IMPERIAL

Imperial College London

Department of Mathematics

Optimal Resource Allocation using Multi-Armed Bandits

Author: Ilie-Alexandru Botosan (CID: 01869692)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2023-2024

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

First and foremost, I want to thank my supervisor, Dr. Paul Bilokon and my tutor Dr. Eyal Neumann for supporting me throughout the course of this year and guiding me on the right way. I would have never been able to complete this work without their help and I'm very grateful.

Furthermore, I would like to thank my external supervisor Daniel Wyles for the support he has given me while doing my internship this summer.

Lastly, I want to thank my family and my friends for the constant support.

Abstract

Efficient resource allocation is crucial across diverse sectors, from finance and public policy to industrial management. Despite advances in financial economics, exemplified by the foundational work of Markowitz [74], Miller, and Sharpe [96], practical applications of these models have often failed to yield significant productivity gains, contributing to the ongoing "productivity paradox." This thesis addresses the limitations of traditional resource allocation methods within a finance-related context and explores innovative solutions using Online Learning techniques, specifically within the Multi-Armed Bandit (MAB) framework, a branch of Reinforcement Learning (RL).

We propose a novel algorithm that improves the MAB framework of Upper Confidence Bound (UCB) algorithms which optimizes the Sharpe ratio and discuss a potential extension to dynamic, non-stationary environments by incorporating changepoint detection techniques, improving mean estimation, and maintaining logarithmic regret bounds. Our method adapts to shifts in data, addressing key vulnerabilities in existing MAB algorithms and outperforming classical models in portfolio selection tasks. Through extensive theoretical analysis and empirical validation, this work demonstrates the potential of our enhanced MAB framework to improve resource allocation in complex, real-world scenarios.

Furthermore, we analyse the performance of various stochastic and adversarial bandit algorithms and suggest an expert selection specific for a finance-related context.

Contents

List of Figures

List of Tables

Introduction

Efficient resource allocation is a critical concern that extends far beyond the realm of portfolio management. It is a fundamental challenge faced by individuals, charities, corporations, governments at all levels, and international organisations. Their success, and sometimes even their survival, depends on how effectively they can allocate their resources.

During the Cold War (1947–1991), the Arms Race and Space Race were emblematic of the broader struggle between market economies and centrally planned economies, centered around differing methods of resource allocation. Market economies relied on decentralized decision-making, while planned economies were characterized by centralized control.

Ultimately, the world has moved toward various mixtures of market and centrally planned economies. The foundational work of Harry M. Markowitz, Merton H. Miller, and William F. Sharpe, for which they were awarded the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel in 1990, has become universally acknowledged in both economic systems. Their pioneering contributions to financial economics have been extensively utilized for allocating resources, not exclusively financial, to achieve an optimal risk-return profile.

However, the consequences of inefficient resource allocation can be severe, leading to artificial scarcity and unmet needs, which contribute to poverty. This in turn hampers productivity, not just through psychological impacts, but also by limiting access to essential resources, including advanced tools and technology [32].

Despite the transformative insights from Markowitz, Miller, and Sharpe, there has not been a dramatic increase in productivity. As Robert Merton Solow, another Nobel laureate, famously observed in 1987, "You can see the computer age everywhere but in the productivity statistics." This observation, known as the "productivity paradox," continues today as "productivity paradox 2.0," spanning from the 1970s through to the present day [20].

This productivity shortfall is concerning, as socio-economic challenges are key drivers of conflict. What are often perceived as ethnic conflicts are frequently struggles over power and economic resources [102]. Enhancing productivity could reduce the likelihood of such conflicts, whereas failing to address these inefficiencies could exacerbate tensions and contribute to future disputes.

While there are numerous non-mathematical and non-scientific reasons for suboptimal resource allocation (such as when optimal allocation conflicts with the incentives of individual decision-makers or implementers) this work will not focus on these issues. Instead, we will examine the classical Markowitz-Miller-Sharpe framework, identify its deficiencies and vulnerabilities, and explore ways to address these issues using recent advances in applied mathematics.

Among these avenues of research, we will focus our attention on Multi-Armed Bandits (MAB), a branch of Reinforcement Learning (RL). The subject has attracted a lot of attention due to the recent emergence of large scale applications such as online web advertisement placement, online topic-detection in social communities, online web ranking, finding shortest path for internet packet routing, email spam filtering, portfolio selection and many more. In recent years, tools from an even broader subject, convex optimization,

have influenced the design of many online learning algorithms. As a result, Online Convex Optimization (OCO) [34, 33, 67] has emerged as a unified abstraction, which helps in solving problems efficiently and reliably and also facilitates the theoretical analysis.

Reinforcement Learning is a powerful framework for developing intelligent agents capable of learning complex behaviors. Despite its challenges, RL continues to be a vibrant area of research and development, with the potential to revolutionize numerous industries and applications. The research of this field is paramount, as the real-world applications are countless and, as opposed to Deep Learning designs, one can analyze and explain how a certain method work (or doesn't) through rigorous analysis.

Among the many useful applications of RL, some of the more notable ones reside in large industries, such as healthcare, energy management, industrial automation and finance. Multi-armed bandits were introduced by William R. Thompson in an article published in 1933 in *Biometrika* [104]. Thompson was interested in medical trials and the cruelty of running a trial blindly, without adapting the treatment allocations on the fly as the drug appears more or less effective.

In this thesis, we aim to analyse the portfolio choice problem within a MAB setting and propose several extensions to the current frameworks. In Chapter 1 we formulate our problem and discuss the first developments towards solving it, transitioning thereafter to Chapter 2 where we introduce the bandit framework. We present a variety of algorithms and discuss their limitations, followed by a novel improvement with theoretical guarantees which maintains logarithmic regret for the returns' suboptimal gaps. In Chapter 3 we take a look at changepoint detection and address the mean estimation problem of incremental returns, tackling the non-stationarity issues and further extending the proposed framework from Chapter 2. We conduct extensive experiments on all algorithms in the final chapter, putting everything together and performing stress tests to discuss the limits of the bandit framework within the portfolio choice problem.

We will model our matter at hand and compare our Multi-Armed Bandit portfolio based on benchmark models against classical methods of resource allocation. At the end, we conclude with proposing new approaches towards this issue and explain why it's an improvement of the current setting.

Chapter 1

Optimal Resource Allocation

Portfolio choice problems have had a significant impact on the finance industry, influencing everything from pension funds and mutual funds to insurance and corporate risk management [4]. Modern portfolio theory and analysis are grounded in the pioneering work of Markowitz [74]. However, due to the disappointing performance of these models in out-of-sample scenarios [19], researchers have continually sought innovative approaches to address portfolio choice challenges. This has led to the development of numerous extensions of the Markowitz framework [36], the exploration of the optimal growth portfolio based on the Kelly criterion [105], and the adoption of linear programming techniques for portfolio optimization [63].

In this chapter we aim to formulate the portfolio choice problem in depth and discuss the limits of classical methods. In section 1.1 we introduce the core notions we will use throughout the paper, whereas in section 1.2 we take a look at the Markowitz framework and its limitations, while 1.3 covers an estimation method for the well-known Sharpe ratio. The purpose of this chapter, though less technical, is crucial as it lays the groundwork for the problem at hand, creating a natural transition to Chapter 2.

1.1 Literature Review and Problem Formulation

Optimal allocation of resources has been and it sill is a heavily debated topic, as it can be noticed everywhere. The most noticeable field in which this subject is directly applied represents transportation logistics in heavy industries, as the costs of shipping are considerable when it comes to enormous quantities. A demand for good allocation comes in all transportation industries, as getting from point A to point B with the lowest cost is something everybody desires.

Simultaneously, globalization and the rapid integration of markets have generated vast amounts of data in the finance industry, necessitating the use of advanced data analysis tools. Among these, machine learning algorithms stand out for their ability to construct, update, and apply models in real-time, efficiently processing large datasets. In response, machine learning researchers have dedicated significant effort to designing trading strategies based on real-time data streams [49, 56, 18, 5, 47]. The Kelly criterion [105], which aims to maximize long-term growth, has emerged as a particularly popular choice. For instance, one notable study utilizes the moving average reversion phenomenon in the stock market to optimize return growth [68]. For a comprehensive overview of online portfolio strategies, we refer to the survey by Li and Hoi [67] and the extensive references therein.

To address these challenges in applying conventional bandit algorithms to portfolio choice, many academics and practitioners researched and developed this topic and sought out to improve the framework and adapt their methods to the problem at hand. Notable work in this direction, which we will use as benchmarks against our proposal, is represented

by a selection of bandit algorithms tailored to a portfolio choice problem.

In a Portfolio Selection Problem (PSP) [44], an investor's goal is to develop a strategy for distributing a limited amount of wealth among various assets. Each asset provides a distinct investment opportunity, and the combined result of these allocations forms the overall portfolio. Assets are deemed risky when their prices are uncertain, necessitating careful consideration of this risk when making allocation decisions. The interval between consecutive portfolio allocation decisions is called a period. When only one allocation decision is made for the entire investment horizon, it is known as a Single-Period PSP. Conversely, in a Multi-Period PSP, the investor continuously makes sequential decisions over multiple periods, transforming it into an online decision-making challenge. Throughout this process, the investor's objective is to optimize a specific function, which could involve maximizing returns, minimizing risk, or finding an optimal balance between the two.

We can think at the problem in the following way: suppose we are in charge of N traders, each one of them having his/her own trading strategy. Our concern is not how they managed to produce it or how they aim to improve it, as we are interested only in their Profit and Loss (PnL) evolution. We will reduce their results to equity curves, so we are left with k such curves that signify each trader's performance. The results may be correlated, combinations between others or black-box algorithms that even their creators cannot explain how they work.

Our job is now to choose from all those N trading strategies only the best ones that will perform well in the future. Because humans are liable to many biases we'll need an algorithm to make that choice for us. Data snooping bias is the main concern [70, 52], as many strategies that seem to good to be true on previous data tend to be flukes and will start loosing money shortly after going in production.

This is why we try to eliminate human judgment completely and try to formulate the problem clearly such that we know how to approach this selection method. As mentioned earlier, we will have an online learning approach, and define the equity curves as

$$M \in \mathcal{M}_{n\times k}(\mathbb{R}), \tag{1.1.1}$$

where n is the number of curves and k is the number of time increments we have at our disposal. At each step t, we will assign weights $W_t = (w_{t1}, w_{t2}, ..., w_{tk})$ to each strategy that will determine our choices for the next time interval, where we will compute the aggregate PnL of our allocation, representing the incremental reward that we will count towards making the next decision.

The computational cost of considering every asset combination grows exponentially as the number of assets increases, thus posing a great challenge on finding an optimal solution. Ito [55] discussed about the "Full-Feedback" algorithm in his work, proving that the issue at hand can become NP-complete.

Now, weight allocation is the central task of the thesis and we need to define how a portfolio is valued. While there is the option of using log returns and take the cumulative wealth as multiplicative, we will instead consider incremental returns, so that the PnL contribution after each round is represented by

$$\Delta_t = \sum_{i=1}^{k} w_{t-1,i}(M_{t,i} - M_{t-1,i}).$$

The reasoning behind this decision is that we can manage the returns within a bandit framework more easily if they are just increments of the PnL curves, instead of dividing every round by the previous entry in the array. We want to avoid numerical errors as much as possible and emphasize on the inherent features of the MAB algorithms. Moreover,

there is good cause for considering incremental returns, as they have real life applications, especially in commodity trading simply because there are extreme cases where the prices can go negative [42].

