higher Sharpe ratio implies better compensation for the risk exposure. We compute the Sharpe ratio of a portfolio as α_{Sharpe}=$(R-R_b)/\sqrt{var(R-R_b)}$ where R is the return for the portfolio and R_b is the benchmark return which is typically a large index such as the S&P500.

(c) Sortino ratio The Sortino ratio [114] is similar to the Sharpe ratio, however it only measures the downwards volatility. Typically, upwards volatility is encouraged as we would gladly accept the price of a stock we have invested long in to go up. However, the Sharpe ratio penalizes this type of volatility where the Sortino ratio does

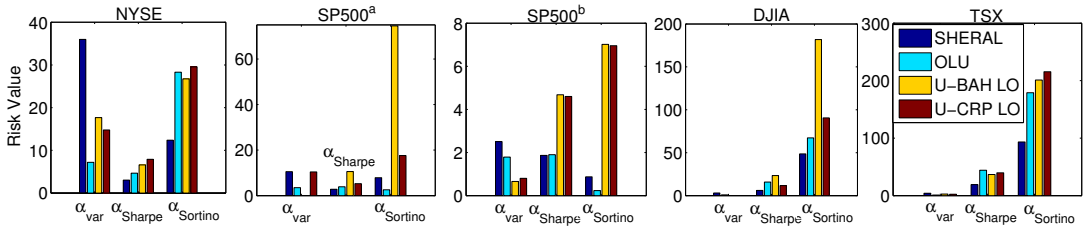


Figure 8.5: Average risk for each algorithm and dataset with optimal parameters in terms of wealth returned. SHERAL computes portfolios with less risk than U-BAH (LO) and U-CRP (LO) for almost all risk measures and datasets and is competitive with the non-leveraged algorithm OLU.

not. We compute the Sortino ratio as $\alpha_{Sortino} = (R - R_b)/DR$ where DR is the standard deviation of negative returns (losses).

To be consistent in the plots and have each bar represent the level of risk exposure, we have plotted the negative Sharpe and Sortino ratios since a low ratio implies a high risk relative to the return. Therefore, for each of the bar plots in Figure 8.5, a higher bar height implies higher risk. Additionally, we compare algorithms that are not special cases of SHERAL, i.e., EG or EG variants. We also only compare the best performing algorithms and variants with parameters tuned for optimal wealth for each algorithm.

From Figure 8.5, we can see that SHERAL is competitive with OLU in terms of risk, and in many cases having less risk. SHERAL consistently has less risk than U-BAH (LO) and U-CRP (LO). These results are encouraging because in addition to earning more wealth, SHERAL is able to reduce risk in spite of using leverage which inherently increases risk. Additionally, for algorithms which do use leverage, U-BAH (LO) and U-CRP (LO), SHERAL is exposed to much less risk than these algorithms and earns more wealth.

8.6.5 Risk and λ

Even though our hedged resource allocation with leverage framework (8.6) does not explicitly take risk into account, we can control it by setting the value of λ. We observe how the value of λ affects the amount of risk our portfolios are exposed to using variance as the measure of risk. We do not consider the Sharpe or Sortino ratios here since

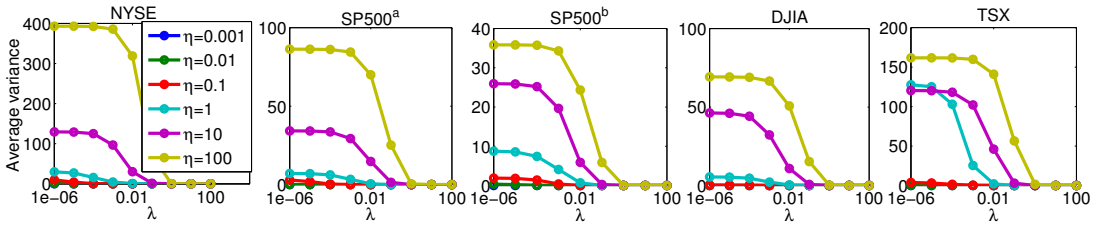


Figure 8.6: Average α_{var} risk for each dataset with varying η and λ. With a higher η value, the average variance risk is higher, however as we increase λ the risk decreases for each η value across all datasets.

with varying parameters, the return in wealth changes drastically and the risk-adjusted returns from the Sharpe and Sortino ratios cannot effectively illustrate how only the risk changes.

From Figure 8.6, we can see that across each dataset the behavior is consistent. As we increase η, the average variance is higher across all values of λ. As we increase λ, the average variance decreases to zero for all η values. This illustrates how as we increase λ, we are encouraging more hedged portfolios and reducing risk. Thus, we can effectively control the level of risk by setting the value of λ.

Chapter 9

Structured Stochastic Linear Bandits

In this chapter, we switch from considering problems which admit full information feedback to problems with bandit information feedback. Recall, in such problems, the algorithm can only observe the loss of the action selected $\ell_t(x_t)$ for $x_t \in \mathcal{X} \subset \mathbb{R}^p$ rather than the loss of all actions the algorithm could have selected $\ell_t(x)\ \forall x \in \mathcal{X}$. Since the algorithm receives less feedback, learning is more difficult and it has been shown in [37] for linear losses selected adversarially that there exists a gap in regret lower bounds between the full and bandit information settings which is of the order $\sqrt{p}$. We[1] will consider a specific bandit problem in this chapter and show how structural assumptions on the loss function parameter vector leads to sharper upper bounds on the pseudo regret. The work in this paper first appeared as a technical report on the arXiv repository [78].

9.1 Introduction

We consider the stochastic linear bandit problem [38, 1] which proceeds in rounds $t = 1, \ldots, T$ where at each round t the algorithm selects a vector x_t from some decision set $\mathcal{X} \subset \mathbb{R}^p$ and receives a noisy loss defined as $\ell_t(x_t) = \langle x_t, \theta^* \rangle + \eta_t$ where θ^* is an unknown parameter and η_t is martingale difference noise. The algorithm observes only $\ell_t(x_t)$ at

[1] The work in this chapter was done in collaboration with Vidyashankar Sivakumar.

each round t and its goal is to minimize the cumulative loss $\sum_{t=1}^{T} \ell_t(x_t)$. We measure its performance by the pseudo regret [24] defined as

$$R_T = \sum_{t=1}^{T} \langle x_t, \theta^* \rangle - \operatorname*{argmin}_{x^* \in \mathcal{X}} \sum_{t=1}^{T} \langle x^*, \theta^* \rangle \ . \tag{9.1}$$

The stochastic linear bandit can be used to model problems in several real-world applications ranging from recommender systems to medical treatments to network security. Frequently, in such applications, one has knowledge of the structure of the unknown parameter θ^*, for example, θ^* may be sparse, group sparse, or low-rank. Previous works [38, 1] either made no structural assumptions on θ^* and proved regret bounds[2] of the form $\widetilde{O}(p\sqrt{T})$ or assumed θ^* was s-sparse (s non-zero elements) and showed [2] the regret sharpens to $\widetilde{O}(\sqrt{spT})$. In this chapter, we consider the setting where θ^* is any generally structure vector (sparse, group sparse, low-rank, etc.) such that the structure can be captured by some norm (L_1, $L_{(1,2)}$, nuclear norm, etc.).

Our approach follows previous works [38, 1, 2] which use the *optimism-in-the-face-of-uncertainty* (OFU) principle [24] to design a class of algorithms which construct a confidence ellipsoid C_t such that $\theta^* \in C_t$ across all rounds with high-probability. After which, the algorithm optimistically selects an $x_{t+1} \in \mathcal{X}$ and $\widetilde{\theta}_{t+1} \in C_t$ such that $\langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle \leq \langle x^*, \theta^* \rangle$.

Our algorithm differs from previous algorithms [38, 1, 2] since we select samples uniformly at random from specific subsets of $\mathcal{X}$. More specifically, previous works selected the sample $x_{t+1} \in \mathcal{X}$ and $\widetilde{\theta}_{t+1} \in C_t$ which minimizes the quantity $\langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle$ by solving a bilinear optimization problem however, such an approach only gives one, possibly unique, solution which will not allow our analysis to hold. We build on such works by first solving a similar bilinear optimization problem to get an optimistic x'_{t+1} but then we center a L_2 ball of suitable radius over x'_{t+1} and select x_{t+1} uniformly at random from the ball and deterministically compute $\widetilde{\theta}_{t+1}$.

Overview of Results. The main technical challenge in previous works [38, 1, 2] is constructing confidence ellipsoids which contain θ^* across all rounds with high-probability. The focus of our work is again to construct confidence ellipsoids such that $\theta^* \in C_t$ across all rounds with high-probability but which are general enough to hold for

[2] The $\widetilde{O}(\cdot)$ notation selectively hides constants and log terms.

any norm structured θ^*. Moreover, we desire that the ellipsoids are tighter than previous works in order to provide sharper regret bounds. Previous works [38, 1] constructed the confidence ellipsoids by solving a ridge regression problem to compute an estimate $\hat{\theta}_t$ and centered an ellipsoid over the estimate. We generalize such an approach by instead solving a norm regularized regression problem, e.g., Lasso, given the structure of θ^*. We show our construction of C_t contains θ^* across all rounds with high-probability by extending results in structured estimation [25, 30, 101, 9] which rely on i.i.d. samples.

The main technical result we show is that the radius of our confidence ellipsoids depend on the Gaussian width[3] of sets associated with the structure of θ^* which leads to tighter confidence ellipsoids than previous works [38, 1] when θ^* is structured. For example, with an s-sparse θ^* the radius of the confidence ellipsoid scales as $O(\sqrt{s \log p})$ compared to $O(\sqrt{p})$ in the unstructured settings considered in [38, 1].

The regret bounds for our algorithm follow from the analysis in [38] and depend on the radius of the confidence ellipsoid therefore, our regret bounds scale with the structure of θ^* as measured by the Gaussian width. Recall from Section 3.2 that the regret bounds for an unstructured and s-sparse θ^* were shown in [38, 2] to be $\widetilde{O}(p\sqrt{T})$ and $\widetilde{O}(\sqrt{spT})$ respectively. We show the regret of our algorithms matches such works and further shows how the regret scales for any other type of structure such as group sparse or low-rank which has not been considered in the literature.

9.2 Background: High-Dimensional Structured Estimation

We rely on developments in the analysis of non-asymptotic bounds for structured estimation in high-dimensional statistics. Here, we will discuss the main results needed for our analysis which can be found in the following papers [25, 13, 30, 100, 121, 18, 9, 10].

In high-dimensional structured estimation, one is concerned with settings in which the dimension p of the parameter θ^* to be estimated is significantly larger than the sample size t, i.e., $p \gg t$. It is known that for t i.i.d. Gaussian samples, one can compute an estimate $\hat{\theta}_t$ using least squares regression which converges to θ^* at a rate of $O\left(\sqrt{\frac{p}{t}}\right)$. The convergence rate can be improved when θ^* is structured which is usually

[3] The Gaussian width is a geometric characterization of the size of a set and the definition is presented in Section C.1.

characterized as having a small value according to some norm $R(\cdot)$. For such problems, estimation is performed by solving a norm regularized regression problem

$$\hat{\theta}_t := \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \, \mathcal{L}(\theta, Z_t) + \lambda_t R(\theta) \tag{9.2}$$

where $\mathcal{L}(\cdot, \cdot)$ is a convex loss function[4] , Z_t is a dataset of random i.i.d. pairs $\{(x_i, y_i)\}_{i=1}^t$ where $x_i \in \mathbb{R}^p$ is a sample, $y_i \in \mathbb{R}$ is the response, and λ_t is the regularization parameter.

For such problems, let $\hat{\theta}_t - \theta^*$ be the estimation error vector and note that $\hat{\theta}_t$ is a random vector due to the inherent randomness of the dataset and is realized after solving (9.2). For a suitably large λ_t, [9] showed the random error vector deterministically belongs to the restricted error set

$$E_{r,t} = \left\{ \hat{\theta}_t - \theta^* \in \mathbb{R}^p : R(\hat{\theta}_t) \leq R(\theta^*) + \tfrac{1}{\rho} R(\hat{\theta}_t - \theta^*) \right\} \tag{9.3}$$

where $\rho > 1$ is a constant which we fix as $\rho = 2$ for ease of exposition. For such a ρ, $E_{r,t}$ is a restricted set of directions, in particular, the error vector $\hat{\theta}_t - \theta^*$ cannot be in the direction of θ^*. Using the restricted error set, bounds on the estimation error can be established which hold with high-probability under two assumptions. First, the regularization parameter λ_t must satisfy the inequality

$$\lambda_t \geq 2R^*(\nabla \mathcal{L}(\theta^*, Z_t)) \tag{9.4}$$

where $R^*(\cdot)$ is the dual norm of $R(\cdot)$. Second, the loss function must satisfy the restricted strong convexity (RSC) condition in the restricted error set $E_{r,t}$ as illustrated in [100]. Specifically, there exists a $\kappa > 0$ such that

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \quad \forall \hat{\theta}_t - \theta^* \in E_{r,t} \, . \tag{9.5}$$

For the squared loss, the RSC condition simplifies to the restricted eigenvalue (RE) condition

$$\frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \quad \forall \hat{\theta}_t - \theta^* \in E_{r,t} \tag{9.6}$$

where $X_t \in \mathbb{R}^{t \times p}$ is the design matrix [30]. Under such conditions, the following bound holds with high-probability [100, 9]

$$\|\hat{\theta}_t - \theta^*\|_2 \leq c\psi(E_{r,t}) \frac{\lambda_t}{\kappa} \tag{9.7}$$

[4] We drop the second argument when it is clear from the context.

where $\psi(E_{r,t}) = \sup_{u \in E_{r,t}} \frac{R(u)}{\|u\|_2}$ is the norm compatibility constant and $c > 0$ is a constant. For an s-sparse θ^*, one obtains $\|\hat{\theta}_t - \theta^*\|_2 \leq O\left(\sqrt{\frac{s \log p}{t}}\right)$ and for a group sparse θ^*, one obtains $\|\hat{\theta}_t - \theta^*\|_2 \leq O\left(\sqrt{\frac{s_{\mathcal{G}}(m + \log K)}{t}}\right)$ where K is the number of groups, m is the maximum group size, and $s_{\mathcal{G}}$ is the group sparsity level. Similar bounds can be computed for other types of structure including low-rank.

9.3 Structured Bandits: Problem and Algorithm

Here, we will formally define the problem, mention the assumptions for our analysis, and present our algorithm. The results and analysis are presented in subsequent sections.

9.3.1 Problem Setting

We consider the stochastic linear bandit problem [38, 1] where in each round $t = 1, \ldots, T$ the algorithm selects a vector x_t from the decision set $\mathcal{X} \subset \mathbb{R}^p$ and receives a loss of $\ell_t(x_t) = \langle x_t, \theta^* \rangle + \eta_t$. Our focus is on settings where the unknown parameter $\theta^* \in \mathbb{R}^p$ is structured which we characterize as having a small value according to some norm $R(\cdot)$.

The goal of the algorithm is to minimize its cumulative loss $\sum_t \ell_t(x_t)$ and we measure the performance of the algorithm in terms of the fixed cumulative pseudo regret

$$R_T = \sum_{t=1}^{T} \langle x_t, \theta^* \rangle - \min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} \langle x^*, \theta^* \rangle \ . \tag{9.8}$$

We require that the algorithm's regret grows sublinearly in T, i.e., $R_T = o(T)$, and desire it grows with the structure of θ^* rather than the ambient dimensionality p with high-probability. The following are assumptions under which our analysis holds and most are standard in the literature [38, 1, 2].

Assumptions and Definitions

Assumption 1 The decision set $\mathcal{X} \subset \mathbb{R}^p$ is a compact (closed and bounded) convex set with non-empty interior. For ease of exposition, we assume $\mathcal{X} \subseteq \bar{B}_2^p$, the (closed) unit L_2 ball defined as $\bar{B}_2^p = \{x \in \mathbb{R}^p : \|x\|_2 \leq 1\}$, to avoid scaling factors.

Assumption 2 The noise sequence $\{\eta_1, \ldots, \eta_T\}$ is bounded martingale difference sequence (MDS), i.e., $|\eta_t| \leq B, \mathbb{E}[\eta_t | F_{t-1}] = 0 \; \forall t$ where $F_t = \{x_1, \ldots, x_{t+1}, \eta_1, \ldots, \eta_t\}$ is a filtration (sequence of σ-algebras). We assume bounded noise for simplicity however, the results hold for any sub-Gaussian noise (refer to Section C.1 for the definition).

Assumption 3 We assume the unknown parameter θ^* is fixed for all rounds, the structure is known, for example, for an s-sparse θ^* the value of s is known, and $\|\theta^*\|_2 = 1$.

Assumption 4 The number of rounds T is known a priori.

Assumption 5 There exists a constant $\kappa > 0$ such that the RE condition (9.6) holds for a design matrix X_t where each row is drawn randomly from the intersection of the decision set $\mathcal{X}$ and an L_2 ball with radius which decreases at a rate of the order $\frac{1}{\sqrt{t}}$.

Definition 4 The Gaussian width [30] of a set A is $w(A) = \mathbb{E}\left[\sup_{u \in A} \langle g, u \rangle\right]$ where the expectation is over g which is a zero mean, unit variance Gaussian random variable.

Definition 5 The diameter of the set A is $\phi(A) = \sup_{u,v \in A} \|u - v\|_2 = \sup_{u \in A} 2\|u\|_2$.

Definition 6 Let $E_{r,t} := \left\{\hat{\theta}_t - \theta^* \in \mathbb{R}^p : R(\hat{\theta}_t) \leq R(\theta^*) + \frac{1}{2} R(\hat{\theta}_t - \theta^*)\right\}$ be the restricted error set and $E_{r,\max} = \operatorname{argmax}_{E_r \in \{E_{r,1}, \ldots, E_{r,T}\}} w(E_r)$ the largest such set.

Definition 7 The set $A_t := \operatorname{cone}(E_{r,t}) \cap S^{p-1}$ is a spherical cap where S^{p-1} is the unit sphere in p-dimensions and $A_{\max} := \operatorname{cone}(E_{r,\max}) \cap S^{p-1}$ is the largest such cap.

Definition 8 The unit norm $R(\cdot)$ ball is $\Omega_R := \{u \in \mathbb{R}^p : R(u) \leq 1\}$. With respect to vectors in $E_{r,t}$, the norm compatibility constant at round t is $\psi(E_{r,t}) = \sup_{u \in E_{r,t}} \frac{R(u)}{\|u\|_2}$.

9.3.2 Algorithm

For the initial $t = 1, \ldots, n = c'p$ rounds where $c' > 0$ is a constant, our algorithm selects vectors $x_{1:n} := \{x_1, \ldots, x_n\}$ uniformly at random from $\mathcal{X}$ and receives the losses $\ell_{1:n} := \{\ell_1(x_1), \ldots, \ell_n(x_n)\}$. The random estimation rounds can be considered the "burn-in" period similar to the use of a barycentric spanner or identity matrix as in [38, 1].

After the loss $\ell_n(x_n)$ is received in round n, the algorithm constructs an $(n \times p)$-dimensional design matrix $X_n = [x_1 \ldots x_n]^\top$, a sample covariance matrix $D_n = X_n^\top X_n$, and an n-dimensional response vector $y_n = [\ell_1(x_1) \ldots \ell_n(x_n)]^\top$. The algorithm then computes an estimate $\hat{\theta}_n$ by solving a norm regularized regression problem, constructs a confidence ellipsoid using the Mahalanobis distance defined as $\|\theta - \hat{\theta}_n\|_{2,D_n} =$

Algorithm 9 Structured Stochastic Linear Bandit

Input: p, $\mathcal{X}$, $R(\cdot)$, T, $E_{r,\max}$,Ω_R, γ, c_0, c', C
Set $\beta = C\psi(E_{r,\max})(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\phi(\Omega_R)/2)$ (9.28)
Play $n = c'p$ uniform i.i.d. random vectors $x_{1:n} \in \mathcal{X}$ and receive losses $\ell_{1:n}$
For $t = n, \ldots, T$
  Compute $X_t = [x_1 \ldots x_t]^\top$, $y_t = [\ell_1(x_1) \ldots \ell_t(x_t)]^\top$, and $D_t = X_t^\top X_t$
  Set $\lambda_t = c_0(w(\Omega_R) + \sqrt{\gamma^2 + \log T})/\sqrt{t}$ (9.27)
  Compute $\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\|y_t - X_t\theta\|_2^2 + \lambda_t R(\theta)$
  Construct $C_t := \{\theta : \|\theta - \hat{\theta}_t\|_{2,D_t} \leq \beta\}$
  Compute $(x'_{t+1}, \theta'_{t+1}) := \operatorname{argmin}_{x \in \mathcal{X},\, \theta \in C_t \cap S^{p-1}} \langle x, \theta \rangle$
  Play $x_{t+1} \sim \text{Uniform}\left(\mathcal{X} \cap \bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t})\right)$ and receive loss $\ell_{t+1}(x_{t+1})$
End For

$\sqrt{(\theta - \hat{\theta}_n)^\top D_n(\theta - \hat{\theta}_n)}$, then selects a sample to play. Specifically, the algorithm performs the following four main steps sequentially in each round thereafter.