1.2 The Markowitz Framework

The Markowitz-Miller-Sharpe framework is widely known among practitioners in economics, finance, and management, though it is not without significant issues. One of the key components of this framework, the *Sharpe ratio*, was introduced by Sharpe in 1966 [96] as a measure of risk-adjusted returns. Since its introduction, the Sharpe ratio and its variants have become popular objective functions in resource allocation optimization, as first proposed by Markowitz in 1952 [74], and further discussed in subsequent literature [97, 29].

Given a time series of returns, or risk premia, denoted as R_t, which are independent and identically distributed (i.i.d.), $R_t \sim \mathcal{N}(\mu, \sigma^2)$ (where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2), the Sharpe ratio is defined as:

$$SR = \frac{\mu}{\sigma}.$$

Since μ and σ are typically unknown, the true value of the Sharpe ratio cannot be precisely determined, making its calculation subject to significant estimation errors.

Consider an asset whose price follows the geometric Brownian motion, described by the stochastic differential equation [15]:

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t,$$

where μ is the **percentage drift** and σ is the **percentage volatility**. Suppose we sample S_t over a time interval $0 \leq t \leq T$, divided into N equal segments of length $\delta t = \frac{T}{N}$. Let the observed price at time t_i be S_i. The return over the period $[t_i, t_{i+1}]$ is defined as:

$$R_i = \frac{S_{i+1} - S_i}{S_i}.$$

We propose a set of estimators for μ and σ characterised by the following lemma:

Lemma 1.2.1. *Let $(S_t)_{0\leq t\leq T}$ be a Geometric Brownian Motion process with drift μ and volatility σ, described by the following stochastic differential equation:*

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t,$$

where S_t is sampled over a time interval $0 \leq t \leq T$ divided into N equal segments of length $\delta t = \frac{T}{N}$. Then, the estimators $\hat{\mu}$ and $\hat{\sigma}^2$ defined as

$$\hat{\mu} = \frac{1}{N\delta t} \sum_{i=0}^{N-1} R_i, \quad \hat{\sigma}^2 = \frac{1}{(N-1)\delta t} \sum_{i=0}^{N-1} (R_i - \hat{\mu}\delta t)^2 \tag{1.2.1}$$

are unbiased as $\delta t \longrightarrow 0$.

We prove in Appendix A.1 that the estimators are unbiased and, moreover, we provide a characterisation regarding the accuracy for estimating the parameters.

Lemma 1.2.2. *Let $\hat{\mu}$ and $\hat{\sigma}^2$ be the estimators defined in Lemma 1.2.1. Then, the following properties hold:*

$$Var(\hat{\mu} - \mu) = \mathcal{O}\left(\frac{1}{T}\right), \quad Var\left(\frac{\hat{\sigma}^2}{\sigma^2}\right) = \mathcal{O}\left(\frac{1}{N}\right). \tag{1.2.2}$$

Thus, the accuracy of the estimate for μ depends only on T and not on N, indicating that more frequent sampling does not improve the accuracy; only a longer time interval does. Conversely, the accuracy of the estimate for σ depends solely on N and not on T. However, increasing T poses practical challenges due to potential nonstationarity in the process, while increasing N may be limited by microstructure noise [11].

These issues in estimating the Sharpe ratio have long been recognized, as noted by Robert C. Merton [77], and are not limited to geometric Brownian motion but extend to more realistic asset price models. They underscore fundamental challenges inherent in time series datasets [37].

Moreover, financial returns exhibit nonstationarity [59], leptokurtosis [78], and other stylized facts [95]. Nonstationarity implies that resource allocation must be dynamic rather than static, requiring periodic adjustments. The leptokurtic nature of returns necessitates robust statistical procedures for model calibration.

1.3 The Probabilistic Sharpe Ratio

Modern economies largely operate within the Markowitz framework [74], focusing on optimizing the Sharpe ratio or similar metrics. This framework emphasizes optimization processes, assuming known values for μ and σ, and often overlooks the significant challenges in estimating these parameters. Furthermore, it treats μ and σ as fixed, ignoring the inherent nonstationarity in financial markets. Addressing these limitations is crucial for both static and dynamic portfolio allocation, calling for a re-evaluation of traditional approaches.

In [13], Bailey and de Prado ask the question, what is the probability that an estimated Sharpe ratio exceeds a given threshold in the presence of non-normal returns. They show that this new uncertainty-adjusted investment skill metric, called **probabilistic Sharpe ratio (PSR)**, has a number of important applications. For example, it allows to establish the track record length needed for rejecting the hypothesis that a measured Sharpe ratio is below a certain threshold with a given confidence level.

Let $\gamma_3 = \frac{\mathbb{E}[R-\mu]^3}{\sigma^3}$ be skewness and $\gamma_4 = \frac{\mathbb{E}[R-\mu]^4}{\sigma^4}$ be kurtosis. Let SR be the true Sharpe ratio and $\hat{SR}$ the empirically observed, estimated Sharpe ratio. Even without normality and under mild assumptions of returns, Mertens [76] concludes that, the estimated Sharpe ratio will follow a normal distribution with parameters

$$\sqrt{n}(\hat{SR} - SR) \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, 1 - \gamma_3 SR + \frac{\gamma_4 - 1}{4} SR^2\right).$$

The proof for the above claim is realised using the Delta method (see Appendix A.2), a standard technique when working with asymptotically efficient estimators.

The authors of [13] use this result to derive some confidence bands for the estimate $\hat{SR}$. In particular, they deduce that the estimated standard deviation of $\hat{SR}$ is

$$\hat{\sigma}_{\hat{SR}} = \sqrt{\frac{1 - \gamma_3 \hat{SR} + \frac{\hat{\gamma}_4 - 1}{4} \hat{SR}^2}{n-1}},$$

where $n-1$ is due to Bessel's correction. The true value SR is bounded by the estimate $\widehat{SR}$ with a significance level α

$$\mathbb{P}[SR \in (\hat{SR} - Z_{\alpha/2}\hat{\sigma}_{\hat{SR}}/\sqrt{n}, \hat{SR} + Z_{\alpha/2}\hat{\sigma}_{\hat{SR}}/\sqrt{n})] = 1 - \alpha. \tag{1.3.1}$$

While the authors of the paper didn't introduce the $\sqrt{n}$ term in the confidence interval, that quantity is crucial as it is a direct consequence of estimators derived from the Delta

method. To be more specific, Bessel's correction term is for estimating the standard deviation. The additional $\sqrt{n}$ comes from the scaling via the Delta method. If the Sharpe ratio estimator was normally distributed then the removal of the additional factor would be indeed the natural progression of deriving a confidence interval, but scaling is necessary to validate the asymptotic properties of $\hat{SR}$.

The consideration of the skewness and kurtosis adds more information to estimating the true Sharpe ratio. However, there are some drawbacks. In accordance with the authors' opinion who previously studied this estimator, a negative ratio may impact the calibration process. We shall discuss this issue when we introduce the `PSR-UCB` method in the following chapter.

Chapter 2

Multi-Armed Bandits

The multi-armed bandit problem, a key concept for developing online sequential decision-making strategies, has been explored since the early 1930s by Thompson [104]. Later on, Robbins [87] and Bush and Mosteller [22] popularised the concept and conducted further analysis on the subject. The core principle of bandit learning revolves around balancing the acquisition of new information with optimizing rewards based on current knowledge. This principle, known as the exploration-exploitation trade-off in reinforcement learning, naturally relates to the sequential decision-making process involved in portfolio selection. Hoffman utilized a multi-armed bandit approach to construct a portfolio of acquisition functions within the context of Bayesian optimization [51].

There are many reasons to care about bandit problems. Lattimore and Szepesvári [65] talk about the importance of decision-making with uncertainty and how that is a challenge we all face, with bandits providing a simple model of this dilemma. Bandit problems also have practical applications. Major tech companies use bandit algorithms for configuring web interfaces, where applications include news recommendation, dynamic pricing and ad placement. A bandit algorithm plays a role in Monte Carlo Tree Search, an algorithm made famous by the recent success of AlphaGo [86, 94, 101].

However, traditional multi-armed bandit models have very restrictive assumptions regarding the environment that limit the performance of the algorithms through generalisations. Most bandit problems assume stationarity and no correlation between the rewards, although in the finance context that is rarely the case.

We organise this chapter as follows: in Section 2.1 we provide a general overview of bandit algorithms and the evolution of research in their field. In Sections 2.2 and 2.3 introduce the core notions of a multi-armed bandit setting that set the foundations of all further developments. They are essential to understanding portfolio selection from a bandit perspective. In Section 2.4 we discuss the leap from a single-action per round setting to weighted bandits and analyse how the reward estimators change to the new conditions. Lastly, Section 2.5 explores a benchmark bandit model that takes advantage of collinearity.

2.1 Overview of Bandit Algorithms

A bandit algorithm model situations where a decision-maker must choose between multiple options (or "arms") with uncertain rewards. The goal is to maximize cumulative rewards over time while balancing the trade-off between exploring new options and exploiting known ones. We distinguish two core models of bandits:

- **Stochastic**: In a stochastic bandit setting, each arm is associated with a fixed but unknown probability distribution of rewards. The challenge lies in identifying the arms with the highest expected rewards while minimizing the loss due to suboptimal choices. The stationary stochastic bandit problem has been thoroughly studied since the pioneering work of Lai and Robbins [64], which laid the foundation for many subsequent algorithms. Their novel method at the time is still used as the main framework for stochastic bandits, namely the `UCB` algorithms. They are deterministic algorithms aimed to achieve logarithmic regret, and thus they suffer from choosing sub-optimal arms especially in a non-stationary scenario.

- **Adversarial**: These bandits assume that the reward distributions can change in arbitrary ways, possibly in response to the player's actions. This model is more appropriate for scenarios where the environment is dynamic or controlled by an adversary. Algorithms like `Exp3` and `Exp4` [10] and its variants are designed to perform well under these conditions, providing robust performance even when the reward structure is unpredictable. Very recent results improved the regret bounds of this category of bandits [24].

The performance of such algorithm is called **regret**, which measures the deficit suffered by the learner relative to the optimal policy and it often has different definitions relative to the method used. A **policy** dictates how we choose our actions at each step and represents the conditional probability of selection given the current state

$$\mathbb{P}(A_t = a) = \pi_t(a|A_1, X_1, ..., A_{t-1}, X_{t-1}). \tag{2.1.1}$$

There are several types of regret and their names heavily vary in literature. Thus, we need to be very specific about what kind of regret achieves a certain bound. A good survey is done in [72], although it doesn't cover all the algorithms we present in this study. We shall explore the definitions of regret as we go through presenting the general framework.

This measure is also dependent on how much information (feedback) we choose to receive from the environment. Thus, we identify three different types of feedback:

- Full-Bandit: Only information about the selected arm is available.

- Semi-Bandit: The knowledge is limited to a subset of selected arms.

- Full-Information: All rewards are known at each time step.

Portfolio selection is a natural application of bandit algorithms, particularly in the financial markets where the goal is to allocate capital across different assets to maximize returns while managing risk. Before describing what kind of bandit algorithms already exist and are adapted for a Portfolio Optimisation setting, we first enumerate the main issues which separate classical bandits and a finance-related scenario:

1. We can diversify by choosing multiple actions at time t: $\{a_1, \ldots, a_{k_t}\} \subseteq \mathcal{A}$.

2. There may be significant correlations among the actions.

3. The data we work with is usually a non-stationary time series.

Notice that the way we choose to tackle this problem sits on a boundary between bandits and online learning. We will analyse problems 1 and 2 in this chapter, leaving the final point 3 for the subsequent one, where we discuss about changepoint detection.

The main direction to follow when adapting bandit algorithms to a portfolio choice problem is to take into account performance metrics, such as the Sharpe ratio we introduce in Chapter 1. Moreover, this ratio is not the only information one can have about financial data. In fact, there are many metrics to look at such as the *Calmar, Sterling or Sortino ratios*. Thus, we have a context here we can exploit and we will discuss further on this issue in the following sections.

2.2 Stochastic Bandits

We now proceed with the necessary definitions of the bandit framework, inspired from Lattimore and Szepesvári's reference work [65].

Definition 2.2.1 (Section 4.1 [65])**.** A **stochastic bandit** is a collection of distributions $\nu = (P_a : a \in \mathcal{A})$, where $\mathcal{A} = \{1, 2, ..., k\}$ is the available set of actions.

We (the learner) interact with the environment (the market) sequentially, selecting at each round (time step) $t \in \{1, 2, 3..., n\}$ an action $A_t \in \mathcal{A}$. Each action generates a reward $X_t \in \mathbb{R}$ from distribution P_{A_t}. Thus, we generate a sequence of rewards and actions $A_1, X_1, A_2, X_2, ..., A_n, X_n$. The keyword in this scenario in this setting is **online learning** which allows the learner to improve the selection using information only up to the current round to select an action for the next.

We can see immediately the challenges of single-action selection and how one may inaccurately provide a strategy using the wrong assumptions. One example is the fact that in a canonical bandit setting, one does not observe the rewards of the dismissed actions, whereas in our portfolio choice problem we have all the PnLs from all the equity curves. However, the computational cost to consider all asset combinations at every time can become exponential [55].

For the first two sections we will consider the classic bandit scheme where we select only one action and we take into account the information only from the observed selection (full-bandit feedback), thus our weight vector is determined by W_t, for which there is exactly one $j \in \{1, 2, ..., k\}$ such that $w_{tj} = 1$ and $w_{ti} = 0$ for all $i \in \{1, 2, ..., k\} - \{j\}$. The reward X_t at round t is then described, according to the notation at (1.1.1), by

$$X_t = \sum_{i=1}^{k} w_{t-1,j}(M_{tj} - M_{t-1,j}) = \sum_{i=1}^{k} \mathbb{1}_{\{A_{t-1}=j\}}(M_{tj} - M_{t-1,j})$$

This definition of reward is consistent because we don't consider transaction costs or market impact so we can focus solely on maximising the rewards. So we can treat every round as exiting the previous position and taking the PnL. As mentioned earlier, our objective is to maximise the total reward

$$S_n = \sum_{t=1}^{n} X_t$$

We define the regret for stochastic bandits as

$$R_n(\pi, \nu) := n\mu^*(\nu) - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right], \tag{2.2.1}$$

for which $\mu_a(\nu) := \int_{-\infty}^{\infty} x dP_a(x)$ and $\mu^*(\nu) := \max_{a \in \mathcal{A}} \mu_a(\nu)$.

A major assumption regarding stochastic bandits lies on the behaviour of the distribution of rewards, as it's very hard to control the bounds of the expected value for variables which are not subgaussian, a notion which we will introduce now.

Definition 2.2.2. (Subgaussian random variable) A random variable X is called **subgaussian** if

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right), \text{ for any } \lambda \in \mathbb{R}.$$

Corollary 2.2.3. *(Example 5.6 [65]) A normal random variable with mean* 0 *and variance* σ^2 *is* σ*-subgaussian.*

As Definition (2.2.1) is purely theoretical, we can't use it to analyze convergence, since we don't have any information about the rewards, aside from the subgaussianity assumption. Thus, using the Regret Decomposition Lemma (Lemma 4.5 [65]), we can break it down into parts as

$$R_n := \sum_a \Delta_a \mathbb{E}[T_a(n)], \tag{2.2.2}$$

where $T_a(n)$ is the random variable that governs how many times we choose action a in the first n rounds and $\Delta_a := \mu^* - \mu_a(n)$ is called the **suboptimality gap** for the same arm.

2.2.1 Upper Confidence Bound

The nature of the stochastic bandit problem can be interpreted from a hypothesis testing perspective. Given an estimation of the mean reward, what is the probability that we're the true optimal value by some margin U? The quantity U is called the **Upper Confidence Bound** (UCB) and the first bandit algorithm we aim to explore shares the name with the method itself. This technique was first analysed by Lai and Robbins [64] and most of the modifications and adaptations for a specific task are closely related to the same framework. Auer [9] popularised this concept with what is now called the `UCB1` algorithm. This method is grounded in the principle of optimism under uncertainty, which suggests selecting actions under the assumption that the environment behaves in the most favorable way within the bounds of plausibility.

If we consider $\hat{\mu} = \frac{1}{n} \sum_{t=1}^{n} X_t$ to be the mean estimator for our rewards, we can denote $T_i(t-1)$ as the number of times action i was selected until time $t-1$. Actions are selected based on minimising this upper theoretical limit to achieve optimal regret.

Then, the Upper Confidence Bound is defined as

$$\text{UCB}_i(t-1, \delta) = \begin{cases} \infty & , T_i(t-1) = 0 \\ \hat{\mu}_i(t-1) + \sqrt{\frac{2\log(1/\delta)}{T_i(t-1)}} & , \text{ otherwise} \end{cases}$$

The infinite value of overlooked arms can be easily avoided by a delayed start, with an initial pure exploration phase where we observe each arm at least once and then we can start the actual implementation of the algorithm. However, if we delay the start we may hinder (or boost) the algorithm's performance. Due to its deterministic nature, determining how much we can explore before starting the exploitation phase may force the algorithm to commit to another arm. This problem wouldn't pose at an asymptotic level, due to the regret bound theorems we shall uncover, so a problem wouldn't exist for very large values of n. However, in a finance-related context we cannot have the data for millions of trading days and, as we shall see in Chapter 3, increasing the sample frequency

Algorithm 1: Phased Upper Confidence Bound (`UCB`)

Input : k, γ and δ

for $t \in 1, 2, ..., \lfloor \gamma \rfloor \cdot n$ **do**
  Choose action $A_t = t \bmod k$
  Observe reward X_t and update UCB_t
for $t \in \lfloor \gamma \rfloor \cdot n + 1, \lfloor \gamma \rfloor \cdot n + 2, ..., n$ **do**
  Choose action $A_t = \text{argmax}_i \text{UCB}_i(t-1, \delta)$
  Observe reward X_t and update UCBs

will not improve parameter estimation. This is what we aim to fix in this section by proposing a new algorithm.

The following theorem describes the regret of the UCB algorithm:

Theorem 2.2.4. *(Theorem 7.2 [65]) If $\delta = \frac{1}{n^2}$, then the regret of Algorithm 1 is bounded by*

$$R_n \leq 8\sqrt{nk \log(n)} + 3 \sum_{i=1}^{k} \Delta_i$$

Algorithm 1 can be extended to be asymptotically optimal and achieve a better regret ([65], Theorem 8.1), by replacing the function under the logarithm in the confidence bound. Among the more notable and relevant improvements we find `MOSS` [8] and Thompson Sampling [104]. The latter, originally designed to solve the stochastic bandit problem, was later deemed more flexible and suitable for a wider range of settings [91], although the asymptotic optimality was guaranteed only for the stochastic problem. However, for a financial context where we need to maximise returns we can take advantage of more information to make more informed decisions.

2.2.2 PSR-UCB

We propose a novel UCB algorithm that outperforms the known benchmarks. The key idea comes from replacing the reward estimation within the UCB definition with the Sharpe ratio. This idea was discussed in earlier works, but within different settings.

Shen first offered intuition on this matter by constructing the "Orthogonal Bandit Portfolios" (OBPs) [99], which use the UCB for selecting the best performing portfolio from a set designed by the authors. His work was further continued in [83] by implementing the PSR in the OBP framework. Even though the empirical results are promising, they don't capture full-bandit feedback, because the learner can still see all returns from the arms. Moreover, all previously mentioned paper in this section present only empirical results, with no theoretical guarantees. The closest improvements on single-action selection with full-bandit feedback was achieved recently in [61, 23] with theoretical guarantees on regret for optimising the Sharpe ratio. Even though the results are promising, the authors provide regret bounds for the specific measures that used in their definition of the upper confidence bound. More specifically, the rewards used in the regret analysis are the measures themselves (Sharpe ratio, variance), whereas the rewards should still be kept as being the returns. We propose a different approach while studying the regret where the rewards represent the returns and not the ratios themselves.

If we take $\mu(a) := \mathbb{E}_t[X_t \mid A_t = a]$, we then require

$$\mu(a) \leq \hat{\mu}_t(a) + U_t(a)$$

with high probability, where $U_t(a)$ is the upper bound. Put differently, we need

$$\mathbb{P}[\mu(a) > \hat{\mu}_t(a) + U_t(a)] \leq p$$

for some $p \in [0, 1]$. In our case, we pick $\mu(a) = SR(a)$, $\hat{\mu}_t(a) = \hat{SR}_t(a)$. In order to determine $U_t(a)$, we solve the equation

$$\mathbb{P}[\mu(a) > \hat{\mu}_t(a) + U_t(a)] = p$$

for $U_t(a)$ in terms of p. But

$$\mathbb{P}[\mu(a) > \hat{\mu}_t(a) + U_t(a)] = \mathbb{P}[SR(a) > \hat{SR}_t(a) + U_t(a)]$$

and from (1.3.1) we have

$$\mathbb{P}[SR(a) > \hat{SR}_t(a) + Z_p \frac{\hat{\sigma}_{\hat{SR}_t(a)}}{\sqrt{T_a(t)}}] = p, \tag{2.2.3}$$

and hence

$$U_t(a) = Z_p \frac{\hat{\sigma}_{\hat{SR}_t(a)}}{\sqrt{T_a(t)}},$$

where $Z_p = \Phi^{-1}(1-p)$, $\Phi(\cdot)$ is the quantile function (probit) of the standard normal distribution, and

$$\hat{\sigma}_{\hat{SR}_t(a)} = \sqrt{\frac{1 - \hat{\gamma}_3(a)\hat{SR}_t(a) + \frac{\hat{\gamma}_4(a)-1}{4}\hat{SR}_t^2(a)}{T_a(t) - 1}},$$

where $\hat{SR}_t(a)$ is the estimated Sharpe ratio based on the observed rewards on arm a at time t.

Thus, we define the PSR bound as

Definition 2.2.5. (PSR-UCB bound)

$$\text{UCB}_\text{i}^{\text{PSR}}(t-1, p) = \begin{cases} \infty & , T_i(t-1) = 0 \\ \hat{SR}_i(t-1) + Z_p \frac{\hat{\sigma}_{\hat{SR}_t(a)}}{\sqrt{T_i(t-1)}} & , \text{ otherwise} \end{cases} \tag{2.2.4}$$

Notice that in the actual confidence bound term there is the Sharpe ratio estimator for the observed rewards on each arm instead of deriving a quantity from a Chernoff-Cramer inequality (Theorem 5.3 [65]). In fact, we get both the lower and upper bounds for our algorithm since the Sharpe ratio estimator is a normal random variable.