For each $t = n, \ldots, T$:

1. Compute an estimate: $$\hat{\theta}_t := \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\|y_t - X_t\theta\|_2^2 + \lambda_t R(\theta) \tag{9.9}$$
2. Construct a confidence ellipsoid: $$C_t := \left\{\theta \in \mathbb{R}^p : \|\theta - \hat{\theta}_t\|_{2,D_t} \leq \beta\right\} \tag{9.10}$$
3. Compute an optimal solution: $$(x'_{t+1}, \theta'_{t+1}) := \underset{\substack{x \in \mathcal{X} \\ \theta \in C_t \cap S^{p-1}}}{\operatorname{argmin}} \langle x, \theta \rangle \tag{9.11}$$
4. Play vector $x_{t+1} \sim \text{Uniform}(\mathcal{X} \cap \bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t}))$ and receive loss $\ell_{t+1}(x_{t+1})$

where $\bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t}))$ is a closed L_2 ball centered at x'_{t+1} which has a radius of $\|x'_{t+1}\|_2/2\sqrt{t}$. After receiving $\ell_{t+1}(x_{t+1})$, the design matrix X_{t+1} and response vector y_{t+1} are updated with x_{t+1} and $\ell_{t+1}(x_{t+1})$ respectively, the sample covariance matrix $D_{t+1} = X_{t+1}^\top X_{t+1}$ is recomputed, and the regularization parameter λ_{t+1} is updated.

Discussion

Step 1. An estimate is computed by solving a norm regularized regression problem following existing results discussed in Section 9.2. This generalizes previous works [38, 1] which only consider computing an estimate by solving the ridge regression problem.

Step 2. A confidence ellipsoid is constructed in order to allow the algorithm to explore in certain directions. Since the confidence ellipsoid is defined as $C_t = \{\theta \in \mathbb{R}^p : \|\theta - \hat{\theta}_t\|_{2,D_t} \leq \beta\}$, we focus on bounds for $\|\hat{\theta}_t - \theta^*\|_{2,D_t}$. Extending the results in Section 9.2, we will show in Section 9.5 high-probability bounds on the estimation error of the form $\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq c\psi(E_{r,t})\frac{\lambda_t}{\kappa}\sqrt{t}$. Therefore, setting β to the right hand side will give bounds such that $\theta^* \in C_t$ with high-probability. The value of β then depends on two key terms: the regularization parameter λ_t and the restricted eigenvalue (RE) constant κ detailed in (9.6). The value of λ_t is set by the user and we will provide an explicit characterization of its value in Section 9.5. Moreover, the estimation error bound holds with the RE constant κ under Assumption 5.

Steps 3 and 4. These steps are motivated from the regret analysis established in [38] and to make Assumption 5 reasonable. Let the instantaneous regret at round $t+1$ be defined as $r_{t+1} = \langle x_{t+1}, \theta^* \rangle - \langle x^*, \theta^* \rangle$ where $x^* = \text{argmin}_{x \in \mathcal{X}} \langle x, \theta^* \rangle$. As shown in [38], by selecting an x_{t+1} and $\widetilde{\theta}_{t+1}$ via

$$(x_{t+1}, \widetilde{\theta}_{t+1}) := \underset{\substack{x \in \mathcal{X} \\ \theta \in C_t}}{\text{argmin}} \langle x, \theta \rangle \tag{9.12}$$

the instantaneous regret can be bounded as $r_{t+1} = \langle x_{t+1}, \theta^* \rangle - \langle x^*, \theta^* \rangle \leq \langle x_{t+1}, \theta^* \rangle - \langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle$ because we optimize over both x and θ. Therefore, one obtains the following inequality $\langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle \leq \langle x^*, \theta^* \rangle$ on which the entire regret analysis relies. We will use the regret analysis from [38] therefore, we need to select an x_{t+1} and $\widetilde{\theta}_{t+1}$ such that the above inequality holds. Further, recall the RE condition in (9.6)

$$\frac{1}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \quad \forall \hat{\theta}_t - \theta^* \in E_{r,t} \ .$$

We must have $\kappa > 0$ for the estimation error bound used to compute β to hold. Therefore, in order for such a κ to exist, we need samples which are not too correlated otherwise the design matrix will be ill-conditioned.

To use the regret analysis and satisfy the RE condition, we cannot exactly follow existing work [38, 1, 2] and select an x_{t+1} by solving (9.12) since we may obtain a single unique solution and the rows of the design matrix will be too correlated. Instead, we select samples uniformly at random from specific subsets of $\mathcal{X}$ which spreads the samples out enough such that it is reasonable to assume the RE condition holds. Moreover, as we will show in Section C.3, for any random sample x_{t+1} we select, we can deterministically

compute a $\widetilde{\theta}_{t+1} \in C_t$ such that the inequality $\langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle \leq \langle x^*, \theta^* \rangle$ holds. The only requirement is that we select an x_{t+1} from an L_2 ball which shrinks at a rate of $O(\frac{1}{\sqrt{T}})$ in order to obtain sublinear regret in T.

Steps 1, 2, and 4 can be performed efficiently, in particular, there are several efficient methods for computing the estimate in (9.9) for common regularizers, e.g., L_1, $L_{(1,2)}$, nuclear norm, etc. [43, 105, 19]. Step 3 is computationally difficult in general (similar to all previous work) however, for simple decision sets such as the unit the L_2 ball, a solution can be computed efficiently by solving the corresponding quadratically constrained quadratic program. Our algorithm for structured stochastic linear bandits is presented in Algorithm 9.

9.4 Regret Bound for Structured Bandits

Here, we present the main result which is a high-probability bound on the regret of Algorithm 9 and show examples for popular types of structure. The analysis of the bound is presented in Section 9.5.

First, we recall a few definitions introduced in Section 9.3.1 which will help interpret the main result. We define the set Ω_R as the unit norm $R(\cdot)$ ball and $A_{\max} \subset S^{p-1}$ as the spherical cap of the largest restricted error set. For such sets, $w(\Omega_R)$ and $w(A_{\max})$ are the Gaussian widths. Moreover, we define $\phi(\Omega_R)$ to be the diameter of the set Ω_R, $E_{r,\max}$ as the largest restricted error set, and $\psi(E_{r,\max})$ as the norm compatibility constant of the largest restricted error set. Under such assumptions, we present the main result in a high-level form, which hides the exact nature of the constants involved. A more explicit form of the constants is presented in the appendix.

The main result consists of two theorems for the problem independent and problem dependent settings [38]. Let $\mathcal{E}$ be the set of all extremal points. The problem independent setting occurs when the difference between the expected loss of the best extremal point x^* and the expected loss of the second best extremal point is zero, i.e., $\Delta = \inf_{x \in \mathcal{E}} \langle x, \theta^* \rangle - \langle x^*, \theta^* \rangle = 0$. Such a setting occurs, for example, when the decision set is the unit L_2 ball. The problem dependent setting occurs when $\Delta > 0$, for example, when the decision set is a polytope.

Theorem 9 (Problem Independent Regret Bound) *Under the Section 9.3.1 assumptions and for any $\gamma > 0$, choose the radius of the ellipsoid in Algorithm 9 as*

$$\beta = c_0 \psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \frac{\phi(\Omega_R)}{2} \right) . \tag{9.13}$$

Then, for any $T > c'p$, with probability at least $1 - c_1 \exp(-\gamma^2)$, the fixed cumulative regret of Algorithm 9 is at most

$$R_T \leq O\left(\psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \right) \sqrt{p} \sqrt{T \log T} \right) , \tag{9.14}$$

where $c', c_0, c_1 > 0$ are constants.

Theorem 10 (Problem Dependent Regret Bound) *Under the Section 9.3.1 assumptions and for any $\gamma > 0$, choose the radius of the ellipsoid in Algorithm 9 as*

$$\beta = c_0 \psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \frac{\phi(\Omega_R)}{2} \right) . \tag{9.15}$$

Then, for any $T > c'p$, with probability at least $1 - c_1 \exp(-\gamma^2)$, the fixed cumulative regret of Algorithm 9 with a decision set which has non-zero gap $\Delta > 0$ is at most

$$R_T \leq O\left(\psi^2(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \right)^2 p \log T / \Delta \right) , \tag{9.16}$$

where $c', c_0, c_1 > 0$ are constants.

9.4.1 Examples

We present the problem independent regret of popular types of structured θ^* using Theorem 9 and the values of $\psi(E_{r,\max})$ and $w(\Omega_R)$ from [30, 10, 31]. The problem dependent regret can be similarly computed. Only unstructured and sparse structures have been considered [38, 1, 2, 26]. No previous works have considered any other types of structure including group sparse and low-rank.

Example 1 (Unstructured) For problems where θ^* is not structured, we simply use $R(\theta) = \|\theta\|_2^2$ and solve the ridge regression problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t} \|y_t - X_t \theta\|_2^2 + \lambda_t \|\theta\|_2^2 = \left(X_t^\top X_t + \lambda_t \mathbb{I}_{p \times p} \right)^{-1} X_t^\top y_t. \tag{9.17}$$

We compute the regret by plugging in the values $\psi(E_{r,\max}) = O(1)$ and $w(\Omega_R) = O(\sqrt{p})$ to obtain a regret of $\widetilde{O}(p\sqrt{T})$. The regret matches [38, 1] up to log and constant factors.

Example 2 (Sparse) For problems where θ^* is s-sparse (s non-zeros), one common regularizer to induce sparse solutions is $R(\theta) = \|\theta\|_1$. With such a regularizer, we solve the Lasso problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\|y_t - X_t\theta\|_2^2 + \lambda_t\|\theta\|_1 \ . \tag{9.18}$$

We compute the regret by plugging in the values $\psi(E_{r,\max}) = O(\sqrt{s})$ and $w(\Omega_R) = O(\sqrt{\log p})$ to obtain a regret of $\widetilde{O}(\sqrt{s \log p}\sqrt{p}\sqrt{T})$ which matches [2] up to log and constant factors. Note, it is worse than the regret from [26] which is $\widetilde{O}(s\sqrt{T})$ however, they consider a different noise model in the loss function.

Example 3 (Group Sparse) Let $\{1, \ldots, p\}$ be an index set of θ^*, $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$ be a known set of K groups which define a disjoint partitioning of the index set. For group sparse problems, one common regularizer is $R(\theta) = \sum_{i=1}^K \|\theta_{\mathcal{G}_i}\|_2$ where $\theta_{\mathcal{G}_i}$ is a vector with elements equal to θ for indices in $\mathcal{G}_i$ and 0 otherwise. With such a regularizer, we solve the group lasso problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\|y_t - X_t\theta\|_2^2 + \lambda_t \sum_{i=1}^K \|\theta_{\mathcal{G}_i}\|_2 \ . \tag{9.19}$$

With maximum group size $m = \max_i |\mathcal{G}_i|$ and subset $\mathcal{S}_\mathcal{G} \subset \{1, \ldots, K\}$ of the groups with cardinality $s_\mathcal{G}$ which denotes the number of active groups, we compute the regret by plugging in the values $\psi(E_{r,\max}) = O(\sqrt{s_\mathcal{G}})$ and $w(\Omega_R) = O(\sqrt{m + \log K})$ to obtain a regret of $\widetilde{O}(\sqrt{s_\mathcal{G}(m + \log K)}\sqrt{p}\sqrt{T})$.

Example 4 (Low-Rank) Let $\Theta^* \in \mathbb{R}^{d \times p}$ be a matrix with rank r and we select the matrix $X_t \in \mathbb{R}^{d \times p}$ at each round. Define the loss we receive as $\ell_t(X_t) = \operatorname{trace}(X_t^\top \Theta^*) + \eta_t$. For problems where the rank of Θ^* is small, for example, $r \leq \min(d, p)$, one common regularizer to use is the nuclear norm $R(\Theta) = \|\Theta\|_* = \sum_{j=1}^{\min\{d,p\}} \sigma_j(\Theta)$ where $\sigma_j(\Theta)$ are the singular values of the Θ. With such a regularizer, we solve the trace-norm regularized least squares problem

$$\hat{\Theta}_t = \underset{\Theta \in \mathbb{R}^{d \times p}}{\operatorname{argmin}} \frac{1}{t}\sum_{i=1}^t \Big(y_i - \operatorname{trace}(X_i^\top \Theta)\Big)^2 + \lambda_t\|\Theta\|_* \ . \tag{9.20}$$

We compute the regret by plugging in the values $\psi(E_{r,\max}) = O(\sqrt{r})$ and $w(\Omega_R) = O(\sqrt{d + p})$ from [101] to obtain a regret of $\widetilde{O}(\sqrt{r(d + p)}\sqrt{dp}\sqrt{T})$.

9.5 Overview of the Analysis

The analysis starts from a regret result established in [38]. First, denote $r_t = \langle x_t, \theta^* \rangle - \langle x^*, \theta^* \rangle$ as the instantaneous regret acquired by the algorithm on round t where the optimal vector x^* is defined as $x^* = \text{argmin}_{x \in \mathcal{X}} \langle x, \theta^* \rangle$. Then for Algorithm 9, as long as we have $\theta^* \in C_t$ over all rounds t, [38, Theorem 6] shows that $\sum_{t=1}^{T} r_t^2 \leq 8\beta^2 p \log T$. Then, to establish a problem independent regret bound we directly apply the Cauchy-Schwarz inequality to get

$$R_T = \sum_{t=1}^{T} r_t \leq \left(T \sum_{t=1}^{T} r_t^2 \right)^{1/2} \leq \beta \sqrt{8pT \log T} \ , \tag{9.21}$$

which holds conditioned on $\theta^* \in C_t$ over all rounds t. Moreover, for a problem dependent regret bound, we follow the proof of [38, Theorem 1] which shows

$$R_T = \sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T} \frac{r_t^2}{\Delta} \leq \frac{8p\beta^2 \log T}{\Delta} \ , \tag{9.22}$$

which holds conditioned on $\theta^* \in C_t$ over all rounds t.

The focus of our analysis is then to choose a β such that the condition holds with high-probability uniformly over all rounds. From Algorithm 9, since $C_t := \{\theta : \|\theta - \hat{\theta}_t\|_{2,D_t} \leq \beta\}$ and we want to have $\theta^* \in C_t$, we focus on bounds for $\|\hat{\theta}_t - \theta^*\|_{2,D_t}$, the instantaneous estimation error. Building on ideas for high-dimensional structured estimation as discussed in Section 9.2, deterministic bounds on the instantaneous estimation error can be obtained under two assumptions. First, we need to choose the regularization parameter λ_t such that

$$\lambda_t \geq 2R^* \left(\frac{1}{t} X_t^\top (y_t - X_t \theta^*) \right) \ . \tag{9.23}$$

Second, for all $\hat{\theta}_t - \theta^* \in E_{r,t}$, we need to have the restricted eigenvalue (RE) condition for constant $\kappa > 0$

$$\inf_{\hat{\theta}_t - \theta^* \in E_{r,t}} \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \ . \tag{9.24}$$

Under these two assumptions, following existing analysis for high-dimensional estimation, we have the following theorem (refer to Section C.2 for the proof).

Theorem 11 *Assume that $\hat{\theta}_t - \theta^* \in E_{r,t}$, the RE condition is satisfied in $E_{r,t}$ with parameter κ, and λ_t is suitably large. Then for any norm $R(\cdot)$ and constant $c > 0$*

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq c\psi(E_{r,t})\frac{\lambda_t}{\sqrt{\kappa}}\sqrt{t} \ . \tag{9.25}$$

In Section C.4 we show that the assumption in (9.23) holds with high-probability and with Assumption 5 this implies Theorem 9.25 holds with high-probability. In particular, for the assumption in (9.23), we show the following result.

Theorem 12 *For any $\gamma > 0$ and for absolute constant $L > 0$, with probability at least $1 - L\exp(-\gamma^2)$, the following bound holds uniformly for all rounds $t = 1, \ldots, T$:*

$$R^*\left(\frac{1}{t}X_t^\top(y_t - X_t\theta^*)\right) \leq 2LB\frac{\left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2}\right)}{\sqrt{t}} \ . \tag{9.26}$$

Then, from (9.23), for $c_0 = 4LB$ we set λ_t as

$$\lambda_t \geq c_0\frac{\left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2}\right)}{\sqrt{t}} \ . \tag{9.27}$$

For a bound on the instantaneous ellipsoidal estimation error, we plug in the value of λ_t from (9.27) into (9.25) and use the norm compatibility constant of the largest restricted error set to obtain

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq C\psi(E_{r,\max})\left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2}\right)$$

where $C = c_0 c/\sqrt{\kappa}$ is a constant which holds with high-probability across all rounds $t = 1, \ldots, T$. Therefore, if we set

$$\beta = C\psi(E_{r,\max})\left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2}\right) \tag{9.28}$$

the confidence ellipsoid C_t will contain θ^* across all rounds with high-probability. Substituting our β into the regret bounds in (9.21) and (9.22) gives our main result in Theorem 9 and Theorem 10.

Chapter 10

Structured Stochastic Generalized Linear Bandits

In Chapter 9 we only considered linear loss functions for the stochastic linear bandit problem. In this chapter, we extend such results and consider certain non-linear loss functions following the framework of Generalized Linear Models in statistics.

10.1 Introduction

We begin by recalling the stochastic linear bandit problem [38, 1, 2, 78] and provide a few more details which will make the contributions in this chapter clear. The stochastic linear bandit problem proceeds in rounds $t = 1, \dots, T$ where at each round the algorithm selects a vector x_t from a decision set $\mathcal{X} \subset \mathbb{R}^p$, observes only the loss function value $f_t(x_t) \in \mathbb{R}$, and suffers a loss of $f_t(x_t)$. Previous works [38, 1, 2, 78] assumed $\mathbb{E}[f_t(x_t)|x_t] = \langle x_t, \theta^* \rangle$ where $\theta^* \in \mathbb{R}^p$ is an unknown vector. Under such an assumption, the noise $\eta_t = f_t(x_t) - \mathbb{E}[f_t(x_t)|x_t]$ so $\{\eta_1, \dots, \eta_T\}$ is a martingale difference sequence by construction and we rewrite the loss as $\ell_t(x_t) = \langle x_t, \theta^* \rangle + \eta_t$ which is the form we used in Chapter 9. The goal of the algorithm is to minimize its cumulative loss $\sum_{t=1}^{T} \ell_t(x_t)$ and we measure the performance by the pseudo regret (hereafter referred to as regret)

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t)\right] - \operatorname*{argmin}_{x^* \in \mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x^*)\right] . \tag{10.1}$$

Previous works [38, 1] have introduced algorithms which incur a regret of $\widetilde{O}(p\sqrt{T})$ which shows the regret depends linearly on the ambient dimensionality p and sublinearly on the time horizon T. Under structural assumptions on θ^*, e.g., θ^* is sparse, group sparse, low-rank, etc., we introduced an algorithm in Chapter 9 and showed sharper regret bounds of the order $\widetilde{O}(\psi(E_{r,\max})w(\Omega_R)\sqrt{pT})$. Each of the previous works [38, 1, 2, 78] make the assumption that the expected loss is a linear function, specifically, $\mathbb{E}[f_t(x_t)|x_t] = \langle x_t, \theta^* \rangle$ which implicitly assumes the loss is drawn i.i.d. from a Gaussian distribution. However, in many applications which can be modeled using the stochastic linear bandit (recommendations, medical treatments, robot exploration, etc.) such an assumption is inappropriate. For example, when recommending a news article, the loss function is the feedback the user provides which is a binary click or no-click event. The loss in such a problem is drawn from a Bernoulli distribution and not a Gaussian. In medical treatments, one may be interested in predicting the number of occurrences or time between recurrences of cancer which can more accurately be assumed to be drawn from a Poisson or exponential distribution.