Remark 2.2.6. The returns do not have to be normally distributed in order for the algorithm to perform well [13].

As the SR estimator follows a normal distribution, it is naturally a subgaussian random variable, and thus a regret bound can instantly be derived if we also use the ratios as rewards. For considering the returns, we provide the following result.

Theorem 2.2.7. *(Regret) If the Sharpe ratios of the arms of a σ-subgaussian stochastic bandit which follows Algorithm 2 are positive and if the optimal arm has the highest ratio, then the regret is upper bounded by*

$$R(n) \leq 2 \max_{i \in [k]} \sigma_{SR_i}^4 \sum_{\Delta_i > 0} \frac{\log(n)}{\Delta_i} + 3 \sum_{i=1}^{k} \Delta_i, \text{ for large values of } n.$$

Algorithm 2: `PSR-UCB`

Input: k, n, γ

Parameters: $T_i(0), \hat{SR}_i(0), \hat{\gamma_3}^i(0), \hat{\gamma_4}^i(0)$

for $i = 1, 2, \ldots, \lfloor \gamma \rfloor \cdot n$ **do**
  Play arm $i \bmod k$
  Update $T_i(t), \hat{SR}_i(t), \hat{\gamma_3}^i(t), \hat{\gamma_4}^i(t)$
Calculate $\text{UCB}_i^{PSR}(\lfloor \gamma \rfloor \cdot n)$ for each $i = 1, 2, \ldots, k$ from (2.2.4)
for $t = \lfloor \gamma \rfloor \cdot n + 1, \lfloor \gamma \rfloor \cdot n + 2, \ldots, n$ **do**
  Play arm $a_i = \arg\max_i \text{UCB}_i^{PSR}(t)$
  Update $T_i(t), \hat{SR}_i(t), \hat{\gamma_3}^i(t), \hat{\gamma_4}^i(t)$
  Calculate $\text{UCB}_i^{PSR}(t)$ as in (2.2.4)

Remark 2.2.8. The regret guarantee lies on a few assumptions which may not reflect the true quality of the `PSR-UCB` algorithm. Bailey and de Prado talk about the $\hat{SR}$ estimator issues, namely regarding the skewness and kurtosis values [13]. More specifically, the probability that $\hat{SR}$ is greater than the true Sharpe ratio will increase with positive skewness, but will decrease with thicker tails.

The proof of Theorem (2.2.7) can be found in Appendix A.3. To the best of our knowledge, the only result which provided theoretical guarantees for using the Sharpe ratio in an UCB-type setting comes from [61]. However, the authors provide a regret bound for which the suboptimality gaps (and implicitly the rewards) are the actual ratios themselves, whereas we provide a regret bound for the actual returns of the equity curves.

While the optimism principle is often effective, it becomes unreliable in more complex scenarios. For instance, the whole framework depends on uncertainty, whereas in a real-world trading situation one has all the information available in the present, as opposed to the classical bandit setting. Thus, in the canonical setting, it wouldn't be feasible to apply an UCB-type algorithm to a full-information feedback setting. However, we shall see in the following sections that we can change this perspective by removing the one-to-one relationship between the arms and the rewards (actions). More specifically, we will analyse a continuous space of rewards, represented by a weighting scheme, while still having a discrete set of arms. Thus, it becomes virtually impossible to look at all rewards and uncertainty emerges again.

2.3 Adversarial Bandits

A key difference between classical bandits and our portfolio choice problem lies within the reward distribution, which is assumed to be fixed throughout the whole interaction with the environment. **Adversarial Bandits** are more clever, on the basis that the environment will respond to the chosen actions by selecting an unfavourable distribution for rewards such that the learner is at a constant disadvantage. Thus, if one uses a predictable tactic, such as UCB, a clever adversary (that is, in our case, another market participant) will exploit this knowledge and counteract it. This essentially translates to a non-stationary distribution of rewards, which the stochastic bandits' asymptotic optimality cannot handle.

Definition 2.3.1. (Section 11.1 [65]). A k-armed **adversarial bandit** is an arbitrary sequence of reward vectors $(x_t)_{0\leq t\leq n}$, for which $x_t \in \mathcal{M}_{k\times 1}(\mathbb{R})$.

The performance of such a policy is measured by **worst-case regret**, represented by the incurred loss by pulling the worst arm at each available step along the time horizon. First, if we take the **expected regret** as

$$R_n(\pi, x) = \max_{1\leq i\leq k} \sum_{t=1}^{n} x_{ti} - \mathbb{E}\left[\sum_{t=1}^{n} x_{tA_t}\right],$$

then the worst-case regret is

$$R_n^*(x) = \sup_x R_n(\pi, x),$$

where the worst-case scenario isn't represented by the maximum mean one can achieve by committing to a certain arm, but instead it gives the worst reward for each selection. Lattimore and Szepesvári argue that deterministic policies, such as `UCB`, will not achieve a sub-linear worst case regret (Section 11.1 [65]).

They argue that the best way to avoid suboptimal choices is to randomise the policy, since that would imply our adversary has less information about our decision. They conclude that good randomised policies can be found, thus our adversary cannot avoid the inevitable.

There is extensive literature on bandits within an adversarial scenario. The starting point is the `Exp3` algorithm, introduced by Auer, Cesa-Bianchi, Freund and Schapire [10] which achieves $\mathcal{O}(\sqrt{n})$ regret. The main idea is to map the decision set to a probability measure which is then updated via an importance sampling estimator, discussed in the next section. The framework was gradually improved in the following years [8, 21]. The main framework for adversarial saw more connections with online convex optimisation, as these algorithms share similarities with the Mirror Descent method, introduced in 1983 by Nemirovsky and Yudin [80]. Thus, the same authors who developed `Exp3` also introduced `Exp4`, which stands for "exponential-weight algorithm for exploration and exploitation with experts'. We will discuss about the relationship between `Exp3` and `Exp4` and continue by describing the latter, as it is closer to solving our problem at hand.

2.3.1 Reward Estimation

The main concept of the exponentially-weighted bandits are **importance weighted** estimators, defined as

$$\hat{X}_{ti} = \frac{\mathbf{1}_{\{A_t=i\}} X_t}{P_{ti}} \tag{2.3.1}$$

If we take $\mathrm{E}_t[\cdot]$ to be the conditional expectation given the history up to time t, these estimators are unbiased for the rewards (Page 151 of [65]) , i.e.

$$\mathrm{E}_{t-1}[\hat{X}_{ti}] = x_{ti}$$

Let $\hat{S}_{ti} = \sum_{s=1}^{t} \hat{X}_{si}$ be the total estimated reward up to time t, according to (2.3.1). Then, we consider that higher-reward arms should be played more often, and thus we map $\hat{S}_{ti}$ to probabilities using **exponential weighting** in the following way

$$P_{ti} = \frac{\exp(\eta \hat{S}_{t-1,i})}{\sum_{j=1}^{k} \exp(\eta \hat{S}_{t-1,j})}.$$

We take η as the learning rate, which governs how often we commit to the largest estimated reward. If the rate is smaller, the probabilities are more uniform, allowing for more exploration. These notions are important because in the later sections we will extend the one-action per round scenario to multiple weights at each stage.

An equivalent estimator for the rewards, which we will use for the prediction with experts algorithm is defined by

$$\hat{X}_{ti} = 1 - \frac{\mathbf{1}_{\{A_t=i\}}(1 - X_t)}{P_{ti}}.$$

2.3.2 Expert advice

The key improvement of `Exp4` is the use of expert advice, which is represented by a set of possible policies to sample an action from. Suppose we have L experts available to sample an action. Then, we define $E^{(t)}_{L_t}$ as the probability distribution of selecting an action taking using expert's L_t advice. The way we choose our expert at round t is governed by the probability distribution Q_t, which is initially a discrete uniform taking values between 1 and L.

Remark 2.3.2. If we consider $Q_t = (Q_{t1}, Q_{t2}, ..., Q_{tL})$ and $E^{(t)} \in \mathcal{M}_{L\times k}(\mathbb{R})$, notice that

$$P_t = Q_t E^{(t)} \in \mathcal{M}_{1\times k}(\mathbb{R}),$$

for which each row vector has the sum equal to 1.

After an action is chosen, a reward is estimated using an importance-weighted estimator. The distribution of choosing an expert is then adjusted using those estimates via exponential weighting.

The following theorem characterizes the regret of the Exp4 algorithm:

Theorem 2.3.3. *(Theorem 11.1 [65]) Let $\gamma = 0$ and $\eta = \sqrt{2\log(M)/(nk)}$ and R_n the expected regret of Exp4 defined in Algorithm 3. Then,*

$$R_n \leq \sqrt{2nk\log(M)} \tag{2.3.2}$$

Algorithm 3: Exp4 [10]

Input : n, k, M, η, γ

Set $Q_1 = (1/M, ..., 1/M)$

for $t \in 1, 2, ..., n$ **do**

- Receive advice $E^{(t)}$
- Choose action $A_t \sim P_t$, where $P_t = Q_t E^{(t)}$
- Receive reward $X_t = x_{tA_t}$
- Estimate action rewards: $\hat{X}_{ti} = 1 - \frac{\mathbb{1}\{A_t = i\}}{P_{ti} + \gamma}(1 - X_t)$
- Propagate rewards to experts: $\tilde{X}_t = E^{(t)} \hat{X}_t$
- Update distribution Q_t using exponential weighting:

$$Q_{t+1,i} = \frac{\exp(\eta \tilde{X}_{ti}) Q_{ti}}{\sum_j \exp(\eta \tilde{X}_{tj}) Q_{tj}}$$

Due to the randomised nature of the policies, there are still open questions regarding the regret bounds that could be achieved. The upper bound was later improved [57], although it still doesn't match with the lower bound, which saw more attention in literature. The most recent improvement was developed by Cesa-Bianchi [24] based on the results of Chen, He, and Zhang [26] on a restricted expert setting using feedback graphs. The authors conjecture the bound can be extended to any kind of expert feedback, although this remains an open topic.

2.4 Adapting Bandits to the Portfolio Choice Problem

While standard bandit learning focuses on identifying the single best option, portfolio selection often involves choosing a combination of multiple assets. From the more general setting of adversaries, another branch of this class was developed, namely **contextual** bandits [28, 30], which have context-adapted policies and so one can improve regret by narrowing down the available options when making a selection. Another sub-category of bandits is represented by the **combinatorial** algorithms, which have been developed to handle the finite selection of multiple arms [25].

Several new concepts arose as a result of researching contextual and combinatorial bandits, such as linear bandits (`linUCB` algorithm [1, 58]), which represented one of the first attempts to separate the arm and the decision to allow for more flexibility. Usually, a linear function is assigned to describe each reward possible using a set of feature as arguments. Both combinatorial and contextual bandits typically make binary decisions regarding which arms to choose and allocate investments equally among them [27]. In contrast, the fundamental challenge in portfolio selection lies in determining the optimal distribution of weights amongst all the proposed assets.

A classic bandit algorithm will map each arm with its rewards and nothing more. If we remove this restriction and consider a weighting scheme , we allow for full flexibility of improving our PnL and not limit ourselves to picking one action at a time.

In fact, using the definition of weights presented in the previous sections we can distinguish between each type of feedback based on the constraints of W. Thus, we can take

the weight vector [38] at time t to be

$$W_t \in \mathcal{D} := \left\{ (w_{ti})_{i\in\overline{1,k}} \,|\, w_{ti} \in [0,1], \text{ for any } 0 \le i \le k, \sum_{i=1}^{k} w_{ti} = 1 \right\} \tag{2.4.1}$$

The flexibility of considering arms and actions separately with the previous definition of weights is evident, and one can characterize the algorithm used only by restricting the class of weights used.

The main challenge of the total reward quantity comes from the limited amount of information regarding the reward distributions. In general, one may look for improving the selection algorithm looking elsewhere and adjusting the existing framework to a finance-related scenario. For instance, we know our data is a time series and, furthermore, we know the rewards of different equity curves are correlated. In addition to this, we shouldn't resort only to choosing only one action per round. We will instead assign weights for each arm and try to maximise the reward

$$S_n = \sum_{t=1}^{n} \sum_{j=1}^{k} w_{tj} X_{tj}, \tag{2.4.2}$$

If we take W^* to be the optimal weights matrix and $\mu_{tj} = \mathbb{E}[X_{tj}]$, then we can define our expected regret as

$$R_n := \sum_{t=1}^{n} \sum_{j=1}^{k} (w^*_{tj}\mu_{tj} - w_{tj}\mu_{tj}) = \sum_{t=1}^{n} \sum_{j=1}^{k} (w^*_{tj} - w_{tj})\mu_{tj}.$$

Remark 2.4.1. The optimal weights matrix consists of only unit vectors from the canonical base of the coordinate vector space $\mathbb{R}^k$ because in the best case scenario we allocate a full measure to the highest mean achieved at every round.

The previous remark points out that the expected regret is only marginally different than the one defined at (2.2.1). If we take $\mu^*_t = \max\limits_{0\le j\le k} \mu_{tj}$ and $a^*_t = \arg\max\limits_{0\le j\le k} \mu_{tj}$ we can provide a better representation of the expected regret as

$$R_n = \sum_{t=1}^{n} \left((1 - w_{ta^*_t})\mu^*_t - \sum_{j=1}^{k} \mathbf{1}_{\{j\neq a^*_t\}} w_{tj}\mu_{tj} \right)$$

2.5 Semi-Bandit and Full-Information Algorithms

In this section we discuss how to improve upon using more information from the environment and try to give some insight on the different approach we take as opposed to Full-Bandit feedback. Furthermore, we extend the PSR framework described in 1.3 and propose an expert selection scheme for portfolio management.

2.5.1 UCB in continuous space

It's a good improvement to have adjusted for multiple choices, although a problem still persists. In the previous definition of regret there is no correlation term, although the returns are usually correlated. On this topic there are a few results that focus on combinatorial bandits with covariance [85, 35], which inspired the more recent paper on "Continuous mean-covariance bandits" [38] to introduce several new algorithms with novel regret guarantees. Taking the mean vector of the rewards at time t as $\theta_t \in \mathcal{M}_{k\times 1}(\mathbb{R})$ and the positive

semi-definite covariance matrix Σ_t with $\Sigma_t^{ii} \leq 1$, for any $i \in \overline{1,k}$, the authors define a reward-risk function as

$$f(W_t) = W_t^T x^* - \rho W_t^T \Sigma^* W_t. \tag{2.5.1}$$

The optimal action at round t is then defined as $W_t^* = \arg\max_{W_t \in \mathcal{D}} f(W_t)$.

Accounting for covariance is essential as it translates to managing the risk, and this issue within a bandit framework was tackled multiple times. The intuition is quite elementary, as one adjust the mean estimators using a covariance matrix of the returns and thus re-adjusting the Sharpe ratio. This is how the "Orthogonal Bandits" technique, which we mentioned in Section 2.2 was developed [99], and then the weights were regularised in [61], although a slightly different approach was taken, where the authors split the portfolio in two parts and compute the UCB values sepparately, optimising the balance of the split afterwards.

There is, however, a key factor to consider here. If one has access to all rewards (full-information feedback), then there is no sense to look at an UCB algorithm, simply because there is no uncertainty left to explore, since all information is available. While it is true that the authors built a different decision space than the one represented by the arms, full-feedback was still available. Nevertheless, besides the orthogonal bandit approach there are several other works which mention UCB algorithms that account for covariance ([53, 98]). Why is that the case? Well, mainly because there is a proxy used for the reward. It could be the Sharpe ratio, or it could be a different quantity that aims to create order amongst the arms.

If we consider a weighting scheme on the final PnL, then the whole reward space is continuous, making it virtually impossible for someone to ever have full-information feedback. It's a matter of perspective, which is heavily discussed in [38], but to the best of our knowledge it wasn't articulated in a manner that would justify why an UCB-type algorithm would suffice, because confidence bound algorithms work best with a finite set of decisions. Thus, a finite decision space can be created, which will reduce the accuracy of a fully continuous model, but it will, on the other hand, reduce computational time.

The results of [38] are most promising in this regard, as the covariance is taken to minimising the regret and new estimators are proposed for the equity curve means. The authors of the mentioned paper propose an algorithm for each type of feedback, each with its own slight modifications and improve the theoretical regret bounds, initially developed in [85].

We will showcase in this paper the algorithm proposed in [99], where the authors present the previously mentioned Orthogonal Bandit approach which takes into account correlation between rewards and further improve that to more realistic model, although the authors don't provide a theoretical guarantee.

Their main idea is to decompose the covariance matrix and build orthogonal portfolios based on its eigenvalues. The respective values are ordered, where the highest one are considered the most significant. Then, a split is made between each bandit portfolio (one per eigenvalue) and only one value is selected from each subset via UCB.

For Algorithm 5, we have the following 2 equations for steps 8 and 9, respectively:

$$\theta_k^* = \arg\min_{\theta_k} = \frac{\tilde{\lambda}_{k,i_k^*}}{\tilde{\lambda}_{k,i_k^*} + \tilde{\lambda}_{k,j_k^*}}$$
$$\omega_k = (1 - \theta_k^*)\tilde{\mathbf{H}}_{k,i_k^*} + \theta_k^* \tilde{\mathbf{H}}_{k,j_k^*},$$

where the $\tilde{\lambda}_k$ parameters come from the diagonal matrix $\tilde{\Lambda}_k$.

2.5.2 Predicting with Expert Advice

Since in a real-world trading environment we usually face a full-information scenario, we will look at a tailored Exp4-variant that has access to all information and it has expert advice suited for a finance context, making it a combination between `Exp4` and `Hedge` [43].

Thus, for the first issue we aim to choose conventional weighting schemes for our experts. The resulting weights are then clipped, filtering only the positive values and then normalised to turn them into probabilities. Amongst our selection, we've picked several weighting schemes such as Sharpe ratio, Calmar ratio and Clever weights, which we will showcase below.

Firstly, if we have a set of returns $X_1, \ldots, X_n$ with mean μ_i, we define the Calmar ratio at time t as

$$CR_t := \frac{\mu_t}{DR_t},$$

where DR is the drawdown, i.e. $\max_{i\in[t]}(X_i) - X_t$.

For the second issue we know that the Exp4 algorithm starts with selecting only one action, and we've mentioned that for a combinatorial setting it's computationally infeasible to consider all possible asset combinations since that would scale exponentially.

The algorithm is more closely aligned with the philosphy behind the `Hedge` method [43, 69], which is suited for predictions with expert advice.

Algorithm 4: Clever Weighted Equity Curve (ClW)

Input: Equity curves matrix M, mean adjustment factor α_{mean}, standard deviation adjustment factor α_{std}

Output: Adjusted weights

$k, n \leftarrow$ shape of M;
`diffs` $\leftarrow$ Calculate the increments of M on axis 1;
`drawdowns` $\leftarrow$ Calculate the drawdowns of M;
`mean_diffs` $\leftarrow$ Calculate the mean of `diffs`;
`std_diffs` $\leftarrow$ Calculate the standard deviation of `diffs`;
`weights` $\leftarrow \frac{\texttt{mean\_diffs}}{(1+\texttt{drawdowns})\cdot(1+\texttt{std\_diffs})}$;
`weights` $\leftarrow \max(\texttt{weights}, 0)$;
`weights` $\leftarrow \text{round}(\texttt{weights}, 4)$;
`sum_of_weights` $\leftarrow \text{sum}(\texttt{weights})$;
`mean_weights` $\leftarrow (1 - \alpha_{\text{mean}}) \cdot \texttt{weights} + \alpha_{\text{mean}} \cdot \text{mean}(\texttt{weights})$;
`std_weights` $\leftarrow (1 - \alpha_{\text{std}}) \cdot \texttt{weights} + \alpha_{\text{std}} \cdot \text{std}(\texttt{weights})$;
`weights` $\leftarrow \frac{\texttt{mean\_weights}}{1+\texttt{std\_weights}}$;
return `weights`;

Algorithm 5: Orthogonal Bandit Portfolio (OBP) [99]

Input : $m, n, l, \Delta t, \mathbf{R}_k, \tau$

for $k = 1$ **to** m **do**

- Estimate the average return $\mathbb{E}[\mathbf{R}_k]$ and covariance matrix of asset returns Σ_k using $\{\mathbf{R}_{-\tau+k}, \ldots, \mathbf{R}_{k-1}\}$.
- Implement the principal component decomposition: $\Sigma_k = \mathbf{H}_k \Lambda_k \mathbf{H}_k^\top$.
- Compute the renormalised similarity matrices and eigenvalues: $\tilde{\Sigma}_k = \tilde{\mathbf{H}}_k \Sigma_k \tilde{\mathbf{H}}_k^\top = \tilde{\Lambda}_k$.
- Compute the Sharpe ratio of each arm. Compute the adjusted reward function of each arm.
- Select the optimal arms according to the adjusted reward function from the first l and the next $n - l$ orthogonal portfolios, respectively.
- Compute the optimal mixture weight θ_k^*.
- Compute the optimal portfolio weight ω_k.

Output: The portfolio weight vectors ω_k and the portfolio returns μ_k for $k = 1, \ldots, m$.

Chapter 3

Statistical Change Point Analysis

So far, we have addressed the potential issues of the classical bandit frameworks such as collinearity and multiple-action selection, but one challenge is still left unaddressed: non-stationarity.

Changepoints are defined loosely as "moments of abrupt change in the behaviour of a time series," which "may signal a significant alteration to the data generating process" [106]. The literature on *changepoint detection (CPD)* is concerned with accurately detecting these events in time series data, as well as with modelling time series in which such events occur (*ibid.*). Methods for CPD can be roughly categorised into (1) online/offline, (2) univariate/multivariate, (3) model-based/nonparametric, (4) Bayesian/frequentist, (5) in nonparametric methods: divergence estimation/heuristics (*ibid.*).

3.1 Background

Change Point Detection (CPD) analysis is not new and the first publications on the subject appeared in the 1950s [31], [84] and 1960s [79, 100, 45, 103], with some publications in the 1970s gaining significant traction [50]. A recent survey of the subject has appeared in [106]. Changepoints have also been considered specifically in the context of finance and economics [66].

The fact that our data is a time series is a crucial aspect of analysis and there is extensive literature on Change Point Detection to signal whenever a significant modification occurred in our data. This extra edge when constructing a portfolio may help re-balance the weights if they over (or under) commit on certain actions, preventing further decline in the total cumulative rewards.