In this chapter, we relax the assumption previous works make and follow the classical Generalized Linear Model (GLM) framework and assume the loss can be drawn from any distribution in the exponential family and the expected loss is defined by a (possibly non-linear) inverse link function, i.e., $\mathbb{E}[f_t(x_t)|x_t] = g^{-1}(\langle x_t, \theta^* \rangle)$. The setting in previous works is a special case of our setting which we can see by setting the link function $g(\cdot)$ to be the identity function which gives an inverse link function of $g^{-1}(\langle x_t, \theta^* \rangle) = \langle x_t, \theta^* \rangle$. For the examples above, we can set the link function to be the logit, log, or inverse functions which give the following inverse link functions: $g^{-1}(\langle x_t, \theta^* \rangle) = \frac{1}{1+\exp(\langle x_t, \theta^* \rangle)}$, $g^{-1}(\langle x_t, \theta^* \rangle) = \exp(\langle x_t, \theta^* \rangle)$, and $g^{-1}(\langle x_t, \theta^* \rangle) = \langle x_t, \theta^* \rangle^{-1}$ respectively. No previous works have considered the stochastic linear bandit problem under the GLM setting with or without structural assumptions on θ^*.

Overview of Results: We present an algorithm which follows previous works [38, 1, 2, 78] and uses the *optimism-in-the-face-of-uncertainty* (OFU) principle [24]. Similar to our algorithm in Chapter 9 our algorithm constructs a confidence ellipsoid C_t in each round which contains θ^* with high-probability then optimistically selects an $x_{t+1} \in \mathcal{X}$ and $\widetilde{\theta}_{t+1} \in C_t$ such that $\langle x_{t+1}, \widetilde{\theta}_{t+1} \rangle \leq \langle x^*, \theta^* \rangle$ in each round. We follow and extend the analysis presented in Chapter 9 and the main technical challenge is showing how

to construct tight confidence ellipsoids where the radius of such ellipsoids depends on the structure of θ^*. Such a challenge requires showing how to bound certain empirical stochastic processes which we show can be accomplished using recent results in structured estimation [9, 10, 98] which rely on techniques such as generic chaining [115, 116]. Using the regret analysis from [38], we show the regret of our algorithm matches previous works even under the GLM setting and is of the form $\widetilde{O}(\psi(E_{r,\max})w(\Omega_R)\sqrt{pT})$. Our results show that the problem is no more difficult than with linear losses and, for an unstructured θ^*, matches the lower bound [38].

10.2 Background: Structured Estimation and Generalized Linear Models

In this section, we briefly recall some background material on high-dimensional structured estimation which was also presented in Section 9.2. We also introduce material on Generalized Linear Models (GLMs) all of which can be found in numerous sources including [102, 11, 23].

10.2.1 Structured Estimation

Recall in high-dimensional structured estimation, one is concerned with estimating an unknown parameter $\theta^* \in \mathbb{R}^p$ which is assumed to be structured. We define a parameter to be structured if it has a small value according to some norm, i.e., the structure can be suitably captured by a norm. The estimation can be performed by solving a norm regularized regression problem

$$\hat{\theta}_t := \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \, \mathcal{L}(\theta, Z_t) + \lambda_t R(\theta) \tag{10.2}$$

where $\mathcal{L}(\cdot, \cdot)$ is a convex loss function, Z_t is a dataset consisting of random i.i.d. pairs $\{(x_i, y_i)\}_{i=1}^t$ where $x_i \in \mathbb{R}^p$ is a sample, $y_i \in \mathbb{R}$ is the response, and λ_t is the regularization parameter. Let $\hat{\theta}_t - \theta^*$ be the estimation error vector and for a suitably large λ_t, [9] showed the random error vector lives in the restricted error set

$$E_{r,t} = \left\{ \hat{\theta}_t - \theta^* \in \mathbb{R}^p : R(\hat{\theta}_t) \leq R(\theta^*) + \tfrac{1}{\rho} R(\hat{\theta}_t - \theta^*) \right\} \tag{10.3}$$

where $\rho > 1$ is a constant which we set as $\rho = 2$ for ease of exposition. Now, bounds on the estimation error can be established which hold with high-probability under two assumptions. First, the regularization parameter λ_t must satisfy the inequality

$$\lambda_t \geq 2R^*(\nabla \mathcal{L}(\theta^*, Z_t)) \tag{10.4}$$

where $R^*(\cdot)$ is the dual norm of $R(\cdot)$. Second, the loss function must satisfy the restricted strong convexity (RSC) condition in the restricted error set $E_{r,t}$ as illustrated in [100]. Specifically, the RSC condition implies that there exists a $\kappa > 0$ such that

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \quad \forall \hat{\theta}_t - \theta^* \in E_{r,t} \ . \tag{10.5}$$

Under such conditions, the following bound holds with high-probability [100, 9]

$$\|\hat{\theta}_t - \theta^*\|_2 \leq c\psi(E_{r,t}) \frac{\lambda_t}{\kappa} \tag{10.6}$$

where $\psi(E_{r,t}) = \sup_{u \in E_{r,t}} \frac{R(u)}{\|u\|_2}$ is the norm compatibility constant and $c > 0$ is a constant.

10.2.2 Generalized Linear Models

Generalized Linear Models (GLMs) [102, 11, 23] predict the expected value of a response variable y_t using three key components: a linear predictor $\eta_t = \langle x_t, \theta^* \rangle$, a probability distribution $p(y_t|\eta_t)$, and a link function $g(\cdot)$. GLMs assume the response variable y_t is drawn from an exponential family distribution $p(y_t|\eta_t)$ and that y_t is conditionally independent given y_i and x_i for $i = 1, \ldots, t-1$. The exponential family is a set of distributions where the conditional distribution can be written in the canonical form

$$p(y_t|\eta) = h(y_t) \exp\left(\eta_t T(y_t) - \varphi(\eta_t)\right)$$

where η_t is the natural parameter, $T(y_t)$ is a sufficient statistic of the distribution, $\varphi(\cdot)$ is the log-partition function which is strictly convex and assumed to be twice continuously differentiable, and $h(y_t)$ is a known function. Assuming the natural parameter is defined as $\eta_t = \langle x_t, \theta^* \rangle$ and $T(\cdot)$ is the identity function, the exponential family is then given by

$$p(y_t|\langle x_t, \theta^* \rangle) = h(y_t) \exp\left(y_t \langle x_t, \theta^* \rangle - \varphi(\langle x_t, \theta^* \rangle)\right) \ .$$

The log-partition function ensures $p(y_t|\langle x_t, \theta^* \rangle)$ is a probability distribution and is defined as

$$\varphi(\langle x_t, \theta^* \rangle) = \log \left(\int_{y_t} h(y_t) \exp \left(y_t \langle x_t, \theta^* \rangle \right) \right) \mathrm{d}y_t \ .$$

The expected value of the response variable can be computed from the log-partition function as

$$\mathbb{E}[y_t|x_t] = \varphi'(\langle x_t, \theta^* \rangle) \ .$$

where we denote the first and second derivatives of $\varphi(\cdot)$ with a prime and double prime respectively. GLMs relate the linear predictor $\langle x_t, \theta^* \rangle$ with the expected value of the response $\mathbb{E}[y_t|x_t]$ through what is called the inverse link function. The inverse link function $g^{-1} : \mathbb{R} \to \mathbb{R}$ can be defined as $\mathbb{E}[y_t|x_t] = g^{-1}(\langle x_t, \theta^* \rangle) = \varphi'(\langle x_t, \theta^* \rangle)$ therefore, it is continuously differentiable and a strictly increasing function.

Estimation of the parameters θ^* of the linear predictor is typically performed using maximum likelihood estimation where the loss function is the negative log likelihood

$$\mathcal{L}(\theta) = -\frac{1}{t} \sum_{i=1}^{t} \log p(y_i|x_i) = \frac{1}{t} \sum_{i=1}^{t} \varphi(\langle x_i, \theta \rangle) - y_i \langle x_i, \theta \rangle \ . \tag{10.7}$$

When θ^* is structured we compute an estimate by solving (10.2) using the negative log likelihood loss following [9, 10]

$$\hat{\theta}_t := \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{t} \sum_{i=1}^{t} \{ \varphi(\langle x_i, \theta \rangle) - y_i \langle x_i, \theta \rangle \} + \lambda_t R(\theta) \ .$$

10.3 Problem Setting and Algorithm

Here, we will formally define the problem, mention the assumptions under which our analysis works, and present our algorithm. The results and analysis are presented in subsequent sections.

We consider the stochastic linear bandit problem [38, 1] where in each round $t = 1, \ldots, T$ the algorithm selects a p-dimensional vector x_t from the decision set $\mathcal{X}$ and receives a loss of $f_t(x_t)$. We assume the expected loss is defined by $\mathbb{E}[f_t(x_t)|x_t] = g^{-1}(\langle x_t, \theta^* \rangle)$ where $g^{-1}(\cdot)$ is an inverse link function and $\theta^* \in \mathbb{R}^p$ is an unknown parameter vector. Let $\eta_t = f_t(x_t) - \mathbb{E}[f_t(x_t)|x_t]$ be the noise which implies $\{\eta_1, \ldots, \eta_T\}$ is a martingale difference sequence (MDS) and rewrite the loss as $\ell_t(x_t) = g^{-1}(\langle x_t, \theta^* \rangle) + \eta_t$.

We further consider settings when θ^* is structured which we define as having a small value according to some norm $R(\cdot)$. The goal of the algorithm is to minimize its cumulative loss $\sum_t \ell_t(x_t)$ and we measure the performance of the algorithm in terms of the fixed cumulative pseudo regret

$$R_T = \sum_{t=1}^{T} g^{-1}(\langle x_t, \theta^* \rangle) - \min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} g^{-1}(\langle x^*, \theta^* \rangle) \ . \tag{10.8}$$

We require that the algorithm's regret grows sublinearly in T, i.e., $R_T = o(T)$, and with the structure of θ^* rather than the ambient dimensionality p with high-probability.

In addition to the assumptions in Section 9.3.1, the following is an additional assumption needed for the analysis.

Assumption 6 The log-partition function $\varphi(\cdot)$ is twice continuously differentiable and (locally) Lipschitz continuous, i.e., $g^{-1}(\cdot) = \varphi'(\cdot) \leq G$ for constant G, and there exists a constant $\ell > 0$ such that $\varphi''(\cdot) \geq \ell$.

10.3.1 Algorithm

The algorithm is quite similar to Algorithm 9 presented in Chapter 9 however, here we must solve a norm regularized regression problem which does not always use the squared loss as in Algorithm 9. Specifically. for the initial $t = 1, \ldots, n = c'p$ rounds where $c' > 0$ is a constant, our algorithm selects vectors $x_{1:n} := \{x_1, \ldots, x_n\}$ uniformly at random from $\mathcal{X}$ and receives the corresponding losses $\ell_{1:n} := \{\ell_1(x_1), \ldots, \ell_n(x_n)\}$. After the loss $\ell_n(x_n)$ is received in round n, the algorithm constructs an $(n \times p)$-dimensional design matrix $X_n = [x_1 \ldots x_n]^\top$, a sample covariance matrix $D_n = X_n^\top X_n$, and an n-dimensional response vector $y_n = [\ell_1(x_1) \ldots \ell_n(x_n)]^\top$. The algorithm then computes an estimate $\hat{\theta}_n$ by solving a norm regularized regression problem, constructs a confidence ellipsoid using the Mahalanobis distance defined as $\|\theta - \hat{\theta}_n\|_{2,D_n} = \sqrt{(\theta - \hat{\theta}_n)^\top D_n (\theta - \hat{\theta}_n)}$, then selects a sample to play. Specifically, the algorithm performs the following four main steps sequentially in each round thereafter.

Algorithm 10 Structured Stochastic Generalized Linear Bandit

Input: p, $\mathcal{X}$, $R(\cdot)$, T, $E_{r,\max}$, Ω_R, γ, c_0, c', C
Set $\beta = C\psi(E_{r,\max})(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\phi(\Omega_R)/2)$ (10.25)
Play $n = c'p$ uniform i.i.d. random vectors $x_{1:n} \in \mathcal{X}$ and receive losses $\ell_{1:n}$
For $t = n, \ldots, T$
  Compute $X_t = [x_1 \ldots x_t]^\top$, $y_t = [\ell_1(x_1) \ldots \ell_t(x_t)]^\top$, and $D_t = X_t^\top X_t$
  Set $\lambda_t = c_0(w(\Omega_R) + \sqrt{\gamma^2 + \log T})/\sqrt{t}$ (9.27)
  Compute $\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\sum_{i=1}^{t} \{\varphi(\langle x_t, \theta\rangle) - y_i\langle x_i, \theta\rangle\} + \lambda_t R(\theta)$
  Construct $C_t := \{\theta : \|\theta - \hat{\theta}_t\|_{2,D_t} \le \beta\}$
  Compute $(x'_{t+1}, \theta'_{t+1}) := \operatorname{argmin}_{x \in \mathcal{X},\, \theta \in C_t \cap S^{p-1}} \langle x, \theta\rangle$
  Play $x_{t+1} \sim \text{Uniform}\left(\mathcal{X} \cap \bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t})\right)$ and receive loss $\ell_{t+1}(x_{t+1})$
End For

For each $t = n, \ldots, T$:

1. Compute an estimate: $\hat{\theta}_t := \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t}\sum_{i=1}^{t} \{\varphi(\langle x_t, \theta\rangle) - y_t\langle x_t, \theta\rangle\} + \lambda_t R(\theta)$ (10.9)
2. Construct a confidence ellipsoid: $C_t := \left\{\theta \in \mathbb{R}^p : \|\theta - \hat{\theta}_t\|_{2,D_t} \le \beta\right\}$ (10.10)
3. Compute an optimal solution: $(x'_{t+1}, \theta'_{t+1}) := \underset{\substack{x \in \mathcal{X} \\ \theta \in C_t \cap S^{p-1}}}{\operatorname{argmin}} \langle x, \theta\rangle$ (10.11)
4. Play vector $x_{t+1} \sim \text{Uniform}(\mathcal{X} \cap \bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t}))$ and receive loss $\ell_{t+1}(x_{t+1})$

where $\bar{B}_2^p(x'_{t+1}, \|x'_{t+1}\|_2/2\sqrt{t}))$ is a closed L_2 ball centered at x'_{t+1} which has a radius of $\|x'_{t+1}\|_2/2\sqrt{t}$. After receiving $\ell_{t+1}(x_{t+1})$, the design matrix X_{t+1} and response vector y_{t+1} are updated with x_{t+1} and $\ell_{t+1}(x_{t+1})$ respectively, the sample covariance matrix $D_{t+1} = X_{t+1}^\top X_{t+1}$ is recomputed, and the regularization parameter λ_{t+1} is updated. Our structured stochastic generalized linear bandits algorithm is presented in Algorithm 10.

10.4 Regret Bound

Here, we present the main result which is a high-probability bound on the regret of Algorithm 10 and show examples for popular types regression problems and structures. The analysis of the bound is presented in Section 10.5.

The main result consists of two theorems for the problem independent and problem dependent settings [38]. Let $\mathcal{E}$ be the set of all extremal points. The problem independent setting occurs when the difference between the expected loss of the best extremal point x^* and the expected loss of the second best extremal point is zero, i.e., $\Delta = \inf_{x \in \mathcal{E}} g^{-1}(\langle x, \theta^* \rangle) - g^{-1}(\langle x^*, \theta^* \rangle) = 0$. Such a setting occurs, for example, when the decision set is the unit L_2 ball. The problem dependent setting occurs when $\Delta > 0$, for example, when the decision set is a polytope.

Theorem 13 (Problem Independent Regret Bound) *Under the assumptions in Section 9.3.1, Assumption 6, and for any $\gamma > 0$, choose the radius of the ellipsoid in Algorithm 10 as*

$$\beta = c_0 \psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \frac{\phi(\Omega_R)}{2} \right) . \tag{10.12}$$

Then, for any $T > c'p$, with probability at least $1 - c_1 \exp(-\gamma^2)$, the fixed cumulative regret of Algorithm 10 is at most

$$R_T \leq O\left(\psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \right) \sqrt{p} \sqrt{T \log T} \right) , \tag{10.13}$$

where $c', c_0, c_1 > 0$ are constants.

Theorem 14 (Problem Dependent Regret Bound) *Under the assumptions in Section 9.3.1, Assumption 6, and for any $\gamma > 0$, choose the radius of the ellipsoid in Algorithm 10 as*

$$\beta = c_0 \psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \frac{\phi(\Omega_R)}{2} \right) . \tag{10.14}$$

Then, for any $T > c'p$, with probability at least $1 - c_1 \exp(-\gamma^2)$, the fixed cumulative regret of Algorithm 10 with a decision set which has non-zero gap $\Delta > 0$ is at most

$$R_T \leq O\left(\psi^2(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T} \right)^2 p \log T / \Delta \right) , \tag{10.15}$$

where $c', c_0, c_1 > 0$ are constants.

10.4.1 Examples

We present the problem independent regret of popular regression problems: linear regression, logistic regression, and Poisson regression each with various types of structure. We compute the regret using Theorem 13 and the values of $\psi(E_{r,\max})$ and $w(\Omega_R)$ from [30, 10, 31]. In the following examples, we consider three types of structure: unstructured, s-sparse, and group sparse. For unstructured, we assume θ^* has no particular structure. For s-sparse structure, we assume θ^* has exactly s non-zero elements. For group sparse structure, let $\{1, \dots, p\}$ be an index set of θ^*, $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_K\}$ be a known set of K groups which define a disjoint partitioning of the index set, $m = \max_i |\mathcal{G}_i|$ be the maximum group size, and $\mathcal{S}_\mathcal{G} \subset \{1, \dots, K\}$ be a subset of the groups which has cardinality $s_\mathcal{G}$ and denotes the number of active groups. We use the notation $\theta_{\mathcal{G}_i}$ to denote a vector with elements equal to θ for indices in $\mathcal{G}_i$ and 0 otherwise.

Note, the regret for linear regression has been shown in [38, 1, 2, 78] for all types of structure we consider. The regret for logistic regression has only been considered without any structural assumptions in [129]. The remaining examples are novel contributions.

Example 5 (Linear Regression) For problems where the loss $\ell_t(\cdot)$ is a real-valued function with support in $[-C, C]$ for some constant C, we can set the inverse link function to be the identity function, i.e., $g^{-1}(\langle x_t, \theta \rangle) = \langle x_t, \theta \rangle$ which implies the conditional probability distribution $p(y_t|x_t)$ is Gaussian. The estimation loss function $\mathcal{L}(\theta)$ reduces to squared loss and we solve the norm regularized linear regression problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t} \|y_t - X_t \theta\|_2^2 + \lambda_t R(\theta) \ . \tag{10.16}$$

If θ^* is unstructured then we set $R(\theta) = \|\theta\|_2^2$ and (10.16) becomes the ridge regression problem, if θ^* is s-sparse then we set $R(\theta) = \|\theta\|_1$ and (10.16) becomes the Lasso problem [119], and if θ^* is group sparse then we set $R(\theta) = \sum_{j=1}^K \|\theta_{\mathcal{G}_j}\|_2$ and (10.16) becomes the group lasso problem [128].