The first results on changepoint detection were made by Page who described the CUMSUM (Cumulative Sum) method [84] by which a cumulative sum is used to identify shifts and/or changes in the distribution of a sample set if the respective quantity exceeds a predefined threshold. The technique was later rigorously analysed by Lorden [71] within a solid theoretical framework. Later on, maximum likelihood methods [50] as well as multiple linear regression estimates [12] were researched

As for nonparametric models, they heavily rely on explicit hypothesis testing where the detection is being done with test statistics, such as Student's T. Of course, many other functions could be used, which is what inspired the research of kernel changepoint analysis [48] and even divergence measures or histograms were used for such testing [75, 17].

The method we will use in Chapter 4 is known as Bayesian online point detection, simultaneously proposed by Fearnhead and Liu [40] and Adams and MacKay [2]. These models learn a probability distribution over the "run length", which is the time since the most recent change point. In their analysis, Adams and MacKay looked at shifts in a time series using the Dow Jones Industrial Average performance during the 1970s

close to the Watergate incident and they noticed spikes in the probability of the posterior distribution of returns very close to key events that occurred, such as the resignation of former President Richard Nixon. Nevertheless, the formulation of Adams and MacKay has been extended in several works to include Gaussian Process segment models ([82], [92]).

It is common knowledge [107, 46] that financial time series are usually nonstationary. Moreover, they are nonstationary in particular ways, usually considered as stylized facts [41]: for example, *volatility clustering*: "large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes" [73]. Moreover, changes in market parameters often happen abruptly and multiple parameters change all at once, as in market crashes [7], although abrupt market dislocations happen far more frequently on a smaller scale [66]. This suggests that changepoint analysis could be used to study such phenomena.

The estimation and/or forecasting of the mean function of a financial stochastic process is at the core of finance including, but not limited to, alpha generation and portfolio selection. We will focus on estimation rather than forecasting. However, even the estimation of the mean of a financial time series is a difficult problem, as acknowledged by Merton [77].

3.2 Incremental Mean Estimation

The standard way to estimate the mean of a distribution is via the sample mean. Suppose that $X_1, \ldots, X_N$ are realizations of a random variable X, so $X_1, \ldots, X_N$ are independent and identically distributed (i.i.d.). Then the sample mean is defined as

$$\overline{X} = \frac{1}{N}\left[X_1 + \ldots + X_N\right] = \frac{1}{N}\sum_{i}^{N} X_i.$$

We can show that, irrespective of the distribution of X, the sample mean is an unbiased estimator:

$$\mathbb{E}[\overline{X}] = \mathbb{E}\left[\frac{1}{N}\left[X_1 + \ldots + X_N\right]\right] = \frac{1}{N}\sum_{i}^{N}\mathbb{E}\left[X_i\right]$$

by the linearity of the expectation operator $\mathbb{E}[\cdot]$ and, if the true mean of X is μ, using the assumption that $X_1, \ldots, X_N$ have the same finite mean μ,

$$\mathbb{E}[\overline{X}] = \frac{1}{N}\sum_{i}^{N}\mu = \frac{1}{N}\cdot N\mu = \mu.$$

Since the mean squared error (MSE) of an estimator can be represented as the sum of the estimator's variance and squared bias, the MSE of the sample mean is equal to the sample mean's variance:

$$Var[\overline{X}] = \frac{1}{N^2}Var[\sum_{i}^{N} X_i].$$

If we use the assumption that $X_1, \ldots, X_N$ are independent, we can rewrite this further as

$$Var[\overline{X}] = \frac{1}{N^2}\sum_{i}^{N} Var[X_i].$$

Furthermore, if we assume that the variance of X is finite and is equal to σ^2, we can rewrite this further as

$$Var[\overline{X}] = \frac{1}{N^2}\cdot N\sigma^2 = \frac{\sigma^2}{N}.$$

The square root of this quantity is known as the *standard error of the mean*,

$$SE_{\overline{X}} = \frac{\sigma}{\sqrt{N}} \propto \frac{1}{\sqrt{N}}. \tag{3.2.1}$$

Thus to halve the standard error, we need four times as many samples; to reduce the standard error by a factor of ten, we need one hundred times as many samples; and so on.

So far our assumptions on the distribution of X have been mild: we assumed that $X_1, \ldots, X_N$ were i.i.d. samples drawn from the distribution of X; that that distribution has a constant mean μ; and that that distribution has a finite variance of σ^2. Apart from these mild assumptions, we didn't say anything about the distribution of X. In particular, we did not require that X be normally distributed. It is difficult to say more about the sample mean estimator without introducing stronger distributional assumptions.

3.3 A different Estimator

There is a strong result proven by Bondesson [16] that the sample mean of observations taken from a location-scale parameter family of distributions is BLUE (best linear unbiased estimator) of the location parameter for all sample sizes if and only if the distribution is normal with mean zero or gamma translated to have mean zero.

In some respect, this is good news. For example, if, for $t \in [0, T]$, $T \in \mathbb{R}^+$, the price process is

$$dS_t = \mu\, dt + \sigma\, dW_t,$$

i.e., a scaled Wiener process with drift μ and volatility σ, then the increments of this process are normally distributed:

$$\Delta S_t = S_t - S_{t-\Delta t} \sim \mathcal{N}\left(\mu\Delta t, \sigma^2\Delta t\right).$$

But a caveat lurks here. Suppose that we have a time series, $X_1, X_2, \ldots, X_N$. When estimating its mean increment via the sample mean estimator, we end up with a telescoping sum:

$$\begin{aligned}
\overline{\Delta X} &= \frac{1}{N-1}\left[(X_2 - X_1) + (X_3 - X_2) + \ldots + (X_N - X_{N-1})\right] \\
&= \frac{1}{N-1}\left[-X_1 + (X_2 - X_2) + (X_3 - X_3) + \ldots + (X_{N-1} - X_{N-1}) + X_N\right] \\
&= \frac{1}{N-1}\left[X_N - X_1\right].
\end{aligned}$$

Thus only the first (X_1) and last (X_N) terms contribute to the sample mean. There is no point in increasing the sampling frequency.

In our case, if we sample the stochastic process at equally spaced times, $0 = t_0, t_1 = \Delta t + t_0, t_2 = \Delta t + t_1 = 2\Delta t + t_0, \ldots, t_N = T = N\Delta t + t_0$, so that $T = N\Delta t$, then

$$\overline{\Delta S} = \frac{1}{N}\left[S_{t_N} - S_{t_0}\right],$$

and $\mathbb{E}[\overline{\Delta S}] = \mu\Delta t$, so an unbiased estimator of the location parameter μ is given by

$$\frac{\overline{\Delta S}}{\Delta t} = \frac{1}{T}\left[S_T - S_0\right].$$

The variance of this estimator will then be given by

$$Var\left[\frac{\overline{\Delta S}}{\Delta t}\right] = \frac{\sigma^2\Delta t}{N\Delta t^2} = \frac{\sigma^2}{T}. \tag{3.3.1}$$

Notice that only T affects the variance of this estimator; Δt and N, and, by implication, the sampling frequency, does not affect this variance.

Trivially, the standard error of this estimator is the square root of the above variance, namely $SE_{\frac{\Delta S}{\Delta t}} = \frac{\sigma}{\sqrt{T}}$. Thus to halve the standard error, we need the dataset to span four times as much time; to reduce the standard error by a factor of ten, we need the dataset to span one hundred times as much time. (An order of magnitude improvement in the accuracy in the estimation of the location parameter μ requires a two orders of magnitude increase in the duration of the dataset, which is difficult in practice.)

It is more common to model the price of an asset as a geometric Brownian motion,

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t, \tag{3.3.2}$$

since then the model price S_t is always nonnegative. In this case the increments $\Delta S_t = S_t - S_{t-\Delta t}$ are no longer normally distributed. The stochastic differential equation (3.3.2) has a closed form solution

$$S_t = S_o \exp\left[\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t\right],$$

whence the log-price $X_t := \log(S_t)$ follows the scaled Wiener process with drift

$$X_t = \ln(S_0) + \left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t.$$

Therefore the increments are, as before, normally distributed

$$\Delta X_t = X_t - X_{t-\Delta t} \sim \mathcal{N}\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma^2 \Delta t\right).$$

The sample mean estimator can be used to estimate the location parameter of the above normal distribution, namely $\left(\mu - \frac{\sigma^2}{2}\right)\Delta t$. One of the numerous estimators of the variance can then be used to estimate the scale parameter, or $\sigma^2 \Delta t$. But we are back to the same problem: only the endpoints and the length (in units of time) of the time series feature in the calculation of the sample mean.

3.4 Bandits with Changepoints

The current literature on implementing a framework for changepoints within a bandit setting is scarce, as the main focus is traditionally directed towards improving the theoretical regret bounds and adjusting for the covariance between different arms. Since our problem is greatly concerned with practical results in an online setting, it is worth taking a look at how could a changepoint framework can help in a bandit setting.

First of all, we recall that one of the three main issues with using classical bandits for a portfolio choice problem is the nonstationarity of rewards. The main approach is to consider piece-wise stationary bandits [14]. More specifically, we assume that in a sequence of rewards $X_1, X_2, ..., X_n$ the sample is not independent and identically distributed (iid) anymore and in fact we assume that there exists a set of changepoints $c_0 < c_1 < ... < c_\tau$ such that each sub-sample $X_{c_k+1}, X_{c_k+2}, ..., X_{c_{k+1}}$ is stationary.

With this simple idea in mind, it's only natural to develop a sub-routine for any bandit framework that uses a mean estimator for the rewards. In such a setting, the mean can be estimated using the results of Section 3.3 in the following way:

Lemma 3.4.1. *(Changepoint mean estimator) Let $S_1, S_2, ..., S_n$ be a sequence of random variables. Let $\tau \leq n$ be the number of changepoints of the sequence, i.e. there exists an increasing array $0 = c_0 \leq c_1 < c_2 < ... < c_{\tau-1} \leq c_\tau = n$ such that the sub-sequence $X_{c_k+1}, X_{c_k+2}, ..., X_{c_{k+1}}$ is stationary. Then, the following estimators for the* ***mean increments*** *are unbiased:*

$$\hat{\mu}_k := \frac{1}{c_k - c_{k-1}}[S_{c_k} - S_{c_{k-1}}], \text{ for any } 1 \leq k \leq \tau.$$

The results from [6] provide a framework for bandits in this scenario by restarting the selected arm count once a changepoint is detected in an UCB framework. This forces the algorithm to learn again what are the best context-dependent arms, adjusted to a new mean. We shall see in the next chapter how changepoints can influence a dataset.

3.4.1 PSRCP-UCB

We can apply the current changepoint framework to our `PSR-UCB` algorithm, defined in 2.4. Restarting the observation count and consequently the Sharpe estimators will essentially form a dynamic moving average which will maintain relevance for the most recent step to be updated. For the actual changepoint detection method we would need to resort to an online method to match the dynamic of the UCB framework.