We compute the regret by plugging in the values of $\psi(E_{r,\max})$ and $w(\Omega_R)$. For ridge regression: $\psi(E_{r,\max}) = O(1)$, $w(\Omega_R) = O(\sqrt{p})$ which gives a regret of $\widetilde{O}(p\sqrt{T})$. For Lasso: $\psi(E_{r,\max}) = O(\sqrt{s})$, $w(\Omega_R) = O(\sqrt{\log p})$ which gives a regret of $\widetilde{O}(\sqrt{s \log p}\sqrt{pT})$. For group lasso: $\psi(E_{r,\max}) = O(\sqrt{s_\mathcal{G}})$, $w(\Omega_R) = O(\sqrt{m + \log K})$ which gives a regret of $\widetilde{O}(\sqrt{s_\mathcal{G}(m + \log K)}\sqrt{pT})$

Example 6 (Logistic Regression) For problems where the loss $\ell_t(\cdot)$ is binary, we can set the inverse link function to be the logistic function, i.e., $g^{-1}(\langle x_t, \theta \rangle) = \frac{1}{1+\exp(\langle x_t, \theta \rangle)}$ which implies the conditional probability distribution $p(y_t|x_t)$ is Bernoulli. The estimation loss function $\mathcal{L}(\theta)$ is the well-known logistic loss function and we solve the norm regularized logistic regression problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} \left\{ \log\left(1 + \exp\left(\langle x_i, \theta \rangle\right)\right) - y_i \langle x_i, \theta \rangle \right\} + \lambda_t R(\theta) \ . \tag{10.17}$$

If θ^* is unstructured then we set $R(\theta) = \|\theta\|_2^2$ and (10.17) becomes logistic regression with L_2 regularization, if θ^* is s-sparse then we set $R(\theta) = \|\theta\|_1$ and (10.17) becomes the sparse logistic regression problem [86], and if θ^* is group sparse then we set $R(\theta) = \sum_{j=1}^{K} \|\theta_{\mathcal{G}_j}\|_2$ and (10.17) becomes the group sparse logistic regression problem [94].

The regret for each of the problems: logistic regression with L_2 regularization, sparse logistic regression, and group sparse logistic regression is $\widetilde{O}(p\sqrt{T})$, $\widetilde{O}(\sqrt{s \log p}\sqrt{pT})$, and $\widetilde{O}(\sqrt{s_{\mathcal{G}}(m + \log K)}\sqrt{pT})$ respectively which is the same as the linear regression example.

Example 7 (Poisson Regression) For problems where the loss $\ell_t(\cdot)$ is an integer, we can set the inverse link function to be the exponential function, i.e., $g^{-1}(\langle x_t, \theta \rangle) = \exp(\langle x_t, \theta \rangle)$ which implies the conditional probability distribution $p(y_t|x_t)$ is Poisson and we solve the norm regularized Poisson regression problem

$$\hat{\theta}_t = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} \left\{ \exp(\langle x_i, \theta \rangle) - y_i \langle x_i, \theta \rangle \right\} + \lambda_t R(\theta) \ . \tag{10.18}$$

If θ^* is unstructured then we set $R(\theta) = \|\theta\|_2^2$ and (10.18) becomes Poisson regression with L_2 regularization, if θ^* is s-sparse then we set $R(\theta) = \|\theta\|_1$ and (10.18) becomes the sparse Poisson regression problem [73], and if θ^* is group sparse then we set $R(\theta) = \sum_{j=1}^{K} \|\theta_{\mathcal{G}_j}\|_2$ and (10.18) becomes the group sparse Poisson regression problem [73].

The regret for each of the problems: Poisson regression with L_2 regularization, sparse Poisson regression, and group sparse Poisson regression is $\widetilde{O}(p\sqrt{T})$, $\widetilde{O}(\sqrt{s \log p}\sqrt{pT})$, and $\widetilde{O}(\sqrt{s_{\mathcal{G}}(m + \log K)}\sqrt{pT})$ respectively which is the same as the linear regression example.

10.5 Overview of the Analysis

The analysis starts by first showing a bound on the non-linear instantaneous regret.

Lemma 5 *For round t, define the non-linear instantaneous regret as $\dot{r}_t = g^{-1}(\langle x_t, \theta^* \rangle) - g^{-1}(\langle x^*, \theta^* \rangle)$ and the linear instantaneous regret as $r_t = \langle x_t, \theta^* \rangle - \langle x^*, \theta^* \rangle$. Under the assumption that $|\langle x, \theta^* \rangle| \leq 1$ and $g^{-1}(\cdot)$ is (locally) Lipschitz continuous with constant G then the following holds for all rounds $t = 1, \ldots, t$:*

$$\dot{r}_t \leq G r_t \tag{10.19}$$

which implies the non-linear regret is upper bounded by a constant times the linear regret.

Proof: The proof builds on [129]. Since $g^{-1}(\cdot)$ is (locally) Lipschitz this implies $\nabla g^{-1}(\cdot) \leq G$. Then for any $-1 \leq a \leq b \leq 1$ we have $g^{-1}(b) = g^{-1}(a) + \int_a^b \nabla g^{-1}(x) \mathrm{d}x$ therefore, $g^{-1}(b) - g^{-1}(a) \leq G(b-a)$. Now, since $g^{-1}(\cdot)$ is strictly increasing

$$x^* := \mathrm{argmin}_{x \in \mathcal{X}} \langle x, \theta^* \rangle = \mathrm{argmin}_{x \in \mathcal{X}} \, g^{-1}(\langle x, \theta^* \rangle) \ .$$

Since $-1 \leq \langle x^*, \theta^* \rangle \leq \langle x_t, \theta^* \rangle \leq 1$ then $g^{-1}(\langle x_t, \theta^* \rangle) - g^{-1}(\langle x^*, \theta^* \rangle) \leq G(\langle x_t, \theta^* \rangle - \langle x^*, \theta^* \rangle)$.

Using Lemma 5, if we can show a bound on the regret with linear loss functions then this implies a bound on the regret with non-linear loss functions. To bound the linear regret, we use the regret analysis established in [38]. Similarly as in Section 9.5, for Algorithm 10, as long as we have $\theta^* \in C_t$ over all rounds t, [38, Theorem 6] shows that $\sum_{t=1}^T r_t^2 \leq 8\beta^2 p \log T$. Then, to establish a problem independent regret bound we directly apply the Cauchy-Schwarz inequality to get

$$R_T = \sum_{t=1}^T r_t \leq \left(T \sum_{t=1}^T r_t^2 \right)^{1/2} \leq \beta \sqrt{8pT \log T} \ , \tag{10.20}$$

which holds conditioned on $\theta^* \in C_t$ over all rounds t. Moreover, for a problem dependent regret bound, we follow the proof of [38, Theorem 1] which shows

$$R_T = \sum_{t=1}^T r_t \leq \sum_{t=1}^T \frac{r_t^2}{\Delta} \leq \frac{8p\beta^2 \log T}{\Delta} \ , \tag{10.21}$$

which holds conditioned on $\theta^* \in C_t$ over all rounds t.

The focus of our analysis is then to choose a β such that the condition holds with high-probability uniformly over all rounds. From Algorithm 10, since $C_t := \{\theta : \|\theta - \hat{\theta}_t\|_{2,D_t} \leq \beta\}$ and we want to have $\theta^* \in C_t$, we focus on bounds for $\|\hat{\theta}_t - \theta^*\|_{2,D_t}$, the instantaneous estimation error. As in Section 9.5, deterministic bounds on the instantaneous estimation error can be obtained under two assumptions. First, we need to choose the regularization parameter λ_t such that

$$\lambda_t \geq 2R^* \left(\nabla_{\theta^*} \mathcal{L}(\theta^*)\right) \ . \tag{10.22}$$

Second, we need to have restricted strong convexity (RSC) for constant $\kappa > 0$

$$\inf_{\hat{\theta}_t - \theta^* \in E_{r,t}} \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla_{\theta^*} \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \ . \tag{10.23}$$

Under these assumptions, we have the following theorem (see Section D.1 for the proof) which generalizes Theorem 11 which was only shown to hold under linear regression.

Theorem 15 *Assume that $\hat{\theta}_t - \theta^* \in E_{r,t}$, the RSC condition is satisfied in the set $E_{r,t}$ with parameter κ, λ_t is suitably large, and $c > 0$ is a constant. Then for any norm $R(\cdot)$*

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq c\psi(E_{r,t}) \frac{\lambda_t}{\sqrt{\kappa}} \sqrt{t} \ . \tag{10.24}$$

Given the following observation presented in [10]

$$\nabla_{\theta^*} \mathcal{L}(\theta^*) = \frac{1}{t} \sum_{i=1}^{t} x_i \nabla_{\langle x_i, \theta^* \rangle} \varphi(\langle x_i, \theta^* \rangle) - y_i x_i = \frac{1}{t} \sum_{i=1}^{t} x_i (\mathbb{E}[y_i | x_i] - y_i) = \frac{1}{t} X_t^\top \omega_t$$

where from Section 10.2 we noted that the gradient of the log-partition function is equal to the expected response, i.e., $\nabla_{\langle x_i, \theta^* \rangle} \varphi(\langle x_i, \theta^* \rangle) = \mathbb{E}[y_i | x_i]$ and observing that $\eta_i = \mathbb{E}[y_i | x_i] - y_i$ is precisely the noise then $\omega_t = [\eta_1, \ldots, \eta_t]^\top$ is the noise vector. Therefore, we need to set $\lambda_t \geq 2R^*(\frac{1}{t} X_t^\top \omega_t)$ which is exactly the same quantity we have already shown how to bound in Theorem 12 therefore, we can use that theorem again.

For the RSC assumption, observe the following from [10].

$$\begin{aligned}
\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle = & \frac{1}{t} \sum_{i=1}^{t} \Big\{ \varphi(\langle x_i, \hat{\theta}_t \rangle) - y_i \langle x_i, \hat{\theta}_t \rangle \Big\} - \frac{1}{t} \sum_{i=1}^{t} \{ \varphi(\langle x_i, \theta^* \rangle) - y_i \langle x_i, \theta^* \rangle \} \\
& - \left\langle \frac{1}{t} \sum_{i=1}^{t} x_i \varphi'(\langle x_i, \theta^* \rangle) - y_i x_i, \hat{\theta}_t - \theta^* \right\rangle
\end{aligned}$$

where $\varphi'(\cdot)$ denotes the first derivative of $\varphi(\cdot)$. Simplifying and applying the mean value theorem twice we obtain

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla_{\theta^*}\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle = \frac{1}{t}\sum_{i=1}^{t} \varphi''(\langle x_i, \theta^* \rangle + \gamma_i \langle x_i, \hat{\theta}_t - \theta^* \rangle)\langle x_i, \hat{\theta}_t - \theta^* \rangle^2$$

where $\gamma \in [0, 1]$. Since the log-partition function is convex, its second derivative is non-negative and we will assume that it is bounded away from zero, i.e., there exists a constant ℓ such that $\varphi''(\cdot) \geq \ell > 0$. Therefore,

$$\frac{1}{t}\sum_{i=1}^{t} \varphi''(\langle x_i, \theta^* \rangle + \gamma_i \langle x_i, \hat{\theta}_t - \theta^* \rangle)\langle x_i, \hat{\theta}_t - \theta^* \rangle^2 \geq \frac{\ell}{t}\sum_{i=1}^{t} \langle x_i, \hat{\theta}_t - \theta^* \rangle^2 = \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

which shows

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla_{\theta^*}\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \geq \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2$$

Given this, and Assumption 5 the following inequality holds with constant $\kappa > 0$.

$$\inf_{\hat{\theta}_t - \theta^* \in E_{r,t}} \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \geq \kappa \|\hat{\theta}_t - \theta^*\|_2^2 \ .$$

For a bound on the instantaneous ellipsoidal estimation error, we plug in the value of λ_t from (9.27) into (10.24) and use the norm compatibility constant of the largest restricted error set to obtain

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq C\psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2} \right)$$

where $C = c_0 c/\sqrt{\kappa}$ is a constant which holds with high-probability across all rounds $t = 1, \ldots, T$. Therefore, if we set

$$\beta = C\psi(E_{r,\max}) \left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2} \right) \tag{10.25}$$

the confidence ellipsoid C_t will contain θ^* across all rounds with high-probability. Substituting our β into the regret bounds in (10.20) and (10.21) gives our main result in Theorem 13 and Theorem 14.

Chapter 11

Conclusions

In this thesis, we considered problems which require interaction between an algorithm and environment, for example, recommender systems, medical treatments, online advertising, and algorithmic trading. We designed and analyze structured online learning algorithms for both full and bandit information feedback for such problems.

In the first half of this thesis, we considered full information settings and general resource allocation problems with a particular focus on algorithmic trading. We made the following contributions. In Chapter 5, we considered the cost of updating one's portfolio and presented an efficient algorithm (OLU) which performs lazy updates to the portfolio. We analyzed the algorithm and showed for general convex and strongly convex functions its fixed regret is bounded as $O(\sqrt{T})$ and $O(\log T)$ respectively. Further, for general convex functions we showed the shifting regret is bounded as $O(\sqrt{T})$. We experimented with our algorithm using real-world stock market data and showed it earns more wealth than existing algorithm even with transaction costs.

In Chapter 6, we considered settings where one invests in groups of assets such as market sectors. We presented an efficient algorithm (OLU-GS) which uses group information and invests in top performing groups. We analyzed the algorithm and showed with group information the fixed regret for general convex and strongly convex functions is bounded as $O(\sqrt{T})$ and $O(\log T)$ respectively. Further, for general convex functions we showed the shifting regret is bounded as $O(\sqrt{T})$. We experimented with our algorithm using real-world stock market data and showed it earns more wealth than our previous algorithm OLU and also existing algorithm even with transaction costs.

In Chapter 7, we were interested in controlling the risk of our portfolios. We presented an efficient algorithm (ORASD) which computes diversified portfolios using a correlation network structure and the novel $L_{(\infty,1)}$ group norm as a constraint to force diversification. We established regret bounds of the form $O(\sqrt{T})$ and experimented with real-world stock market data. We showed the algorithm earns more wealth than our previous two algorithms and existing algorithms while incurring less risk.

In Chapter 8, we relaxed standard assumptions to allow borrowing wealth and shares of stock from the bank. We presented an algorithm (SHERAL) which learns to borrow wealth and shares of stock and uses a correlation network structure to compute hedged portfolios to reduce risk and take advantage of market crashes. We established regret bounds of the form $O(\sqrt{T})$ and experimented with real-world stock market data. We showed the algorithm earns orders of magnitude more wealth than our previous algorithms and existing algorithms and can effectively control various measures of risk.

In the second half of this thesis, we considered bandit information settings We made the following theoretical contributions. In Chapter 9, we considered the stochastic linear bandit problem under structural assumptions on the unknown parameter. We presented an algorithm which constructs tighter confidence ellipsoids which contain the unknown parameter with high-probability across all rounds. We established sharper theoretical regret bounds than what exists in the literature for any norm structure.

In Chapter 10, we considered the stochastic linear bandit problem under relaxed assumptions to allow the expected loss to be a non-linear function from Generalized Linear Models. We presented an algorithm which constructs tight confidence ellipsoids which contain the unknown parameter with high-probability across all rounds. We established theoretical regret bounds which match the regret bounds for linear loss functions even when considering a wider class of non-linear loss functions.

This thesis considered structured online learning algorithms to overcome challenges involved in interactive machine learning problems. It presented several experimental results which illustrated the algorithms' effectiveness and helped close the gap between theory and practice. Furthermore, it established several theoretical results which show sharper regret bounds are achievable under structural assumptions on the underlying model. Such results suggest the use of structure online learning algorithms in interactive learning problems is useful both in theory and practice.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Neural Information Processing Systems*, 2011.

[2] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[3] Alekh Agarwal, Sahand N. Negahban, and Martin J. Wainwright. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. In *Neural Information Processing Systems*, 2012.

[4] Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. Algorithms for portfolio management based on the newton method. In *International Conference on Machine Learning*, 2006.

[5] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[6] Mohamed El Hedi Arouri and Duc Khuong Nguyen. Oil prices, stock markets and portfolio investment: Evidence from sector analysis in europe over the last decade. *Energy Policy*, 38(8):4528–4539, 2010.

[7] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(3):397–422, 2003.

[8] Maria-Florina Balcan, Avrim Blum, Jason D. Hartline, and Yishay Mansour. Mechanism design via machine learning. In *Symposium on Foundations of Computer Science*, 2005.

[9] Arindam Banerjee, Sheng Chen, Farideh Fazayeli, and Vidyashankar Sivakumar. Estimation with norm regularization. In *Neural Information Processing Systems*, 2014.

[10] Arindam Banerjee, Sheng Chen, Farideh Fazayeli, and Vidyashankar Sivakumar. Estimation with norm regularization. Extended version at arXiv:1505.02294, 2015.

[11] Ole Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley, 1978.

[12] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[13] Peter J. Bickel, Yaacov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[14] Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. In *Conference on Learning Theory*, 1997.

[15] Allan Borodin, Ran El-Yaniv, and Vincent Gogan. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 21(1):579–594, 2004.

[16] Leon Bottou. Stochastic gradient learning in neural networks. In *Neuro-Nîmes*, 1991.

[17] Leon Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics*, 2010.

[18] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

[19] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[20] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[21] Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[22] George W. Brown. Iterative solutions of games by fictitious play. In *Activity Analysis of Production and Allocation*. Wiley, 1951.

[23] Lawrence D. Brown. Fundamentals of statistical exponential families with applications in statistical decision theory. *Lecture Notes-Monograph Series*, 9:i–279, 1986.

[24] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[25] Emmanuel Candes and Terence Tao. The dantzig selector : statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[26] Alexandra Carpentier and Remi Munos. Bandit theory meets compressed sensing for high-dimensional stochastic linear bandit. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[27] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[28] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[29] Antonin Chambolle, Ronald A. DeVore, Nam yong Lee, and Bradley J. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *Transactions on Image Processing*, 7(3):319–335, 1998.

[30] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[31] Sheng Chen and Arindam Banerjee. Structured estimation with atomic norms: General bounds and applications. In *Neural Information Processing Systems*, 2015.

[32] Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[33] Thomas M. Cover. Log optimal portfolios. In *Gambling Research: Gambling and Risk Taking*. University of Nevada-Reno, 1987.

[34] Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

[35] Thomas M. Cover and Erik Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996.

[36] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[37] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. The price of bandit information for online optimization. In *Neural Information Processing Systems*, 2007.

[38] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, 2008.

[39] Puja Das and Arindam Banerjee. Meta optimization and its application to portfolio selection. In *Conference on Knowledge Discovery and Data Mining*, 2011.

[40] Puja Das and Arindam Banerjee. Online quadratically constrained convex optimization with applications to risk adjusted portfolio selection. Technical report, University of Minnesota, 2012.

[41] Puja Das, Nicholas Johnson, and Arindam Banerjee. Online lazy updates for portfolio selection with transaction costs. In *Association for the Advancement of Artificial Intelligence*, 2013.

[42] Puja Das, Nicholas Johnson, and Arindam Banerjee. Online portfolio selection with group sparsity. In *Association for the Advancement of Artificial Intelligence*, 2014.

[43] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[44] Mark H.A. Davis and A. R. Norman. Portfolio selection with transaction costs. *Mathematics of Operations Research*, 15(4):676–713, 1990.

[45] Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *Conference on Electronic Commerce*, 2009.

[46] Dmitri A. Dolgov and Edmund H. Durfee. Resource allocation among agents with mdp-induced preferences. *Journal of Artificial Intelligence Research*, 27:505–549, 2006.

[47] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, 2008.

[48] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Conference on Learning Theory*, 2010.

[49] Miroslav Dudík, Daniel J. Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

[50] Stefano Ermon, Jon Conrad, Carla Gomes, and Bart Selman. Risk-sensitive policies for sustainable renewable resource allocation. In *International Joint Conference on Artificial Intelligence*, 2011.

[51] Jerry Fails and Dan Olsen. Interactive machine learning. In *International Conference on Intelligent User Interfaces*, 2003.

[52] Rebecca Fiebrink, Perry R. Cook, and Dan Trueman. Human model evaluation in interactive supervised learning. In *Conference on Human Factors in Computing Systems*, 2011.

[53] Mario A. T. Figueiredo and Robert D. Nowak. An em algorithm for wavelet-based image restoration. *Transactions on Image Processing*, 12(8):906–916, 2003.

[54] Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Neural Information Processing Systems*, 2010.

[55] Dean P. Foster. Prediction in the worst case. *The Annals of Statistics*, 19(2):1084–1090, 1991.

[56] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Conference on Computational Learning Theory*, 1996.

[57] Yoav Freund and Robert 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.

[58] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[59] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. Preprint at arXiv:1001.0736, 2010.