Algorithm 6: PSRCP-UCB

[H]
Input: k, n, γ, p
Parameters: $T_i(0), \hat{SR}_i(0), \hat{\gamma_3}^i(0), \hat{\gamma_4}^i(0)$
Set $\gamma = \lfloor \gamma \rfloor \cdot n$
for $i = 1, 2, \ldots, \gamma$ **do**
  Play arm i
  Update $T_i(t), \hat{SR}_i(t), \hat{\gamma_3}^i(t), \hat{\gamma_4}^i(t)$
end for
Calculate $\text{UCB}_i^{PSR}(\gamma)$ for each $i = 1, 2, \ldots, k$ from 2.2.5
for $t = \gamma + 1, \gamma + 2, \ldots, n$ **do**
  Play arm $a_i = \arg\max_{i \in \{1,2,\ldots,k\}} \text{UCB}_i^{PSR}(k)$ Update $T_i(t), \hat{SR}_i(t), \hat{\gamma_3}^i(t), \hat{\gamma_4}^i(t)$
  Calculate $\text{UCB}_i^{PSR}(t)$ from (2.2.5)
  If t is a changepoint for a specific arm (strategy) with probability $\geq p$, then reset all parameters for the specific action.
end for

The tuning of parameter p is paramount, as depending on the dataset the number may vary. For example, for clear generated sets with artificially introduced changepoints the probabilities will be very high (over 0.9), whereas for a real time-series for which the data has a high correlation, we may need to reduce the threshold. There is a fine balance of choosing this parameter, as lower values will imply detecting more changepoints, thus loosing the accuracy of the UCB algorithms, which are known to be asymptotically optimal and require a lot of arm pulls. In fact, we may expect a lower performance than the conventional UCB methods. However, this idea remains notable as the `PSR-UCB` bound and the idea of estimating the Sharpe ratio even in the absence of normal-distributed returns signify a key starting point for future research.

The actual changepoint detection method we use is, as stated during the introduction, Bayesian Online Detection, based on [3] and using the Python library `sdt-python`. For more information, the reader may consult Appendix B.

Chapter 4

Evaluation

In this final chapter we propose several experiments to analyse all discussed methods and compare them on several sets of data to assess the performance of our proposed work on both synthetic and real-world datasets. For the synthetic dataset, we selected Brownian Motion simulations with various values for drift and volatility and a small adjustment which we will explain in the next section. Our analysis will consist of three experiments:

- `PCR-UCB` performance: We will showcase the advantages and limits of Algorithm 2 by comparing it with the known benchmark `UCB-RSSR`, which also uses Sharpe ratio in the confidence bound estimation.
- Changepoint estimation: We will test the changepoint framework for the UCB-type algorithms and discuss the key ideas of arm restarts.
- Predictions with expert advice: We showcase the performance of an `Exp4` algorithm, more closely aligned with `Hedge` [43] within a full-information feedback scenario.

4.1 Data

First, we have generated synthetic equity curves based on a drifting Brownian Motion Process based on certain parameters, such as mean and volatility. A key addition is that we have introduced an additional parameter which represents **informed** curves, which have reduced variance and therefore will decrease to a lesser extent. A good bandit algorithm will pick those curves and take advantage. However, it is interesting to see what kind of feedback will be best in this scenario. Furthermore, to account for nonstationarity we have split the synthetic data in two categories:

- Constant mean: The theoretical drift parameter is constant throughout the whole equity curve generation.
- Shifting mean: For each curve, we pick a random number of changepoints where the drift will shift, thus generating nonstationary equity curves.

Secondly, we have also looked at real-world data and picked two datasets:

- Financial markets stocks: Using `yfinance` Python library, we have taken 10 stocks from the Dow Jones Index to analyze the performance of adversarial bandits.
- Alternative Data: Using `Ray`, we downloaded Google Trends data for signal detection purposes, with the hope of identifying good PnL curves that resemble stock market activity.

4.2 Experiment 1: PSR-UCB

For the first experiment, we aim to test the performance of the `PSR-UCB` algorithm we defined in 2. Since we want to analyse finance-related scenarios, we generate equity curves, which can be interpreted as PnL curves, using a Brownian Motion process with different parameters. Moreover, we will 'boost' some of the curves and we will name them **informed** curves. More specifically, we incrementally add a fractional drift to a random number of already generated curves and thus we can emulate an effective trading strategy or profitable portfolio. We implement this adjustment to the curves because we want to see if the algorithms will find true value within a large range of options, amongst many of which are actually random. Furthermore, it's also much more practical to directly analyze the PnL generated by the respective algorithms. The methods used in this experiment are described in Table 4.1, where `SW` represents the Sharpe Weighted portfolio, where we weigh each arm by the most recent empirical Sharpe ratio.

Strategy	γ	δ / **p**
SW	N/A	N/A
UCB	0.01	dynamic ($1/n^2$)
UCB-RSSR	0.01	dynamic ($1/n^2$)
PSR-UCB	0.01	0.999

Table 4.1: Parameter Settings for Different Strategies

Figure 4.1 showcases the PnL of the four strategies employed for a full-bandit feedback setting for different numbers of both informed and regular arms among $20,000$ time steps averaged over 30 iterations. The drift and volatility are set to 0.1 at the start of the curve generation process. The informed strategies will have higher incremental means.

Figure 4.2 aims to replicate the exact case study conducted in [61], more specifically we will test the algorithms against an uniform distribution with mean and variance vectors $\mu = [0.05, 0.07, 0.055, 0.07, 0.06], \sigma^2 = [0.1, 0.1, 0.1, 0.1, 0.1]$, picking the first values in their respective order if the action count is smaller than 5. We immediately notice that `PSR-UCB` outperforms all other methods in both scenarios.

We notice that in this scenarios `PSR-UCB` outperforms all benchmarks.Furthermore we make the following two remarks:

- The `PSR-UCB` is derived from a confidence interval inequality, and so we may treat it as a statistical test. More specifically, we can say that the proposed framework will detect inherently good strategies that produce good overall results with a certain confidence level. The intuition behind such a good performance could be justified by the fact that we use the skewness and the kurtosis to estimate the Sharpe ratio, two additional moments that provide more information to our context.

- The other UCB algorithms loose performance as the number of arms increases, whereas `PSR-UCB` maintains the same rate of growth (it even increases in some cases) throughout all simulations. This is because the asymptotic guarantees of the Sharpe ratio estimator provide much tighter regret bounds, thereby increasing overall profits. Notice, however, that the performance for the early stages is actually worse for `PSR-UCB` when using 10 arms, suggesting that we need a broad time horizon for a good analysis. We will see how this impacts changepoint analysis in Experiment 2.

In addition to the synthetic data, we analyse the Google Trends dataset and we build PnLs based on 2 different sets. Set 1 describes the actual trends on various industries, such as *gas, oil, airport, casino, bitcoin*, totalling to 20 curves in total. Set 2 describes

transformations on the first set, such as moving averages and rolling window increments bundled together with the contents of Set 1, amounting to 60 curves.

The reasoning for using Google Trends is the following: because the `PSR-UCB` is built to extract the most value provided value can be found, then a good enough performance will indicate the presence of signals based on alternative data. Due to the more sporadic behaviour of trends data, we need to adjust the exploration parameter γ which controls how many actions we explore at the beginning of the algorithms. Judging from 4.3, we conclude that a $\gamma \approx 0.02$ is most adequate for our setting. Of course, we could explore more, but that would defeat the purpose of exploitation.

In Figure 4.4 we see the outputs based on Sets 1 and 2 with the settings given from Table 4.1 with the modification of the γ parameter to 0.02. We notice that the `PSR-UCB` is consistent amongst the two datasets, whereas `UCB-RSSR` struggles with mixed data and more arms altogether, but still outranking the classical `UCB` with the exception of one case. What is more curious is that the Sharpe ratio UCB methods offer an advantage compared to the classical weighting scheme of `SW`, which has full-information feedback. In that regard, we can affirm that indeed less is more.

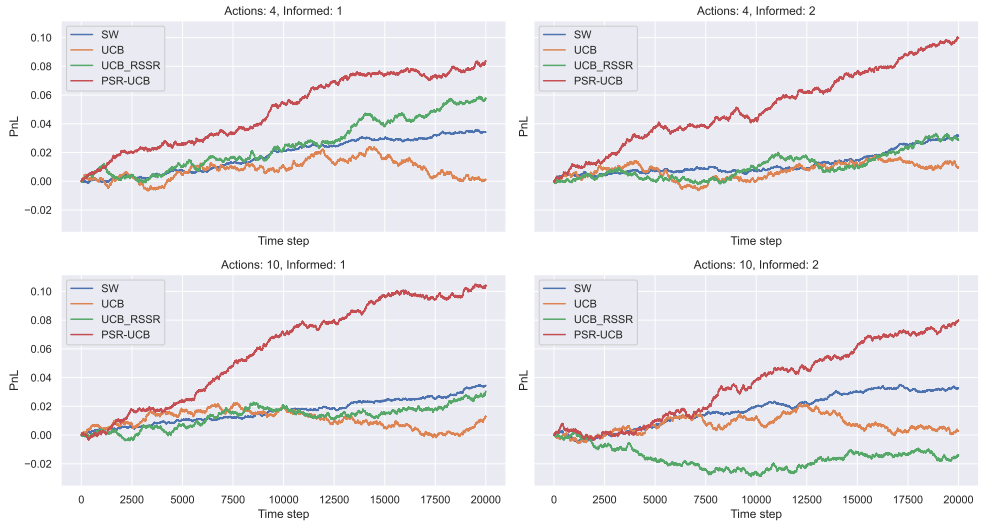

Figure 4.1: Experiment 1 – Synthetic Data

Figure 4.2: Experiment 1 – Uniform Data

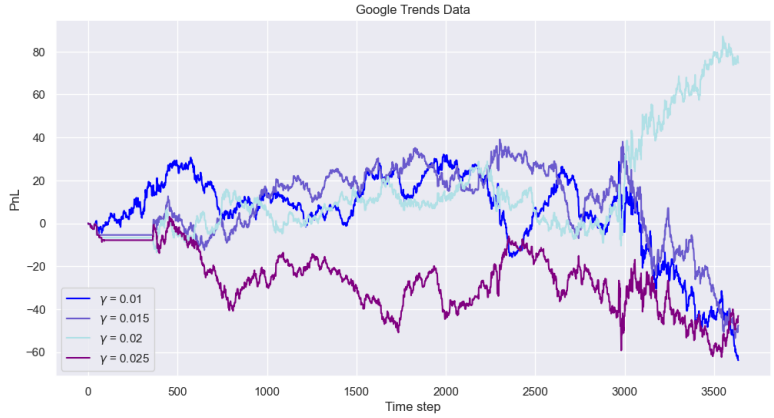

Figure 4.3: Experiment 1 – γ parameter optimisation; `PSR-UCB`

Figure 4.4: Experiment 1 – Google Trends Data

4.3 Experiment 2: Changepoint UCB

In this second experiment, we aim to implement the changepoint algorithms first on synthetic data, which has randomly assigned changepoints during generation. More specifically, we randomly select their position within the time series and increase the drift parameter for the remainder of the generation. The tuning of parameter p, which sets the threshold for detecting changepoints is crucial, as that is what determines the added value of these methods.

For synthetic data, the algorithms detected all artificially added changepoints with a

threshold of 0.85, since the shift of the distribution was evident. Figure 4.5 showcases this.