[60] Drew Fudenberg and David Levine. An easier way to calibrate. *Games and Economic Behavior*, 29(1-2):131–137, 1999.

[61] Sebastien Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. In *Conference on Learning Theory*, 2011.

[62] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Symposium on Discrete Algorithms*, 2008.

[63] Daniel Golovin, Andreas Krause, Beth Gardner, Sarah J. Converse, and Steve Morey. Dynamic resource allocation in conservation planning. In *Association for the Advancement of Artificial Intelligence*, 2011.

[64] Adam J. Grove, Nick Littlestone, and Dale Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

[65] Laszlo Gyorfi, Gyorgy Ottucsak, and Harro Walk. *Machine Learning for Financial Engineering*. Imperial College Press, 2011.

[66] James Hannan. Approximation to bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[67] Elad Hazan. Introduction to online convex optimization. Technical report, Princeton University, 2015.

[68] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine Learning*, 80(2-3):165–188, 2010.

[69] David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[70] Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

[71] Joseph W. Horwood. Risk-sensitive optimal harvesting and control of biological populations. *Mathematical Medicine and Biology*, 13:35–71, 1996.

[72] Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

[73] Stéphane Ivanoff, Franck Picard, and Vincent Rivoirard. Adaptive lasso and group-lasso for functional poisson regression. *Journal of Machine Learning Research*, 17(1):1903–1948, 2016.

[74] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *International Conference on Machine Learning*, 2009.

[75] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *International Conference on Machine Learning*, 2010.

[76] Nicholas Johnson and Arindam Banerjee. Online resource allocation with structured diversification. In *SIAM International Conference on Data Mining*, 2015.

[77] Nicholas Johnson and Arindam Banerjee. Structured hedging for resource allocations with leverage. In *Conference on Knowledge Discovery and Data Mining*, 2015.

[78] Nicholas Johnson, Vidyashankar Sivakumar, and Arindam Banerjee. Structured stochastic linear bandits. Preprint at arXiv:1606.05693, 2016.

[79] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[80] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[81] John L. Kelly. A new interpretation of information rate. *Bell Systems Technical Journal*, 35(4):917–926, 1956.

[82] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[83] Jyrki Kivinen and Manfred K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[84] Tze L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[85] Hoang Le, Andrew Kang, Yisong Yue, and Peter Carr. Smooth imitation learning for online sequence prediction. In *International Conference on Machine Learning*, 2016.

[86] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l1 regularized logistic regression. In *Association for the Advancement of Artificial Intelligence*, 2006.

[87] Bin Li and Steven C.H. Hoi. On-line portfolio selection with moving average reversion. In *International Conference of Machine Learning*, 2012.

[88] Bin Li and Steven C.H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys*, 46(3):1–35, 2014.

[89] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased ofine evaluation of contextual bandit algorithms with generalized linear models. *Journal of Machine Learning Research*, 26:19–36, 2012.

[90] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International World Wide Web Conference*, 2010.

[91] Qing Li, Maria Vassalou, and Yuhang Xing. Sector investment growth rates and the cross section of equity returns. *The Journal of Business*, 79(3):1637–1665, 2006.

[92] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[93] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[94] Peter Bühlmann Lukas Meier, Sara van de Geer. The group lasso for logistic regression. *Journal of the Royal Statistical Society*, 70(1):53–71, 2008.

[95] Michael J. P. Magill and George M. Constantinides. Portfolio selection with transaction costs. *Journal of Economic Theory*, 13(2):245–263, 1976.

[96] Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Network flow algorithms for structured sparsity. In *Neural Information Processing Systems*, 2010.

[97] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[98] Shahar Mendelson, Alain Pajor, and Nicole Tomczak-Jaegermann. Reconstruction and subgaussian operators in asymptotic geometric analysis. *Geometric and Functional Analysis*, 17:1248–1282, 2007.

[99] Rémi Munos. Efficient resources allocation for markov decision processes. In *Neural Information Processing Systems*, 2001.

[100] Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[101] Sahand N. Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

[102] John A. Nelder and Robert W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society*, 135(3):370–384, 1972.

[103] Arkadi S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

[104] Erik Ordentlich and Thomas M. Cover. Online portfolio selection. In *Conference on Learning Theory*, 1996.

[105] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[106] Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. An efficient projection for $\ell_{1,\infty}$ regularization. In *International Conference on Machine Learning*, 2009.

[107] Frank Rosenblatt. The perceptron–a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[108] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[109] Patt Rusmevichientong and John N. Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[110] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[111] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Conference on Learning Theory*, 2009.

[112] William F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442, 1964.

[113] William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

[114] Frank A. Sortino and Rovert van der Meer. Downside risk. *Journal of Portfolio Management*, 17(4):27–31, 1991.

[115] Michel Talagrand. *The Generic Chaining*. Springer Monographs in Mathematics, 2005.

[116] Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes*. Springer-Verlag, 2014.

[117] Andrea Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008.

[118] William 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.

[119] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[120] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society*, 67:91–108, 2005.

[121] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, pages 210–268. Cambridge University Press, 2012.

[122] John von Neumann. Zur theorie der gesellschaftsspiele. In *Mathematische Annalen*, volume 100, pages 295–320, 1928.

[123] Vladimir Vovk. A game of prediction with expert advice. In *Conference on Computational Learning Theory*, 1995.

[124] Huahua Wang and Arindam Banerjee. Online alternating direction method. In *International Conference on Machine Learning*, 2012.

[125] Manfred K. Warmuth, Rätsch Gunnar, Michael Mathieson, Jun Liao, and Christian Lemmen. Active learning in the drug discovery process. In *Neural Information Processing Systems*, 2001.

[126] Shan Xue, Alan Fern, and Daniel Sheldon. Dynamic resource allocation for optimizing population diffusion. In *International Conference on Artificial Intelligence and Statistics*, 2014.

[127] Yaoliang Yu. Better approximation and faster algorithm using the proximal average. In *Neural Information Processing Systems*, 2013.

[128] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68(1):49–67, 2006.

[129] Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi hua Zhou. Online stochastic linear optimization under one-bit feedback. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[130] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, 2003.

Appendix A

Online Lazy Updates

A.1 Regret Analysis

In this section, we present the proofs of the lemmas and theorems from Section 5.4.

Lemma 1 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (5.12), with potentially time-varying* η_t, γ_t*. Let* $d_\psi(\cdot,\cdot)$ *be* λ*-strongly convex with respect to norm* $||\cdot||$*, i.e.,* $d_\psi(\mathbf{p},\hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ *and let* $||\mathbf{p}-\hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p},\hat{\mathbf{p}} \in \mathcal{P}$*. Then, for any* $\mathbf{p}^* \in \mathcal{P}$*,*

$$\begin{aligned} &\eta_t[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \\ &\leq d_\psi(\mathbf{p}^*,\mathbf{p}_t) - d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1}) + \eta_t\gamma_t L + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ . \end{aligned} \tag{A.1}$$

Proof: Let $h_{\mathbf{p}_t}(\mathbf{p}) = \|\mathbf{p}-\mathbf{p}_t\|_1$ and let $g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) \in \partial h_{\mathbf{p}_t}(\mathbf{p}_{t+1})$. Then, for any $\mathbf{p} \in \mathcal{P}$, the optimality condition for (5.12) can be written as

$$\langle \mathbf{p}-\mathbf{p}_{t+1}, \eta_t\nabla\phi_t(\mathbf{p}_t) + \eta_t\gamma_t g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) + \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle \geq 0 \ . \tag{A.2}$$

Further, by convexity we have

$$\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) \leq \langle \mathbf{p}_t-\mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle \ , \tag{A.3}$$

$$\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \|\mathbf{p}^*-\mathbf{p}_t\|_1 \leq \langle \mathbf{p}_{t+1}-\mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \ . \tag{A.4}$$

Hence, for any $\mathbf{p}^* \in \mathcal{P}$

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}^*)+\gamma_t\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \gamma_t\|\mathbf{p}^*-\mathbf{p}_t\|_1] \\
&\quad \leq \eta_t\langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\gamma_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad = \eta_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\gamma_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_t) - \nabla\psi(\mathbf{p}_{t+1}) - \eta_t\nabla\phi_t(\mathbf{p}_t) - \eta_t\gamma_t g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad + \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \ .
\end{aligned}$$

The first term of the last equation is non-positive from (A.2). Thus we have,

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*)+\gamma_t\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \gamma_t\|\mathbf{p}^*-\mathbf{p}_t\|_1] \\
&\quad \leq \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\left\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\right\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\left\langle \sqrt{\frac{\lambda}{\eta_t}}(\mathbf{p}_t - \mathbf{p}_{t+1}), \sqrt{\frac{\eta_t}{\lambda}}\nabla\phi_t(\mathbf{p}_t)\right\rangle \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\lambda}{2}||\mathbf{p}_t - \mathbf{p}_{t+1}||^2 + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ ,
\end{aligned}$$

where in the second to last inequality follows from the Fenchel-Young inequality and from using $d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) \geq \frac{\lambda}{2}\|\mathbf{p}_{t+1} - \mathbf{p}_t\|^2$. Rearranging terms and using $\|\mathbf{p}^* - \mathbf{p}_t\|_1 \leq L$ ends the proof. ∎

Theorem 1 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (5.12). Let ϕ_t be a Lipschitz continuous function for which $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$. Then, by choosing $\eta_t = \eta = \frac{c_1}{\sqrt{T}}$ and $\gamma_t = \frac{c_2}{\sqrt{t}}$ for $c_1, c_2 > 0$, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \leq O(\sqrt{T}) \ . \tag{A.5}$$

Proof: By Lemma 1, noting $\eta_t = \eta$, we have

$$\begin{aligned}
\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] &\leq \sum_{t=1}^{T}\{d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta\gamma_t L\} + \frac{\eta^2 G^2 T}{2\lambda} \\
&= d_\psi(\mathbf{p}^*, \mathbf{p}_1) - d_\psi(\mathbf{p}^*, \mathbf{p}_{T+1}) + \sum_{t=1}^{T}\eta\gamma_t L + \frac{\eta^2 G^2 T}{2\lambda} \ .
\end{aligned}$$

Noting that Bregman divergences are always non-negative, dropping the $\gamma_t\|\mathbf{p}_{T+1}-\mathbf{p}_T\|_1$ term, dividing by η_t, and using $\eta = \frac{c_1}{\sqrt{T}}, \gamma_t = \frac{c_2}{\sqrt{t}}$ where $c_1, c_2, L > 0$ are constants, we have

$$\begin{aligned}\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*)] + \gamma_t \sum_{t=1}^{T-1} \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 &\leq \frac{1}{\eta} d_\psi(\mathbf{p}^*, \mathbf{p}_1) + L\sum_{t=1}^{T}\gamma_t + \frac{\eta G^2 T}{2\lambda} \\ &\leq \sqrt{T} d_\psi(\mathbf{p}^*, \mathbf{p}_1)/c_1 + 2c_2 L\sqrt{T} + \frac{c_1 G^2\sqrt{T}}{2\lambda} \\ &\leq O(\sqrt{T}) \ .\end{aligned}$$

■

Lemma 2 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (5.12) with potentially time-varying η_t and fixed γ. Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||_2$, i.e. $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p}-\hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. Assuming ϕ_t are all β-strongly convex, for any $\gamma < \frac{\beta}{4}$ and any $\mathbf{p}^* \in \mathcal{P}$, we have*

$$\begin{aligned}&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \gamma\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1] \\ &\quad\leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 - \eta_t\left(\frac{\beta}{2} - 2\gamma\right)\|\mathbf{p}^* - \mathbf{p}_t\|^2 \ .\end{aligned} \tag{A.6}$$

Proof: Let $h_{\mathbf{p}_t}(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_t\|_1$ and let $g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) \in \partial h_{\mathbf{p}_t}(\mathbf{p}_{t+1})$. Then, for any $\mathbf{p} \in \mathcal{P}$, the optimality condition for (5.12) can be written as

$$\langle \mathbf{p} - \mathbf{p}_{t+1}, \eta_t\nabla\phi_t(\mathbf{p}_t) + \eta_t\gamma g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) + \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle \geq 0 \ .$$

Further, by convexity we have

$$\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \frac{\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2 \leq \langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle \ , \tag{A.7}$$

$$\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \|\mathbf{p}^* - \mathbf{p}_t\|_1 \leq \langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \ . \tag{A.8}$$

Hence, for any $\mathbf{p}^* \in \mathcal{P}$

$$\begin{aligned}&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \gamma\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \gamma\|\mathbf{p}^* - \mathbf{p}_t\|_1 + \frac{\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2] \\ &\quad\leq \eta_t\langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\gamma\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\ &\quad= \eta_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\gamma\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\ &\quad= \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_t) - \nabla\psi(\mathbf{p}_{t+1}) - \eta_t\nabla\phi_t(\mathbf{p}_t) - \eta_t\gamma g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\ &\qquad+ \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \ .\end{aligned}$$

The first term of the last equation is non-positive. Thus we have,

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}^*)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\gamma\|\mathbf{p}^*-\mathbf{p}_t\|_1+\frac{\beta}{2}\|\mathbf{p}^*-\mathbf{p}_t\|^2]\\
&\quad\le\langle\mathbf{p}^*-\mathbf{p}_{t+1},\nabla\psi(\mathbf{p}_{t+1})-\nabla\psi(\mathbf{p}_t)\rangle+\eta_t\langle\mathbf{p}_t-\mathbf{p}_{t+1},\nabla\phi_t(\mathbf{p}_t)\rangle\\
&\quad=d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}_{t+1},\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\eta_t\langle\mathbf{p}_t-\mathbf{p}_{t+1},\nabla\phi_t(\mathbf{p}_t)\rangle\\
&\quad=d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}_{t+1},\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\eta_t\left\langle\sqrt{\frac{\lambda}{\eta_t}}(\mathbf{p}_t-\mathbf{p}_{t+1}),\sqrt{\frac{\eta_t}{\lambda}}\nabla\phi_t(\mathbf{p}_t)\right\rangle\\
&\quad\le d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}_{t+1},\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\frac{\lambda}{2}||\mathbf{p}_t-\mathbf{p}_{t+1}||^2+\frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2\\
&\quad\le d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2\ ,
\end{aligned}$$

where the second to last inequality follows from the Fenchel-Young inequality and from using $d_\psi(\mathbf{p}_{t+1},\mathbf{p}_t)\ge\frac{\lambda}{2}\|\mathbf{p}_{t+1}-\mathbf{p}_t\|^2$. Rearranging terms, we have

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}^*)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1]\\
&\quad\le d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2+\eta_t\gamma\|\mathbf{p}^*-\mathbf{p}_t\|_1-\frac{\eta_t\beta}{2}\|\mathbf{p}^*-\mathbf{p}_t\|^2\ .
\end{aligned}$$

Now, using the fact that for any $u\in\mathbb{R}$, $|u|\le 2u^2$, we have

$$\|\mathbf{p}^*-\mathbf{p}_t\|_1=\sum_{i=1}^n|p^*(i)-p_t(i)|\le 2\sum_{i=1}^n(p^*(i)-p_t(i))^2=2\|\mathbf{p}^*-\mathbf{p}_t\|^2.$$

Hence, for any $\gamma<\beta/4$, we have $\gamma\|\mathbf{p}^*-\mathbf{p}_t\|_1<\frac{\beta}{2}\|\mathbf{p}^*-\mathbf{p}_t\|^2$, implying

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}^*)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1]\\
&\quad\le d_\psi(\mathbf{p}^*,\mathbf{p}_t)-d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2-\eta_t\left(\frac{\beta}{2}-2\gamma\right)\|\mathbf{p}^*-\mathbf{p}_t\|^2\ .
\end{aligned}$$

■

Theorem 2 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (5.12). Let ϕ_t be all β-strongly convex and $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2\le G$. Then, for any $\gamma<\beta/4$, choosing $\eta_t=\frac{1}{\kappa t}$ where $\kappa\in(0,\beta-4\gamma]$, and with $d_\psi(\mathbf{p},\mathbf{p}')=\frac{1}{2}\|\mathbf{p}-\mathbf{p}'\|^2$, we have*

$$\sum_{t=1}^T[\phi_t(\mathbf{p}_t)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\phi_t(\mathbf{p}^*)]\le O(\log T)\ . \tag{A.9}$$

Proof: By Lemma 2, we have

$$\begin{aligned}
&\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\phi_t(\mathbf{p}^*)]\\
&\le\sum_{t=1}^{T}\left[\frac{1}{\eta_t}d_\psi(\mathbf{p}^*,\mathbf{p}_t)-\frac{1}{\eta_t}d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})-\left(\frac{\beta}{2}-2\gamma\right)\|\mathbf{p}^*-\mathbf{p}_t\|^2+\frac{\eta_t}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2\right]\\
&\le\frac{1}{\eta_1}d_\psi(\mathbf{p}^*,\mathbf{p}_1)-\frac{1}{\eta_T}d_\psi(\mathbf{p}^*,\mathbf{p}_{T+1})\\
&+\sum_{t=1}^{T-1}\left[d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)-\left(\frac{\beta}{2}-2\gamma\right)\|\mathbf{p}^*-\mathbf{p}_{t+1}\|^2\right]+\frac{G^2}{2\lambda}\sum_{t=1}^{T}\eta_t\\
&\le\frac{1}{\eta_1}d_\psi(\mathbf{p}^*,\mathbf{p}_1)-\sum_{t=1}^{T-1}\left(\frac{\beta}{2}-2\gamma-\frac{\kappa}{2}\right)\|\mathbf{p}^*-\mathbf{p}_{t+1}\|^2+\frac{cG^2}{2\lambda\kappa}\log T\ ,
\end{aligned}$$

where the last inequality follows from

$$\sum_{t=1}^{T-1}d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)=\frac{\kappa}{2}\|\mathbf{p}^*-\mathbf{p}_{t+1}\|^2\ ,$$

for $d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})=\frac{1}{2}\|\mathbf{p}^*-\mathbf{p}_{t+1}\|^2$, and $\sum_{t=1}^{T}\frac{1}{t}\le c\log T$. Now, since $(\frac{\beta}{2}-2\gamma-\frac{\kappa}{2})\ge 0$

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t)+\gamma\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\phi_t(\mathbf{p}^*)]\le\kappa d_\psi(\mathbf{p}^*,\mathbf{p}_1)+\frac{cG^2}{\lambda\kappa}\log T=O(\log T)\ .$$

■

Theorem 3 *Let $\{\mathbf{p}_1^*,\cdots,\mathbf{p}_T^*\}$ be any sequence of portfolios serving as a comparator in (5.12). Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p},\hat{\mathbf{p}})\ge\frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p}-\hat{\mathbf{p}}||_1\le L,\forall\mathbf{p},\hat{\mathbf{p}}\in\mathcal{P}$. For $\eta_t=\eta=\frac{c_1}{\sqrt{T}}$ and $\gamma_t=\frac{c_2}{\sqrt{t}}$ for $c_1,c_2>0$, $\frac{1}{r}+\frac{1}{q}=1$, and $||\nabla\psi(\mathbf{p}_t)||_r\le\zeta$, we have*

$$\begin{aligned}
&\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}_t^*)]+\sum_{t=1}^{T}[\gamma_t\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\gamma_t\|\mathbf{p}_{t+1}^*-\mathbf{p}_t^*\|_1]\\
&\le O(\sqrt{T})+\frac{\sqrt{T}}{c_1}\left\{d_\psi(\mathbf{p}_1^*,\mathbf{p}_1)-d_\psi(\mathbf{p}_{T+1}^*,\mathbf{p}_{T+1})+\psi(\mathbf{p}_{T+1}^*)-\psi(\mathbf{p}_1^*)+\zeta\sum_{t=1}^{T}\|\mathbf{p}_t^*-\mathbf{p}_{t+1}^*\|_q\right\}.
\end{aligned}\tag{A.10}$$

Proof: From Lemma 1 we have the following by substituting $\mathbf{p}^* = \mathbf{p}_t^*$

$$\begin{aligned} &\eta[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}_t^*)] \\ &\leq d_\psi(\mathbf{p}_t^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_t^*, \mathbf{p}_{t+1}) + \eta\gamma_t L + \frac{\eta^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ . \end{aligned} \tag{A.11}$$

This does not telescope, so we need to add additional terms. To this end, consider

$$\begin{aligned} d_\psi(\mathbf{p}_t^*, \mathbf{p}_{t+1}) - d_\psi(\mathbf{p}_{t+1}^*, \mathbf{p}_{t+1}) &= \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) + \langle\nabla\psi(\mathbf{p}_{t+1}), \mathbf{p}_{t+1}^* - \mathbf{p}_t^*\rangle \\ &\geq \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) - \zeta||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q \end{aligned} \tag{A.12}$$

where the last inequality follows from Holder's inequality, i.e.,

$$\langle\mathbf{p}_{t+1}^* - \mathbf{p}_t^*, \nabla\psi(\mathbf{p}_{t+1})\rangle \geq -\|\nabla\psi(\mathbf{p}_{t+1})\|_r\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ \geq -\zeta\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ .$$

Adding (A.12) to (A.11), and further adding $-\eta\gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1$ on both sides, we have for $\|\nabla\phi_t(\mathbf{p}_t)\|^2 \leq G$

$$\begin{aligned} &\eta[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}_t^*) - \gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] + \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) - \zeta\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \\ &\leq d_\psi(\mathbf{p}_t^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}^*, \mathbf{p}_{t+1}) - \eta\gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1 + \eta\gamma_t L + \frac{\eta^2}{2\lambda}G^2 \ . \end{aligned} \tag{A.13}$$

Summing over T rounds, we have

$$\begin{aligned} &\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}_t^*) - \gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] + \psi(\mathbf{p}_1^*) - \psi(\mathbf{p}_{T+1}^*) - \zeta\sum_{t=1}^{T}\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \\ &\leq d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \eta L\sum_{t=1}^{T}\gamma_t + \frac{\eta^2}{2\lambda}\sum_{t=1}^{T}G^2 \ . \end{aligned} \tag{A.14}$$

where we ignored some negative terms on the right hand side of the inequality. Rearranging, we have

$$\begin{aligned} &\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}_t^*)] + \eta\sum_{t=1}^{T}[\gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \\ &\leq d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) \\ &+ \zeta\sum_{t=1}^{T}||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q + \eta L\sum_{t=1}^{T}\gamma_t + \frac{\eta^2}{2\lambda}\sum_{t=1}^{T}G^2 \end{aligned} \tag{A.15}$$

Setting $\eta = \frac{c_1}{\sqrt{T}}$ and $\gamma_t = \frac{c_2}{\sqrt{t}}$ and dividing both sides by η gives

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}_t^*)] + \sum_{t=1}^{T}[\gamma_t\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \gamma_t\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \tag{A.16}$$

$$\leq c_2 L\sqrt{T} + \frac{c_1\sqrt{T}G^2}{2\lambda}$$

$$+ \frac{\sqrt{T}}{c_1}\left\{d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) + \zeta\sum_{t=1}^{T}||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q\right\} .$$

■

Appendix B

Online Lazy Updates with Group Sparsity

B.1 Regret Analysis

In this section, we present the proofs of the lemmas and theorems from Section 6.4.

Lemma 3 *Let the sequence of* $\{\mathbf{p}_t\}$ *be defined by the update in (6.19), with potentially time-varying* η_t*. Let* $d_\psi(\cdot,\cdot)$ *be* λ*-strongly convex with respect to norm* $||\cdot||$*, i.e.,* $d_\psi(\mathbf{p},\hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ *and let* $||\mathbf{p}-\hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p},\hat{\mathbf{p}} \in \mathcal{P}$*. Then, for any* $\mathbf{p}^* \in \mathcal{P}$*,*

$$\begin{aligned}&\eta_t[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*) - \lambda_2 r(\mathbf{p}^*)] \\ &\leq d_\psi(\mathbf{p}^*,\mathbf{p}_t) - d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1}) + \eta_t\lambda_2 L + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ .\end{aligned} \tag{B.1}$$

Proof: Let $r'(\mathbf{p}) \in \partial r(\mathbf{p})$, $h_{\mathbf{p}_t}(\mathbf{p}) = \|\mathbf{p}-\mathbf{p}_t\|_1$ and let $g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) \in \partial h_{\mathbf{p}_t}(\mathbf{p}_{t+1})$. Then, for any $\mathbf{p} \in \mathcal{P}$, the optimality condition for (6.19) can be written as

$$\langle \mathbf{p}-\mathbf{p}_{t+1}, \eta_t\nabla\phi_t(\mathbf{p}_t) + \eta_t\lambda_1 r'(\mathbf{p}_{t+1}) + \eta_t\lambda_2 g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) + \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle \geq 0 \ . \tag{B.2}$$

Further, by convexity we have

$$\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) \leq \langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle \ , \tag{B.3}$$

$$r(\mathbf{p}_t) - r(\mathbf{p}^*) \leq \langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle \tag{B.4}$$

$$\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \|\mathbf{p}^*-\mathbf{p}_t\|_1 \leq \langle \mathbf{p}_{t+1}-\mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \ . \tag{B.5}$$

Hence, for any $\mathbf{p}^* \in \mathcal{P}$

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t)-\phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*)+\lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \lambda_2\|\mathbf{p}^*-\mathbf{p}_t\|_1] \\
&\quad \leq \eta_t\langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\lambda_1\langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle + \eta_t\lambda_2\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad = \eta_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\lambda_1\langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle + \eta_t\lambda_2\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_t) - \nabla\psi(\mathbf{p}_{t+1}) - \eta_t\nabla\phi_t(\mathbf{p}_t) - \eta_t\lambda_1 r'(\mathbf{p}_{t+1}) - \eta_t\lambda_2 g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad + \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \ .
\end{aligned}$$

The first term of the last equation is non-positive from (B.2). Thus we have,

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*)+\lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \lambda_2\|\mathbf{p}^*-\mathbf{p}_t\|_1] \\
&\quad \leq \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\left\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\right\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\left\langle \sqrt{\frac{\lambda}{\eta_t}}(\mathbf{p}_t - \mathbf{p}_{t+1}), \sqrt{\frac{\eta_t}{\lambda}}\nabla\phi_t(\mathbf{p}_t)\right\rangle \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\lambda}{2}||\mathbf{p}_t - \mathbf{p}_{t+1}||^2 + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ ,
\end{aligned}$$

where in the second to last inequality follows from the Fenchel-Young inequality and from using $d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) \geq \frac{\lambda}{2}\|\mathbf{p}_{t+1} - \mathbf{p}_t\|^2$. Rearranging terms and using $\|\mathbf{p}^* - \mathbf{p}_t\|_1 \leq L$ ends the proof. ∎

Theorem 4 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by (6.19). Let ϕ_t be a Lipschitz continuous function for which $\|\nabla\phi_t(\mathbf{p}_t)\|_2^2 \leq G$. Then, by choosing $\eta \propto \frac{1}{\sqrt{T}}$ and $\lambda_2 \propto \frac{1}{\sqrt{T}}$, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi(\mathbf{p}^*) - r(\mathbf{p}^*)] \leq O(\sqrt{T}) \ . \tag{B.6}$$

Proof: By Lemma 3, noting $\eta_t = \eta$, we have

$$\begin{aligned}
&\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*) - \lambda_1 r(\mathbf{p}^*)] \\
&\leq\sum_{t=1}^{T}\{d_\psi(\mathbf{p}^*,\mathbf{p}_t) - d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1})+\eta\lambda_2 L\} + \frac{\eta^2 G^2 T}{2\lambda} \\
&= d_\psi(\mathbf{p}^*,\mathbf{p}_1) - d_\psi(\mathbf{p}^*,\mathbf{p}_{T+1}) + \sum_{t=1}^{T}\eta\lambda_2 L + \frac{\eta^2 G^2 T}{2\lambda}\ .
\end{aligned}$$

Noting that Bregman divergences are always non-negative, dropping the $\lambda_2\|\mathbf{p}_{T+1}-\mathbf{p}_T\|_1$ term, dividing by η_t, and using $\eta = \frac{c_1}{\sqrt{T}}, \lambda_2 = \frac{1}{\sqrt{T}}$ where $c_1, L > 0$ are constants, we have

$$\begin{aligned}
&\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) - \lambda_1 r(\mathbf{p}^*)] + \lambda_2\sum_{t=1}^{T-1}\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1 \\
&\leq \frac{1}{\eta}d_\psi(\mathbf{p}^*,\mathbf{p}_1) + L\sum_{t=1}^{T}\lambda_2 + \frac{\eta G^2 T}{2\lambda} \\
&\leq \sqrt{T}d_\psi(\mathbf{p}^*,\mathbf{p}_1)/c_1 + 2L\sqrt{T} + \frac{c_1 G^2\sqrt{T}}{2\lambda} \\
&\leq O(\sqrt{T})\ .
\end{aligned}$$

■

Lemma 4 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by the update in (6.19) with potentially time-varying η_t and λ_2. Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||_2$, i.e. $d_\psi(\mathbf{p},\hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p}-\hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p}-\hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p},\hat{\mathbf{p}} \in \mathcal{P}$. Assuming ϕ_t are all β-strongly convex, for any $\lambda_2 < \frac{\beta}{4}$ and any $\mathbf{p}^* \in \mathcal{P}$, we have*

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1] \\
&\quad\leq d_\psi(\mathbf{p}^*,\mathbf{p}_t) - d_\psi(\mathbf{p}^*,\mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 - \eta_t\left(\frac{\beta}{2}-2\lambda_2\right)\|\mathbf{p}^*-\mathbf{p}_t\|^2\ .
\end{aligned} \tag{B.7}$$

Proof: Let $r'(\mathbf{p}) \in \partial r(\mathbf{p})$, $h_{\mathbf{p}_t}(\mathbf{p}) = \|\mathbf{p}-\mathbf{p}_t\|_1$ and let $g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) \in \partial h_{\mathbf{p}_t}(\mathbf{p}_{t+1})$. Then, for any $\mathbf{p} \in \mathcal{P}$, the optimality condition for (6.19) can be written as

$$\langle \mathbf{p}-\mathbf{p}_{t+1}, \eta_t\nabla\phi_t(\mathbf{p}_t) + \eta_t\lambda_1 r'(\mathbf{p}_{t+1}) + \eta_t\lambda_2 g_{\mathbf{p}_t}(\mathbf{p}_{t+1}) + \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle \geq 0\ .$$

Further, by convexity we have

$$\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \frac{\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2 \leq \langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle \ , \tag{B.8}$$

$$r(\mathbf{p}_t) - r(\mathbf{p}^*) \leq \langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle \tag{B.9}$$

$$\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \|\mathbf{p}^* - \mathbf{p}_t\|_1 \leq \langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \ . \tag{B.10}$$

Hence, for any $\mathbf{p}^* \in \mathcal{P}$

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \lambda_2\|\mathbf{p}^* - \mathbf{p}_t\|_1 + \frac{\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2] \\
&\quad \leq \eta_t\langle \mathbf{p}_t - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\lambda_1\langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle + \eta_t\lambda_2\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\quad = \eta_t\langle \mathbf{p}_{t+1} - \mathbf{p}^*, \nabla\phi_t(\mathbf{p}_t)\rangle + \eta_t\lambda_1\langle \mathbf{p}_{t+1} - \mathbf{p}^*, r'(\mathbf{p}_{t+1})\rangle \\
&\quad + \eta_t\lambda_2\langle \mathbf{p}_{t+1} - \mathbf{p}^*, g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_t) - \nabla\psi(\mathbf{p}_{t+1}) - \eta_t\nabla\phi_t(\mathbf{p}_t) - \eta_t\lambda_1 r'(\mathbf{p}_{t+1}) - \eta_t\lambda_2 g_{\mathbf{p}_t}(\mathbf{p}_{t+1})\rangle \\
&\qquad + \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \ .
\end{aligned}$$

The first term of the last equation is non-positive. Thus we have,

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \lambda_2\|\mathbf{p}^* - \mathbf{p}_t\|_1 + \frac{\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2] \\
&\quad \leq \langle \mathbf{p}^* - \mathbf{p}_{t+1}, \nabla\psi(\mathbf{p}_{t+1}) - \nabla\psi(\mathbf{p}_t)\rangle + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \nabla\phi_t(\mathbf{p}_t)\rangle \\
&\quad = d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \eta_t\left\langle \sqrt{\frac{\lambda}{\eta_t}}(\mathbf{p}_t - \mathbf{p}_{t+1}), \sqrt{\frac{\eta_t}{\lambda}}\nabla\phi_t(\mathbf{p}_t)\right\rangle \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\lambda}{2}||\mathbf{p}_t - \mathbf{p}_{t+1}||^2 + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ ,
\end{aligned}$$

where the second to last inequality follows from the Fenchel-Young inequality and from using $d_\psi(\mathbf{p}_{t+1}, \mathbf{p}_t) \geq \frac{\lambda}{2}\|\mathbf{p}_{t+1} - \mathbf{p}_t\|^2$. Rearranging terms, we have

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1] \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 + \eta_t\lambda_2\|\mathbf{p}^* - \mathbf{p}_t\|_1 - \frac{\eta_t\beta}{2}\|\mathbf{p}^* - \mathbf{p}_t\|^2 \ .
\end{aligned}$$

Now, using the fact that for any $u \in \mathbb{R}$, $|u| \leq 2u^2$, we have

$$\|\mathbf{p}^* - \mathbf{p}_t\|_1 = \sum_{i=1}^{n} |p^*(i) - p_t(i)| \leq 2 \sum_{i=1}^{n} (p^*(i) - p_t(i))^2 = 2\|\mathbf{p}^* - \mathbf{p}_t\|^2.$$

Hence, for any $\lambda_2 < \beta/4$, we have $\lambda_2 \|\mathbf{p}^* - \mathbf{p}_t\|_1 < \frac{\beta}{2} \|\mathbf{p}^* - \mathbf{p}_t\|^2$, implying

$$\begin{aligned}
&\eta_t[\phi_t(\mathbf{p}_t) - \phi_t(\mathbf{p}^*) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2 \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1] \\
&\quad \leq d_\psi(\mathbf{p}^*, \mathbf{p}_t) - d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) + \frac{\eta_t^2}{2\lambda} ||\nabla \phi_t(\mathbf{p}_t)||^2 - \eta_t \left(\frac{\beta}{2} - 2\lambda_2 \right) \|\mathbf{p}^* - \mathbf{p}_t\|^2 \ .
\end{aligned}$$

■

Theorem 5 *Let the sequence of $\{\mathbf{p}_t\}$ be defined by (6.19). Let ϕ_t be all β-strongly convex and $\|\nabla \phi_t(\mathbf{p}_t)\|_2^2 \leq G$. Then, for any $\lambda_2 < \beta/4$, choosing $\eta_t = \frac{1}{\kappa t}$ where $\kappa \in (0, \beta - \lambda_2]$ and with $d_\psi(\mathbf{p}, \mathbf{p}') = \frac{1}{2}\|\mathbf{p} - \mathbf{p}'\|_2^2$, we have*

$$\sum_{t=1}^{T} [\phi_t(\mathbf{p}_t) + r(\mathbf{p}_t) + \lambda_2 \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi(\mathbf{p}^*) - r(\mathbf{p}^*)] \leq O(\log T) \ . \tag{B.11}$$

Proof: By Lemma 4, we have

$$\begin{aligned}
&\sum_{t=1}^{T} [\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) - \lambda_1 r(\mathbf{p}^*) + \lambda_2 \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*)] \\
&\leq \sum_{t=1}^{T} \left[\frac{1}{\eta_t} d_\psi(\mathbf{p}^*, \mathbf{p}_t) - \frac{1}{\eta_t} d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) - \left(\frac{\beta}{2} - 2\lambda_2 \right) \|\mathbf{p}^* - \mathbf{p}_t\|^2 + \frac{\eta_t}{2\lambda} ||\nabla \phi_t(\mathbf{p}_t)||^2 \right] \\
&\leq \frac{1}{\eta_1} d_\psi(\mathbf{p}^*, \mathbf{p}_1) - \frac{1}{\eta_T} d_\psi(\mathbf{p}^*, \mathbf{p}_{T+1}) \\
&+ \sum_{t=1}^{T-1} \left[d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) - \left(\frac{\beta}{2} - 2\lambda_2 \right) \|\mathbf{p}^* - \mathbf{p}_{t+1}\|^2 \right] + \frac{G^2}{2\lambda} \sum_{t=1}^{T} \eta_t \\
&\leq \frac{1}{\eta_1} d_\psi(\mathbf{p}^*, \mathbf{p}_1) - \sum_{t=1}^{T-1} \left(\frac{\beta}{2} - 2\lambda_2 - \frac{\kappa}{2} \right) \|\mathbf{p}^* - \mathbf{p}_{t+1}\|^2 + \frac{cG^2}{2\lambda\kappa} \log T \ ,
\end{aligned}$$

where the last inequality follows from

$$\sum_{t=1}^{T-1} d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) = \frac{\kappa}{2} \|\mathbf{p}^* - \mathbf{p}_{t+1}\|^2 \ ,$$

for $d_\psi(\mathbf{p}^*, \mathbf{p}_{t+1}) = \frac{1}{2}\|\mathbf{p}^* - \mathbf{p}_{t+1}\|^2$, and $\sum_{t=1}^{T} \frac{1}{t} \leq c \log T$. Now, since $(\frac{\beta}{2} - 2\lambda_2 - \frac{\kappa}{2}) \geq 0$

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}^*) - \lambda_1 r(\mathbf{p}^*)]$$
$$\leq \kappa d_\psi(\mathbf{p}^*, \mathbf{p}_1) + \frac{cG^2}{\lambda\kappa}\log T = O(\log T) \ .$$

■

Theorem 6 *Let $\{\mathbf{p}_1^*, \cdots, \mathbf{p}_T^*\}$ be any sequence of portfolios serving as a comparator in (6.19). Let $d_\psi(\cdot,\cdot)$ be λ-strongly convex with respect to norm $||\cdot||$, i.e., $d_\psi(\mathbf{p}, \hat{\mathbf{p}}) \geq \frac{\lambda}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_2^2$ and $||\mathbf{p} - \hat{\mathbf{p}}||_1 \leq L, \forall \mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}$. For $\eta_t = \eta = \frac{c_1}{\sqrt{T}}$ and $\lambda_2 = \frac{c_2}{\sqrt{T}}$ for $c_1, c_2 > 0$, $\frac{1}{r} + \frac{1}{q} = 1$, and $||\nabla\psi(\mathbf{p}_t)||_r \leq \zeta$, we have*

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) - \phi_t(\mathbf{p}_t^*) - \lambda_1 r(\mathbf{p}^*)] + \sum_{t=1}^{T}[\lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \lambda_2\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \tag{B.12}$$
$$\leq O(\sqrt{T}) + \frac{\sqrt{T}}{c_1}\left\{d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) + \zeta\sum_{t=1}^{T}\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q\right\}.$$

Proof: From Lemma 3 we have the following by substituting $\mathbf{p}^* = \mathbf{p}_t^*$

$$\eta[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}_t^*) - \lambda_1 r(\mathbf{p}_t^*)] \tag{B.13}$$
$$\leq d_\psi(\mathbf{p}_t^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_t^*, \mathbf{p}_{t+1}) + \eta\lambda_2 L + \frac{\eta^2}{2\lambda}||\nabla\phi_t(\mathbf{p}_t)||^2 \ .$$

This does not telescope, so we need to add additional terms. To this end, consider

$$\begin{aligned} d_\psi(\mathbf{p}_t^*, \mathbf{p}_{t+1}) - d_\psi(\mathbf{p}_{t+1}^*, \mathbf{p}_{t+1}) &= \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) + \langle\nabla\psi(\mathbf{p}_{t+1}), \mathbf{p}_{t+1}^* - \mathbf{p}_t^*\rangle \\ &\geq \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) - \zeta||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q \end{aligned} \tag{B.14}$$

where the last inequality follows from Holder's inequality, i.e.,

$$\langle\mathbf{p}_{t+1}^* - \mathbf{p}_t^*, \nabla\psi(\mathbf{p}_{t+1})\rangle \geq -\|\nabla\psi(\mathbf{p}_{t+1})\|_r\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ \geq -\zeta\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q \ .$$

Adding (B.14) to (B.13), and further adding $-\eta\lambda_2\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1$ on both sides, we have for $\|\nabla\phi_t(\mathbf{p}_t)\|^2 \leq G$

$$\eta[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t) + \lambda_2\|\mathbf{p}_{t+1} - \mathbf{p}_t\|_1 - \phi_t(\mathbf{p}_t^*) - \lambda_1 r(\mathbf{p}_t^*) - \lambda_2\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1] \tag{B.15}$$
$$+ \psi(\mathbf{p}_t^*) - \psi(\mathbf{p}_{t+1}^*) - \zeta\|\mathbf{p}_t^* - \mathbf{p}_{t+1}^*\|_q$$
$$\leq d_\psi(\mathbf{p}_t^*, \mathbf{p}_t) - d_\psi(\mathbf{p}_{t+1}^*, \mathbf{p}_{t+1}) - \eta\lambda_2\|\mathbf{p}_{t+1}^* - \mathbf{p}_t^*\|_1 + \eta\lambda_2 L + \frac{\eta^2}{2\lambda}G^2 \ . \tag{B.16}$$

Summing over T rounds, we have

$$\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t) + \lambda_1 r(\mathbf{p}_t)+\lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\phi_t(\mathbf{p}_t^*) - \lambda_1 r(\mathbf{p}_t^*)-\lambda_2\|\mathbf{p}_{t+1}^*-\mathbf{p}_t^*\|_1] \tag{B.17}$$
$$+\psi(\mathbf{p}_1^*)-\psi(\mathbf{p}_{T+1}^*)-\zeta\sum_{t=1}^{T}\|\mathbf{p}_t^*-\mathbf{p}_{t+1}^*\|_q$$
$$\leq d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \eta L\sum_{t=1}^{T}\lambda_2 + \frac{\eta^2}{2\lambda}\sum_{t=1}^{T}G^2 \ .$$

where we ignored some negative terms on the right hand side of the inequality. Rearranging, we have

$$\eta\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t)+\lambda_1 r(\mathbf{p}_t)-\phi_t(\mathbf{p}_t^*)-\lambda_1 r(\mathbf{p}_t^*)]+\eta\sum_{t=1}^{T}[\lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\lambda_2\|\mathbf{p}_{t+1}^*-\mathbf{p}_t^*\|_1] \tag{B.18}$$
$$\leq d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*)$$
$$+ \zeta\sum_{t=1}^{T}||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q + \eta L\sum_{t=1}^{T}\lambda_2 + \frac{\eta^2}{2\lambda}\sum_{t=1}^{T}G^2$$

Setting $\eta = \frac{c_1}{\sqrt{T}}$ and $\lambda_2 = \frac{c_2}{\sqrt{T}}$ and dividing both sides by η gives

$$\sum_{t=1}^{T}[\phi_t(\mathbf{p}_t)+\lambda_1 r(\mathbf{p}_t)-\phi_t(\mathbf{p}_t^*)-\lambda_1 r(\mathbf{p}_t^*)]+\sum_{t=1}^{T}[\lambda_2\|\mathbf{p}_{t+1}-\mathbf{p}_t\|_1-\lambda_2\|\mathbf{p}_{t+1}^*-\mathbf{p}_t^*\|_1] \tag{B.19}$$
$$\leq c_2 L\sqrt{T} + \frac{c_1\sqrt{T}G^2}{2\lambda}$$
$$+ \frac{\sqrt{T}}{c_1}\left\{d_\psi(\mathbf{p}_1^*, \mathbf{p}_1) - d_\psi(\mathbf{p}_{T+1}^*, \mathbf{p}_{T+1}) + \psi(\mathbf{p}_{T+1}^*) - \psi(\mathbf{p}_1^*) + \zeta\sum_{t=1}^{T}||\mathbf{p}_t^* - \mathbf{p}_{t+1}^*||_q\right\} \ .$$

■

Appendix C

Structured Stochastic Linear Bandits

C.1 Definitions and Background

The following definitions and lemmas can be found in [9, 10, 121].

Definition 9 A random variable x is sub-Gaussian if the moments satisfies

$$[\mathbb{E}|x|^p]^{\frac{1}{p}} \leq K\sqrt{p} \tag{C.1}$$

for any $p \geq 1$ with constant K. The minimum value of K is called the sub-Gaussian norm of x and denoted by $\|\|x\|\|_{\psi_2}$.

Additionally, every sub-Gaussian random variable satisfies

$$P(|x| > t) \leq \exp\left(1 - c\frac{t^2}{\|x\|_{\psi_2}^2}\right) \tag{C.2}$$

for all $t \geq 0$.

Definition 10 A random vector $X \in \mathbb{R}^p$ is sub-Gaussian if the one-dimensional marginals $\langle X, x\rangle$ are sub-Gaussian random variables for all $x \in \mathbb{R}^p$. The sub-Gaussian norm of X is defined as

$$\|\|X\|\|_{\psi_2} = \sup_{x \in S^{p-1}} \|\langle X, x\rangle\|_{\psi_2} \tag{C.3}$$

Definition 11 For any set $A \in \mathbb{R}^p$, the Gaussian width of the set A is defined as

$$w(A) = \mathbb{E}\left[\sup_{u \in A} \langle g, u \rangle\right] \tag{C.4}$$

where the expectation is over $g \sim N(0, \mathbb{I}_{p \times p})$ which is a vector of independent zero-mean unit-variance Gaussian random variables.

Lemma 6 *For any bounded random variable $|X| \leq B$, then X is a sub-Gaussian random variable with $\|X\|_{\psi_2} \leq B$.*

Lemma 7 *Consider a sub-Gaussian random vector X with sub-Gaussian norm $K = \max_i \|\|X_i\|\|_{\psi_2}$, then, for vector a, $Z = \langle X, a \rangle$ is a sub-Gaussian random variable with sub-Gaussian norm $\|\|Z\|\|_{\psi_2} \leq CK\|a\|_2$ for absolute constant C.*

C.2 Ellipsoid Bound

Theorem 11 *Assume that $\hat{\theta}_t - \theta^* \in E_{r,t}$, the RE condition is satisfied in the set $E_{r,t}$ with parameter κ, and λ_t is suitably large. Then for any norm $R(\cdot)$ we have for constant $c > 0$*

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq c\psi(E_{r,t})\frac{\lambda_t}{\sqrt{\kappa}}\sqrt{t} \ . \tag{C.5}$$

Proof: Proof of Theorem 11. Define $\mathcal{L}(\theta) = \frac{1}{t}\|y_t - X_t\theta\|_2^2$ then

$$\begin{aligned}
\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle &= \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \\
\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) &= \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle + \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \\
\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) &= \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle + \frac{1}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .
\end{aligned} \tag{C.6}$$

By the definition of a dual norm

$$|\langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle| \leq R^*(\nabla\mathcal{L}(\theta^*))R(\hat{\theta}_t - \theta^*) \ .$$

By construction following (9.4) from Section 9.2, for any $\rho > 0$ (not just $\rho = 2$) we get

$$R^*(\nabla\mathcal{L}(\theta^*)) \leq \frac{\lambda_t}{\rho}$$

which implies

$$|\langle \nabla \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle| \leq \frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*)$$
$$\Rightarrow \langle \nabla \mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) \ . \tag{C.7}$$

Therefore, substituting (C.7) in (C.6)

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ . \tag{C.8}$$

By the triangle inequality we have

$$R(\hat{\theta}_t) - R(\theta^*) \geq -R(\hat{\theta}_t - \theta^*) \ .$$

Adding $\lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \geq -\lambda_t R(\hat{\theta}_t - \theta^*)$ to (C.8)

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) + \lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 - \lambda_t R(\hat{\theta}_t - \theta^*) \ .$$

Since $\hat{\theta}_t := \operatorname{argmin}_\theta \mathcal{L}(\theta) + \lambda_t R(\theta)$ then $\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) + \lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \leq 0$ therefore

$$0 \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 - \lambda_t R(\hat{\theta}_t - \theta^*) \ .$$

Re-arranging

$$0 \geq -\frac{1+\rho}{\rho} \lambda_t R(\hat{\theta}_t - \theta^*) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

By the definition of the norm compatibility constant $\psi(E_{r,t}) = \sup_{u \in E_{r,t}} \frac{R(u)}{\|u\|_2}$ we have $R(\hat{\theta}_t - \theta^*) \leq \|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t})$ which implies $-R(\hat{\theta}_t - \theta^*) \geq -\|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t})$. Therefore

$$0 \geq -\frac{1+\rho}{\rho} \lambda_t \|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t}) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

Substituting in the bound $\|\hat{\theta}_t - \theta^*\|_2 \leq \frac{1+\rho}{\rho} \frac{\lambda_t}{\kappa} \psi(E_{r,t})$ we obtain

$$0 \geq -\frac{1+\rho}{\rho} \lambda_t \frac{1+\rho}{\rho} \frac{\lambda_t}{\kappa} \psi(E_{r,t}) \psi(E_{r,t}) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2$$
$$0 \geq -\left(\frac{1+\rho}{\rho}\right)^2 \frac{\lambda_t^2}{\kappa} \psi^2(E_{r,t}) + \frac{1}{t} \|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

Therefore,

$$\left(\frac{1+\rho}{\rho}\right)^2 \frac{\lambda_t^2}{\kappa}\psi^2(E_{r,t}) \geq \frac{1}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2$$

and multiplying by t and taking the square root on both sides we obtain

$$\frac{1+\rho}{\rho}\frac{\lambda_t}{\sqrt{\kappa}}\psi(E_{r,t})\sqrt{t} \geq \|X_t(\hat{\theta}_t - \theta^*)\|_2 \ .$$

Noting that $\|X_t(\hat{\theta}_t - \theta^*)\|_2 = \|\hat{\theta}_t - \theta^*\|_{2,D_t}$ where $D_t = X_t^\top X_t$ end the proof. ∎

C.3 Algorithm

In this section, we will show that selecting an x_{t+1} following Algorithm 9, that we can compute a $\widetilde{\theta}_{t+1}$ such that the inequality $\langle x_{t+1}, \widetilde{\theta}_{t+1}\rangle \leq \langle x^*, \theta^*\rangle$ holds.

Lemma 8 *For a decision set $\mathcal{X}$ and and a confidence ellipsoid C_t, if we compute*

$$(x'_{t+1}, \theta'_{t+1}) = \underset{\substack{x\in\mathcal{X}\\ \theta\in C_t\cap S^{p-1}}}{\operatorname{argmin}} \ \langle x, \theta\rangle$$

and set x_{t+1} and $\widetilde{\theta}_{t+1}$ as

$$x_{t+1} = x'_{t+1} + \frac{\|x'_{t+1}\|_2}{2\sqrt{t}} v \cap \mathcal{X}$$
$$\widetilde{\theta}_{t+1} = \theta'_{t+1} - \frac{1}{\sqrt{t}}\frac{x'_{t+1}}{\|x'_{t+1}\|_2}$$

where v is a random vector such that $\|v\|_2 \leq 1$ then the inequality $\langle x_{t+1}, \widetilde{\theta}_{t+1}\rangle \leq \langle x'_{t+1}, \theta'_{t+1}\rangle \leq \langle x^*, \theta^*\rangle$ holds.

Proof: First, the inequality $\langle x'_{t+1}, \theta'_{t+1}\rangle \leq \langle x^*, \theta^*\rangle$ holds because we assume $\|\theta^*\|_2 = 1$, $\theta^* \in C_t$ with high-probability, and the optimization is over both x and θ. Now, we will show that for all x_{t+1} we can compute as above (without the intersection) there exists a $\widetilde{\theta}_{t+1}$ such that the inequality is satisfied which implies that the same holds for those

x_{t+1} in the intersection. First, observe

$$\begin{aligned}\langle x_{t+1}, \widetilde{\theta}_{t+1}\rangle &= \left\langle x'_{t+1} + \frac{\|x'_{t+1}\|_2}{2\sqrt{t}} v, \theta'_{t+1} - \frac{1}{\sqrt{t}} \frac{x'_{t+1}}{\|x'_{t+1}\|_2} \right\rangle \\ &= \langle x'_{t+1}, \theta'_{t+1}\rangle - \frac{\|x'_{t+1}\|_2}{\sqrt{t}} + \frac{\|x'_{t+1}\|_2}{2\sqrt{t}} \langle v, \theta'_{t+1}\rangle - \frac{1}{2t}\langle v, x'_{t+1}\rangle \ .\end{aligned}$$

We need to show that

$$\begin{aligned}&\langle x'_{t+1}, \theta'_{t+1}\rangle - \frac{\|x'_{t+1}\|_2}{\sqrt{t}} + \frac{\|x'_{t+1}\|_2}{2\sqrt{t}} \langle v, \theta'_{t+1}\rangle - \frac{1}{2t}\langle v, x'_{t+1}\rangle \leq \langle x'_{t+1}, \theta'_{t+1}\rangle \\ &\Rightarrow \frac{\|x'_{t+1}\|_2}{2\sqrt{t}} \langle v, \theta'_{t+1}\rangle \leq \frac{\|x'_{t+1}\|_2}{\sqrt{t}} + \frac{1}{2t}\langle v, x'_{t+1}\rangle \\ &\Rightarrow \frac{\|x'_{t+1}\|_2}{2} \langle v, \theta'_{t+1}\rangle \leq \|x'_{t+1}\|_2 + \frac{1}{2\sqrt{t}}\langle v, x'_{t+1}\rangle \\ &\Rightarrow -\frac{\|x'_{t+1}\|_2}{2} \leq \frac{1}{2\sqrt{t}}\langle v, x'_{t+1}\rangle \quad (\text{since } |\langle v, \theta'_{t+1}\rangle| \leq 1) \\ &\Rightarrow -\|x'_{t+1}\|_2 \leq \frac{1}{\sqrt{t}}\langle v, x'_{t+1}\rangle \\ &\Rightarrow 0 \leq \|x_{t+1}\|_2 + \frac{1}{\sqrt{t}}\langle v, x'_{t+1}\rangle \\ &\Rightarrow 0 \leq \|x_{t+1}\|_2 - \frac{1}{\sqrt{t}}\|x'_{t+1}\|_2 \quad (\text{from Cauchy-Schwarz inequality and } \|v\|_2 = 1) \quad \text{(C.9)}\end{aligned}$$

where the last line holds with equality for $t = 1$ and strict inequality for $t > 1$. Finally, since $\widetilde{\theta}_{t+1}$ lives inside a ball centered at $\theta'_{t+1} \in C_t$ with radius of the order $\frac{1}{\sqrt{t}}$ this implies that $\widetilde{\theta}_{t+1}$ is captured within the confidence ellipsoid $C'_t \supseteq C_t$ with radius $c\beta$ for constant $c \geq 1$. ∎

C.4 Bound on Regularization Parameter λ_t

We will prove the following main theorem.

Theorem 12 *For any $\gamma > 0$ and for absolute constant $L > 0$, with probability at least $1 - L\exp(-\gamma^2)$, the following bound holds uniformly for all $t = 1, \ldots, T$:*

$$R^*\left(\frac{1}{t} X_t^\top (y_t - X_t\theta^*)\right) \leq 2LKB \frac{\left(w(\Omega_R) + \sqrt{\gamma^2 + \log T}\frac{\phi(\Omega_R)}{2}\right)}{\sqrt{t}} \ . \qquad \text{(C.10)}$$

Proof: Proof of Theorem 12.

Recall the regularization parameter λ_t needs to satisfy the inequality

$$\lambda_t \geq \rho R^*(\nabla \mathcal{L}(\theta^*, Z_t)) = \rho R^* \left(\frac{1}{t} X_t^\top (y_t - X_t \theta^*) \right) \tag{C.11}$$

for $\rho > 1$. Two issues of the right hand side are (1) the expression depends on the unknown parameter θ^* and (2) the expression is a random variable since it depends on n vectors selected uniformly at random from the decision set $\mathcal{X}$ and a sequence of random noise terms $\eta_1, \ldots, \eta_t$. We can remove the dependence on θ^* by observing that $y_t - X_t \theta^*$ is precisely the t-dimensional noise vector $\omega_t = [\eta_1 \ldots \eta_t]^\top$. Therefore,

$$R^* \left(\frac{1}{t} X_t^\top (y_t - X_t \theta^*) \right) = R^* \left(\frac{1}{t} X_t^\top \omega_t \right) . \tag{C.12}$$

By the definition of the dual norm $R^* \left(\frac{1}{t} X_t^\top \omega_t \right) = \sup_{R(u) \leq 1} \frac{1}{t} \left\langle X_t^\top \omega_t, u \right\rangle$. The proof involves showing that $\frac{1}{t} \langle X_t^\top \omega_t, u \rangle$ is a martingale difference sequence (MDS) which concentrates as a sub-Gaussian random variable. Then, using a generic chaining argument, we show the supremum of such a quantity also concentrates as a sub-Gaussian random variable. We begin by observing that $\frac{1}{t} \langle X_t^\top \omega_t, u \rangle = \frac{1}{\sqrt{t}} \frac{1}{\sqrt{t}} \langle X_t^\top \omega_t, u \rangle$. We will save one of the $\frac{1}{\sqrt{t}}$ terms for later and now proceed to show how $\frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, u \right\rangle$ concentrates.

$\frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, u \right\rangle$ Concentrates as a Sub-Gaussian

First, let

$$\frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, u \right\rangle = \|u\|_2 \frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2} \right\rangle = \|u\|_2 \frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, q \right\rangle \tag{C.13}$$

where $q = \frac{u}{\|u\|_2}$. We focus on the term $\frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, q \right\rangle$. We can construct a martingale difference sequence (MDS) by observing that

$$\left\langle X_t^\top \omega_t, q \right\rangle = \langle \omega_t, X_t q \rangle = \sum_{\tau=1}^{t} \eta_\tau \langle x_\tau, q \rangle = \sum_{\tau=1}^{t} z_\tau \tag{C.14}$$

for $z_\tau = \eta_t \langle x_\tau, q \rangle$. Let the filtration be defined as $F_t = \{x_1, \ldots, x_{t+1}, \eta_1, \ldots, \eta_t\}$. Each z_τ can be seen as a MDS since

$$\mathbb{E}[z_\tau | F_{\tau-1}] = \mathbb{E}[\eta_\tau \langle x_\tau, q \rangle | F_{\tau-1}] = \langle x_\tau, q \rangle \cdot \mathbb{E}[\eta_\tau | F_{\tau-1}] = 0 \tag{C.15}$$

because x_τ is $F_{\tau-1}$ measurable and η_τ is F_τ measurable. Additionally, each z_τ follows a sub-Gaussian distribution with parameter KB because $\|\eta_\tau \langle x_\tau, q\rangle\|_{\psi_2} \leq KB$ (Assumption 2). Since each z_τ is a bounded MDS, we use the Azuma-Hoeffding inequality to show $\sum_{\tau=1}^{t} z_\tau$ concentrates as a sub-Gaussian with parameter KB. For all $\gamma \geq 0$

$$\begin{aligned} P\left(\left|\sum_{\tau=1}^{t} z_\tau\right| \geq \gamma\right) &= P\left(\left|\langle X_t^\top \omega_t, q\rangle\right| \geq \gamma\right) \\ &= P\left(\left|\left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\right\rangle\right| \geq \gamma\right) \leq 2\exp\left(\frac{-\gamma^2}{2tK^2B^2}\right) \\ &= P\left(\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\right\rangle\right| \geq \zeta\right) \leq 2\exp\left(\frac{-\zeta^2}{2K^2B^2}\right) . \end{aligned} \tag{C.16}$$

From (C.16) and (C.2) in Definition 9 (Section C.1) we can see that $\frac{1}{\sqrt{t}}\left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\right\rangle$ concentrations as a sub-Gaussian with $\|\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\rangle\|_{\psi_2} \leq KB$.

Next, we show that the term $\frac{1}{\sqrt{t}}\left\langle X_t^\top \omega_t, u\right\rangle$ also concentrations as a sub-Gaussian with $\|\langle X_t^\top \omega_t, u\rangle\|_{\psi_2} \leq \|u\|_2 KB$ using (C.16) as

$$\begin{aligned} & P\left(\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\right\rangle\right| \geq \zeta\right) \\ =& P\left(\|u\|_2\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, \frac{u}{\|u\|_2}\right\rangle\right| \geq \|u\|_2\zeta\right) \\ =& P\left(\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, u\right\rangle\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-\epsilon^2}{2\|u\|_2^2K^2B^2}\right) \end{aligned} \tag{C.17}$$

where $\epsilon = \|u\|_2\zeta$ which implies $\zeta = \epsilon/\|u\|_2$. We went through showing the above because the generic chaining argument we will invoke to bound $\sup_{R(u)\leq 1}\frac{1}{\sqrt{t}}\left\langle X_t^\top \omega_t, u\right\rangle$ requires that $\frac{1}{\sqrt{t}}\left\langle X_t^\top \omega_t, u\right\rangle$ is a sub-Gaussian random variable.

Bound on $\sup_{R(u)\leq 1}\frac{1}{\sqrt{t}}\langle X_t^\top \omega_t, u\rangle$ via Generic Chaining

We obtain a high-probability bound on $\sup_{R(u)\leq 1}\frac{1}{\sqrt{t}}\langle X_t^\top \omega_t, u\rangle$ using a generic chaining argument from [115, 116]. This involves (1) showing that the absolute difference of two sub-Gaussian processes concentrates as a sub-Gaussian, (2) showing the expectation over the supremum of the absolute difference of two sub-Gaussian processes is upper bounded by the sub-Gaussian width of a set from which the processes are indexed from, and (3) showing the supremum of a sub-Gaussian process is concentrated around its expectation and therefore, around the sub-Gaussian width with high-probability.

(1) Sub-Gaussian Process Concentration

First, we show that the absolute difference of two sub-Gaussian processes concentrates as a sub-Gaussian. Let $Y_u = \frac{1}{\sqrt{t}}\langle X_t^\top \omega_t, u\rangle$ indexed by $u \in \Omega_R$ and $Y_v = \frac{1}{\sqrt{t}}\langle X_t^\top \omega_t, v\rangle$ indexed by $v \in \Omega_R$ be two zero-mean (since they are both a MDS sum), random symmetric processes (since $(Y_u)_{u\in\Omega_r}$ has the same law as $(-Y_u)_{u\in\Omega_R}$ via (C.16) and ω_t is symmetric and similarly for Y_v). Then by construction

$$|Y_u - Y_v| = \frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, u - v\right\rangle\right| \ .$$

Using the bound we established in (C.17), we obtain the following bound on the absolute difference of two sub-Gaussian random processes Y_u and Y_v as

$$P\left(\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top \omega_t, u - v\right\rangle\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-\epsilon^2}{2\|u-v\|_2^2 K^2 B^2}\right) \tag{C.18}$$

which shows $|Y_u - Y_v|$ concentrates as a sub-Gaussian random variable with $\|\|Y_u - Y_v\|\|_{\psi_2} = \|u-v\|_2 KB$.

(2) Bound on $\mathbb{E}\left[\sup_{R(u)\leq 1}\frac{1}{t}\langle X_t^\top \omega_t, u\rangle\right]$

In order to establish a high-probability bound on $\sup_{R(u)\leq 1}\frac{1}{t}\langle X_t^\top \omega_t, u\rangle$ we need to prove a bound on $\mathbb{E}\left[\sup_{R(u)\leq 1}\frac{1}{t}\langle X_t^\top \omega_t, u\rangle\right]$. To prove such a bound, we will apply a generic chaining argument for upper bounds on such sub-Gaussian processes. For the generic chaining argument, we will need the result in (C.18) and the following lemma.

Lemma 9 ***([115], Theorem 2.1.5)*** *Consider two processes $(Y_u)_{u\in\Omega_R}$ and $(X_u)_{u\in\Omega_R}$ indexed by the same set. Assume that the process $(X_u)_{u\in\Omega_R}$ is Gaussian and that the process $(Y_u)_{u\in\Omega_R}$ satisfies the condition*

$$\forall \epsilon > 0, \forall u, v \in \Omega_R, P(|Y_u - Y_v| \geq \epsilon) \leq 2\exp\left(-\frac{\epsilon^2}{d(u,v)^2}\right) \tag{C.19}$$

where $d(u,v)$ is a distance function which we assume is $d(u,v) = \|u-v\|_2$ for the set Ω_R. Then we have

$$\mathbb{E}\left[\sup_{u,v\in\Omega_R}|Y_u - Y_v|\right] \leq L\mathbb{E}\left[\sup_{u\in\Omega_R} X_u\right] \tag{C.20}$$

where L is an absolute constant.

$\mathbb{E}\left[\sup_{u\in\Omega_R} X_u\right]$ is the Gaussian width $w(\Omega_R)$ of the set Ω_R (Definition 11, Section C.1). Since our process $|Y_u - Y_v|$ concentrates as a sub-Gaussian, we scale the Gaussian width by the sub-Gaussian parameter similar to [10, Theorem 8] to get

$$\mathbb{E}\left[\sup_{u,v\in\Omega_R} |Y_u - Y_v|\right] \leq LKB\mathbb{E}\left[\sup_{u\in\Omega_R} X_u\right] = LKBw(\Omega_R)\ . \tag{C.21}$$

The second result we need is the following.

Lemma 10 ***([115], Lemma 1.2.8)*** *If the process $(Y_u)_{u\in\Omega_R}$ is symmetric then*

$$\mathbb{E}\left[\sup_{u,v\in\Omega_R} |Y_u - Y_v|\right] = 2\mathbb{E}\left[\sup_{u\in\Omega_R} Y_u\right]\ . \tag{C.22}$$

Since our processes are symmetric we get the following lemma.

Lemma 11 *From (C.18) the condition of Lemma 9 is satisfied in the sub-Gaussian case so using Lemma 9 and Lemma 10 for some absolute constant L we obtain*

$$\begin{aligned} &\mathbb{E}\left[\sup_{u,v\in\Omega_R} |Y_u - Y_v|\right] = 2\mathbb{E}\left[\sup_{u\in\Omega_R} |Y_u|\right] \leq 2LKBw(\Omega_R) \\ &\Rightarrow 2\mathbb{E}\left[\sup_{u\in\Omega_R} \frac{1}{\sqrt{t}} \left|\left\langle X_t^\top \omega_t, u\right\rangle\right|\right] \leq 2LKBw(\Omega_R)\ . \end{aligned} \tag{C.23}$$

(3) Concentration of $\sup_{R(u)\leq 1} \frac{1}{\sqrt{t}} \left\langle X_t^\top \omega_t, u\right\rangle$

To complete the argument, we need the following lemma.

Lemma 12 ***([116], Theorem 2.2.27)*** *If the process (Y_u) satisfies (C.19) or similarly (C.18) for the sub-Gaussian case then for $\epsilon > 0$ one has*

$$P\left(\sup_{u,v\in\Omega_R} |Y_u - Y_v| \geq L\big(\gamma_2(\Omega_R, d(u,v)) + \epsilon\Delta(\Omega_R)\big)\right) \leq L\exp(-\epsilon^2)\ . \tag{C.24}$$

Note, the function $\Delta(\Omega_R) = \sup_{u,v\in\Omega_R} d(u,v)$ is the diameter of the set Ω_R. For our setting, $d(u,v) = \|u - v\|_2$ so we replace $\Delta(\Omega_R)$ with $\phi(\Omega_R)$ as detailed in Definition 5 in Section 9.3.1. The specifics of the $\gamma_2(\cdot,\cdot)$ function are not necessary for this work since we can bound it and simplify Lemma 12 by using the following lemma.

Lemma 13 ***([116], Theorem 2.4.1)*** *For some universal constant L we have*

$$\frac{1}{L}\gamma_2(\Omega_R, d(u,v)) \leq \mathbb{E}\left[\sup_{u\in\Omega_R} Y_u\right] \leq L\gamma_2(\Omega_R, d(u,v)) \ . \tag{C.25}$$

Combining Lemma 12 with Lemma 13, using Lemma 11, and our definitions of Y_u and Y_v for any $\epsilon > 0$ we get

Lemma 14

$$P\left(\sup_{R(u)\leq 1}\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top\omega_t, u\right\rangle\right| \geq 2LKBw(\Omega_R)+\epsilon\right) \leq L\exp\left(-\left(\frac{\epsilon}{LKB\phi(\Omega_R)}\right)^2\right) \ . \tag{C.26}$$

Proof: Proof of Lemma 14.

$$\begin{aligned}
&P\left(\sup_{u,v\in\Omega_R}|Y_u - Y_v| \geq L\big(\gamma_2(\Omega_R, d(u,v)) + \zeta\Delta(\Omega_R)\big)\right)\\
&= P\left(\sup_{u,v\in\Omega_R}|Y_u - Y_v| \geq L\gamma_2(\Omega_R, d(u,v)) + \epsilon\right)\\
&\leq P\left(\sup_{u,v\in\Omega_R}|Y_u - Y_v| \geq \mathbb{E}\left[\sup_{u,v\in\Omega_R}|Y_u - Y_v|\right] + \epsilon\right)\\
&= P\left(\sup_{u\in\Omega_R}|Y_u| \geq 2\mathbb{E}\left[\sup_{u\in\Omega_R}|Y_u|\right] + \epsilon\right)\\
&= P\left(\sup_{R(u)\leq 1}\frac{1}{\sqrt{t}}\left|\left\langle X_t^\top\omega_t, u\right\rangle\right| \geq 2LKBw(\Omega_R)+\epsilon\right) \leq L\exp\left(-\left(\frac{\epsilon}{LKB\phi(\Omega_R)}\right)^2\right) \ .
\end{aligned}$$

where the first line comes from the left-hand side of Lemma 12, the second line comes from the fact that $\Delta(\Omega_R) \leq \gamma_2(\Omega_R, d(u,v))$ from [116] Definition 2.2.19, the third line comes from Lemma 13, the fourth line comes from Lemma 10, the fifth line comes from Lemma 11, and the last line follows from our construction of the process Y_u and the right-hand side of Lemma 12. ∎

Dividing the other $\sqrt{t}$ through and setting $\epsilon/\sqrt{t} = \alpha 2LKBw(\Omega_R)/\sqrt{t}$ we get

Lemma 15

$$P\left(R^*\left(\frac{1}{t}X_t^\top\omega_t\right) \geq 2LKB(1+\alpha)\frac{w(\Omega_R)}{\sqrt{t}}\right) \leq L\exp\left(-\left(\frac{2\alpha w(\Omega_R)}{\phi(\Omega_R)}\right)^2\right) \ . \tag{C.27}$$

Proof: Proof of Lemma 15

$$P\left(R^*\left(\frac{1}{\sqrt{t}}X_t^\top\omega_t\right) \geq 2LKBw(\Omega_R)+\epsilon\right) \leq L\exp\left(-\left(\frac{\epsilon}{LKB\phi(\Omega_R)}\right)^2\right)$$

$$P\left(R^*\left(\frac{1}{t}X_t^\top\omega_t\right) \geq 2LKB\frac{w(\Omega_R)}{\sqrt{t}}+\gamma\right) \leq L\exp\left(-\left(\frac{\sqrt{t}\gamma}{LKB\phi(\Omega_R)}\right)^2\right)$$

$$P\left(R^*\left(\frac{1}{t}X_t^\top\omega_t\right) \geq 2LKB\frac{w(\Omega_R)}{\sqrt{t}}+2LKB\alpha\frac{w(\Omega_R)}{\sqrt{t}}\right) \leq L\exp\left(-\left(\frac{\sqrt{t}\alpha 2LKB\frac{w(\Omega_R)}{\sqrt{t}}}{LKB\phi(\Omega_R)}\right)^2\right)$$

$$P\left(R^*\left(\frac{1}{t}X_t^\top\omega_t\right) \geq 2LKB(1+\alpha)\frac{w(\Omega_R)}{\sqrt{t}}\right) \leq L\exp\left(-\left(\frac{2\alpha w(\Omega_R)}{\phi(\Omega_R)}\right)^2\right) .$$

where the first inequality is from Lemma 14, the second inequality is from multiplying both sides by $\frac{1}{\sqrt{t}}$ and setting $\gamma = \frac{\epsilon}{\sqrt{t}}$, and the third inequality is from setting $\gamma = \alpha 2LKB\frac{w(\Omega_R)}{\sqrt{t}}$. ∎

Lemma 15 gives a high-probability bound on the value of $R^*\left(X_t^\top\omega_t\right)$ for round t but to complete the proof of Theorem 12 we need a bound which holds simultaneously for all rounds T with high-probability. To obtain such a bound, we can set $\alpha^2 = (\gamma^2+\log T)\left(\frac{\phi(\Omega_R)}{2w(\Omega_R)}\right)^2$ and apply a union bound for all t

$$\begin{aligned}
&\bigcup_{t=1}^{T} P\left(R^*\left(\frac{1}{t}X_t^\top\omega_t\right) \geq 2LKB\left(1+\sqrt{\gamma^2+\log T}\left(\frac{\phi(\Omega_R)}{2w(\Omega_R)}\right)\right)\frac{w(\Omega_R)}{\sqrt{t}}\right)\\
&\leq \sum_{t=1}^{T} L\exp\left(-(\gamma^2+\log T)\left(\frac{\phi(\Omega_R)}{2w(\Omega_R)}\right)^2\left(\frac{2w(\Omega_R)}{\phi(\Omega_R)}\right)^2\right)\\
&= L\sum_{t=1}^{T}\exp\left(-\gamma^2-\log T\right)\\
&= L\sum_{t=1}^{T}\exp\left(-\gamma^2\right)\times\frac{1}{T}\\
&= L\exp\left(-\gamma^2\right) .
\end{aligned}$$

Rearranging the terms ends the proof of Theorem 12. ∎

Appendix D

Structured Stochastic Generalized Linear Bandits

D.1 Generalized Ellipsoid Bound

Theorem 15 *Assume that $\hat{\theta}_t - \theta^* \in E_{r,t}$, the RSC condition is satisfied in the set $E_{r,t}$ with parameter κ, and λ_t is suitably large. Then for any norm $R(\cdot)$ we have for constant $c > 0$*

$$\|\hat{\theta}_t - \theta^*\|_{2,D_t} \leq c\psi(E_{r,t})\frac{\lambda_t}{\sqrt{\kappa}}\sqrt{t} \ . \tag{D.1}$$

Proof: The following decomposition follows [10]. Let $\mathcal{L}(\theta)$ be the negative log likelihood loss function $\mathcal{L}(\theta) = \frac{1}{t}\sum_{i=1}^{t}\{\varphi(\langle x_i, \theta\rangle) - y_i\langle x_i, \theta\rangle\}$ then we have

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^*\rangle = \frac{1}{t}\sum_{i=1}^{t}\Big\{\varphi(\langle x_i, \hat{\theta}_t\rangle) - y_i\langle x_i, \hat{\theta}_t\rangle\Big\} - \frac{1}{t}\sum_{i=1}^{t}\{\varphi(\langle x_i, \theta^*\rangle) - y_i\langle x_i, \theta^*\rangle\}$$
$$- \left\langle \frac{1}{t}\sum_{i=1}^{t} x_i\varphi'(\langle x_i, \theta^*\rangle) - y_i x_i, \hat{\theta}_t - \theta^* \right\rangle$$

where $\varphi'(\cdot)$ denotes the first derivative of $\varphi(\cdot)$. Simplifying and applying the mean value theorem twice we obtain

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla_{\theta^*}\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^*\rangle = \frac{1}{t}\sum_{i=1}^{t}\varphi''(\langle x_i, \theta^*\rangle + \gamma_i\langle x_i, \hat{\theta}_t - \theta^*\rangle)\langle x_i, \hat{\theta}_t - \theta^*\rangle^2$$

where $\gamma \in [0,1]$. Since the log-partition function is convex, its second derivative is non-negative and we will assume that it is bounded away from zero, i.e., there exists a constant ℓ such that $\varphi''(\cdot) \geq \ell > 0$. Therefore,

$$\begin{aligned}\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla_{\theta^*}\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle &= \frac{1}{t}\sum_{i=1}^{t} \varphi''(\langle x_i, \theta^* \rangle + \gamma_i \langle x_i, \hat{\theta}_t - \theta^* \rangle)\langle x_i, \hat{\theta}_t - \theta^* \rangle^2 \\ &\geq \frac{\ell}{t}\sum_{i=1}^{t} \langle x_i, \hat{\theta}_t - \theta^* \rangle^2 \\ &= \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .\end{aligned}$$

Then we can follow the analysis in Section C.2 as

$$\begin{aligned}\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle &= \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \\ \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) &= \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle + \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) - \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle \\ \mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) &\geq \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ . \qquad \text{(D.2)}\end{aligned}$$

By the definition of a dual norm

$$|\langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle| \leq R^*(\nabla\mathcal{L}(\theta^*))R(\hat{\theta}_t - \theta^*) \ .$$

By construction following (10.4) from Section 10.2.1, for any $\rho > 0$ we get

$$R^*(\nabla\mathcal{L}(\theta^*)) \leq \frac{\lambda_t}{\rho}$$

which implies

$$\begin{aligned}|\langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle| &\leq \frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) \\ \Rightarrow \langle \nabla\mathcal{L}(\theta^*), \hat{\theta}_t - \theta^* \rangle &\geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) \ . \qquad \text{(D.3)}\end{aligned}$$

Therefore, substituting (D.3) in (D.2)

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ . \qquad \text{(D.4)}$$

By the triangle inequality we have

$$R(\hat{\theta}_t) - R(\theta^*) \geq -R(\hat{\theta}_t - \theta^*) \ .$$

Adding $\lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \geq -\lambda_t R(\hat{\theta}_t - \theta^*)$ to (D.4)

$$\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) + \lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 - \lambda_t R(\hat{\theta}_t - \theta^*) \ .$$

Since $\hat{\theta}_t := \text{argmin}_\theta \, \mathcal{L}(\theta) + \lambda_t R(\theta)$ then $\mathcal{L}(\hat{\theta}_t) - \mathcal{L}(\theta^*) + \lambda_t(R(\hat{\theta}_t) - R(\theta^*)) \leq 0$ therefore

$$0 \geq -\frac{\lambda_t}{\rho} R(\hat{\theta}_t - \theta^*) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 - \lambda_t R(\hat{\theta}_t - \theta^*) \ .$$

Re-arranging

$$0 \geq -\frac{1+\rho}{\rho}\lambda_t R(\hat{\theta}_t - \theta^*) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

By the definition of the norm compatibility constant $\psi(E_{r,t}) = \sup_{u \in E_{r,t}} \frac{R(u)}{\|u\|_2}$ we have $R(\hat{\theta}_t - \theta^*) \leq \|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t})$ which implies $-R(\hat{\theta}_t - \theta^*) \geq -\|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t})$. Therefore

$$0 \geq -\frac{1+\rho}{\rho}\lambda_t \|\hat{\theta}_t - \theta^*\|_2 \psi(E_{r,t}) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ .$$

Substituting in the bound $\|\hat{\theta}_t - \theta^*\|_2 \leq \frac{1+\rho}{\rho}\frac{\lambda_t}{\kappa}\psi(E_{r,t})$ we obtain

$$\begin{aligned} 0 &\geq -\frac{1+\rho}{\rho}\lambda_t \frac{1+\rho}{\rho}\frac{\lambda_t}{\kappa}\psi(E_{r,t})\psi(E_{r,t}) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \\ 0 &\geq -\left(\frac{1+\rho}{\rho}\right)^2 \frac{\lambda_t^2}{\kappa}\psi^2(E_{r,t}) + \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2 \ . \end{aligned}$$

Therefore,

$$\left(\frac{1+\rho}{\rho}\right)^2 \frac{\lambda_t^2}{\kappa}\psi^2(E_{r,t}) \geq \frac{\ell}{t}\|X_t(\hat{\theta}_t - \theta^*)\|_2^2$$

and multiplying by $\frac{t}{\ell}$ and taking the square root on both sides we obtain

$$\frac{1+\rho}{\rho\sqrt{\ell}}\frac{\lambda_t}{\sqrt{\kappa}}\psi(E_{r,t})\sqrt{t} \geq \|X_t(\hat{\theta}_t - \theta^*)\|_2 \ .$$

Noting that $\|X_t(\hat{\theta}_t - \theta^*)\|_2 = \|\hat{\theta}_t - \theta^*\|_{2,D_t}$ where $D_t = X_t^\top X_t$ ends the proof. ∎