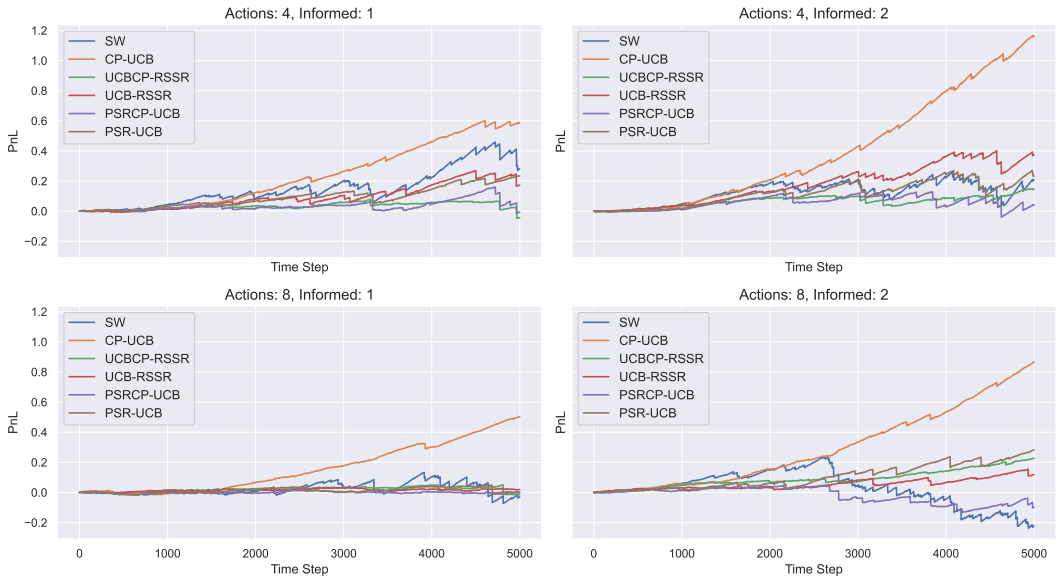

Figure 4.5: Experiment 2 – Synthetic Data with artificially added changepoints

Again, for Google Trends we maintain the same data set as in Section 4.2. The settings for the experiment are described in Table 4.2.

Strategy	γ	δ / **p**	**Probability Threshold**
PSR-UCB	0.02	0.999	0.85
CP-UCB	0.02	dynamic $(1/n^2)$	0.85
UCBCP-RSSR	0.02	dynamic $(1/n^2)$	0.85
PSRCP-UCB	0.02	0.999	0.85

Table 4.2: Parameter Settings for Changepoint Strategies

A big drawback of using changepoint analysis within the full-bandit setting is that we don't have access to all data points to correctly detect unusual behaviour, so we can expect a drop in performance and also an increase in computational cost. The UCB algorithms have asymptotic optimality, and thus the number of arms explored needs to increase in order to reach convergence, whereas if we keep restarting the analysis for every partial observation made, we may in fact hinder our strategy.

Assuming we have contextual information regarding the changepoints, i.e. we know where every changepoint is and restarting based on this assumption, we can look at Figure 4.5 to notice that we have actually value within this framework. This situation would relate to a scenario where the changepoint information is extracted from another source than the actual equity curve. Moreover, the additional increase in the drift parameter is significant, as setting it to be to high will cause the overall Sharpe ratios to decrease, since variance will increase due to the large gap. To showcase the key aspects of the changepoint analysis, we have picked a limit value, namely for an initial drift parameter of $\mu = 0.1$, we add increments of 0.5 at every changepoint.

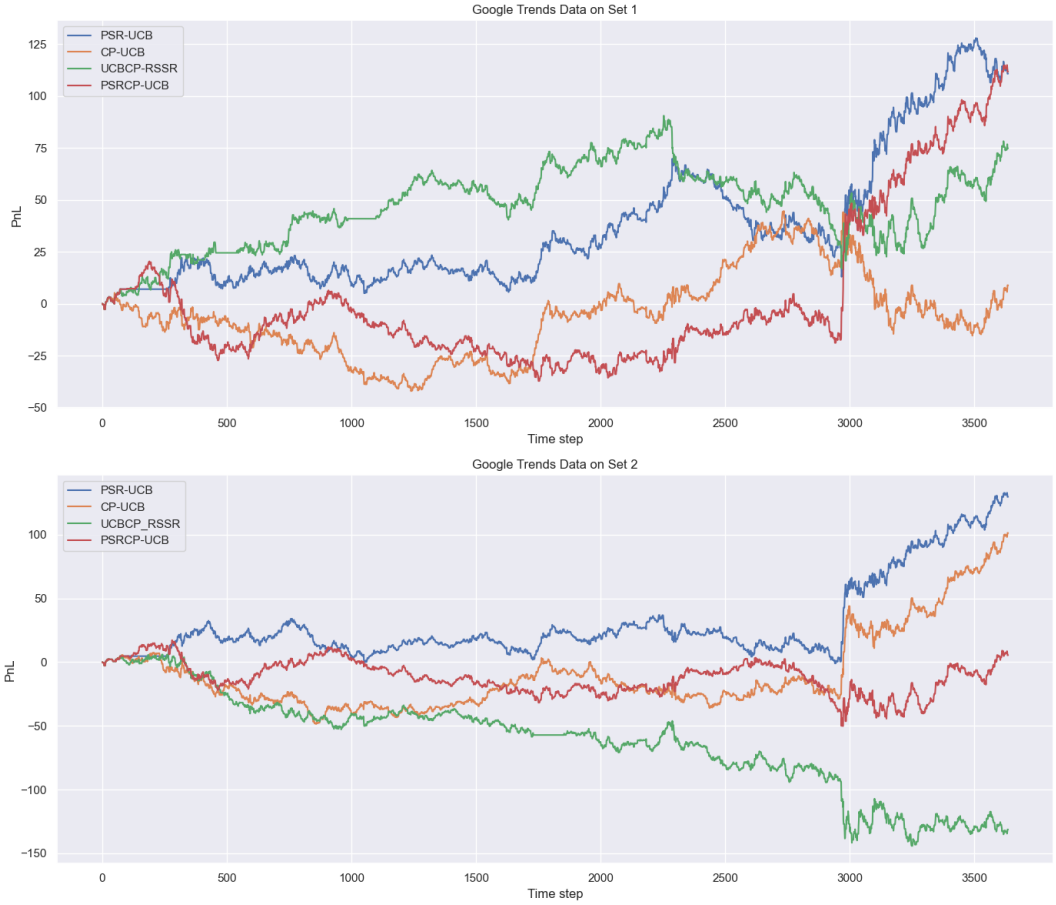

Figure 4.6: Experiment 2 – Google Trends Data

We conclude this experiment with the following observations:

- The `PSRCP-UCB` algorithm will underperform in its current form. A possible explanation is that we don't rely on the asymptotic optimality of the regret bounds, but also on the asymptotic behaviour of the Sharpe ratio estimator. A large number of samples is required to achieve the properties of a normal distribution for the confidence bounds to hold. As we saw from Experiment 1, for the first few thousand of rounds the performance of all methods is marginal and we can't determine if we have found true value until a later period is reached. Thus, the changepoint algorithm as it is now may work only for a limited number of restarts for the PSR methods.

- On the other hand, the `UCB` and `UCBCP-RSSR` algorithms are reacting very well to this dataset. The explanation, which also justifies the poor performance of the `PSR-UCB` framework, is that the upper confidence term does **not** relate to the Sharpe ratio for the former methods. For the PSR model, the mentioned term is precisely the estimator for the quantity that hinders the whole progression, which is also confirmed by the low values of the `SW` results.

4.4 Experiment 3: Adversarial Bandits

For adversarial bandits, we maintain the same datasets as in Experiment 1. In addition, we will also analyse the Dow Jones Stocks taken from the `yfinance` and listed in Table 4.4 from 2012/01/01 to 2023/01/01.

Stock Indices
MMM, AXP, AMGN, AAPL, BA, CAT, CVX, CSCO, KO, DIS

Table 4.3: List of Stocks used for Experiment 3

For this scenario, we use `Exp4-prediction`, a combination between `Exp4` and `Hedge` using Sharpe Weighted (SW), Calmar Weighted (CW) and Clever Weighted (ClW) experts. We denote it as a combination because we utilise the `Exp4` code, but the experts have access to full-information feedback. Moreover, we will analyse the `OBP` [99], the `MC-empirical` [38] and the `SW-CP` algorithms. The latter is only a modification of `SW` in which we implement the changepoint detection sub-routine. As for the `MC-empirical` method, the authors use the mean-variance reward function defined in (2.5.1) and thus risk is managed very well. We can expect a lower, but much more stable PnL.

Figure 4.7 displays the results of the synthetic data simulations with a time horizon of $4,000$ time steps over 20 iterations. We notice that the `Exp4-prediction` algorithm handles well the limited expert advice it has, namely the SW, CW and ClW weighting schemes. Because a probability distribution is used to sample an arm, constrained by the expert advice, which is at its turn represented by a set of probabilities, we can affirm the overall performance of the method maintaining a fairly stable evolution of PnL.

Figure 4.8 showcases the results of the Google Trends dataset PnLs. We notice the increased volatility of applying change point detection to a full-feedback method that aims to optimise the Sharpe ratio, thus proving again a poor performance when it comes to resetting the sample space at the discovery of a new change in the distribution of our data. Moreover, we also remark a volatile, although effective curve produced by our proposed expert selection scheme, especially on Set 1, where the cutoff point at around timestep 3000 is managed very well. Furthermore, the evolution of `OBP` is stable and doesn't yet exploit the value of the dataset. We suspect the algorithm is only effective on very long windows of time, as the original analysis conducted in [99] is made based on data ranging from the 1970s to the present day (2015 at the time of the article). There isn't a large discussion regarding what sliding window should be used nor additional results on different time horizon settings to achieve the optimal parameter selection.

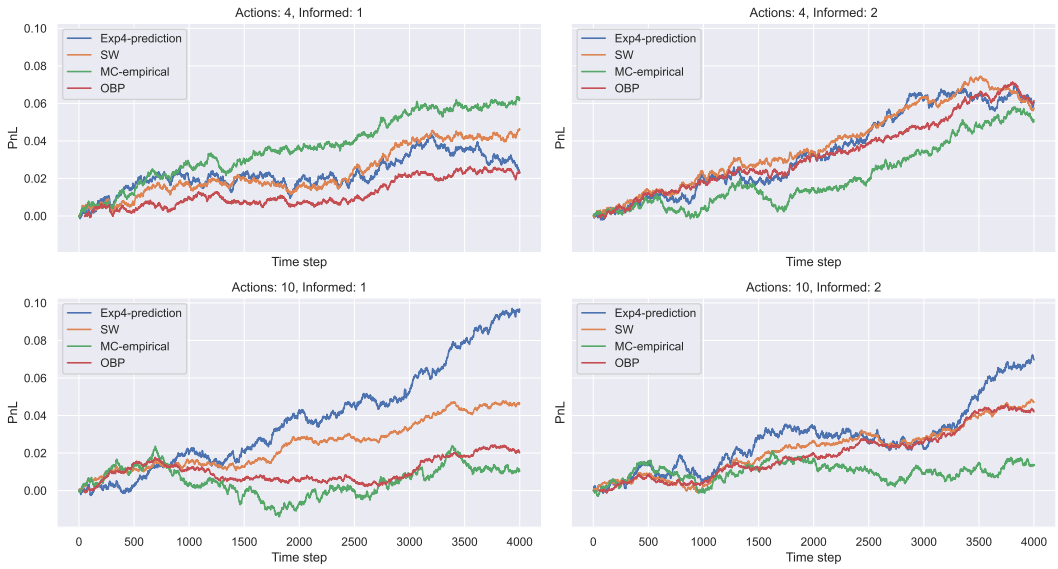

Figure 4.7: Experiment 3 – Synthetic Data

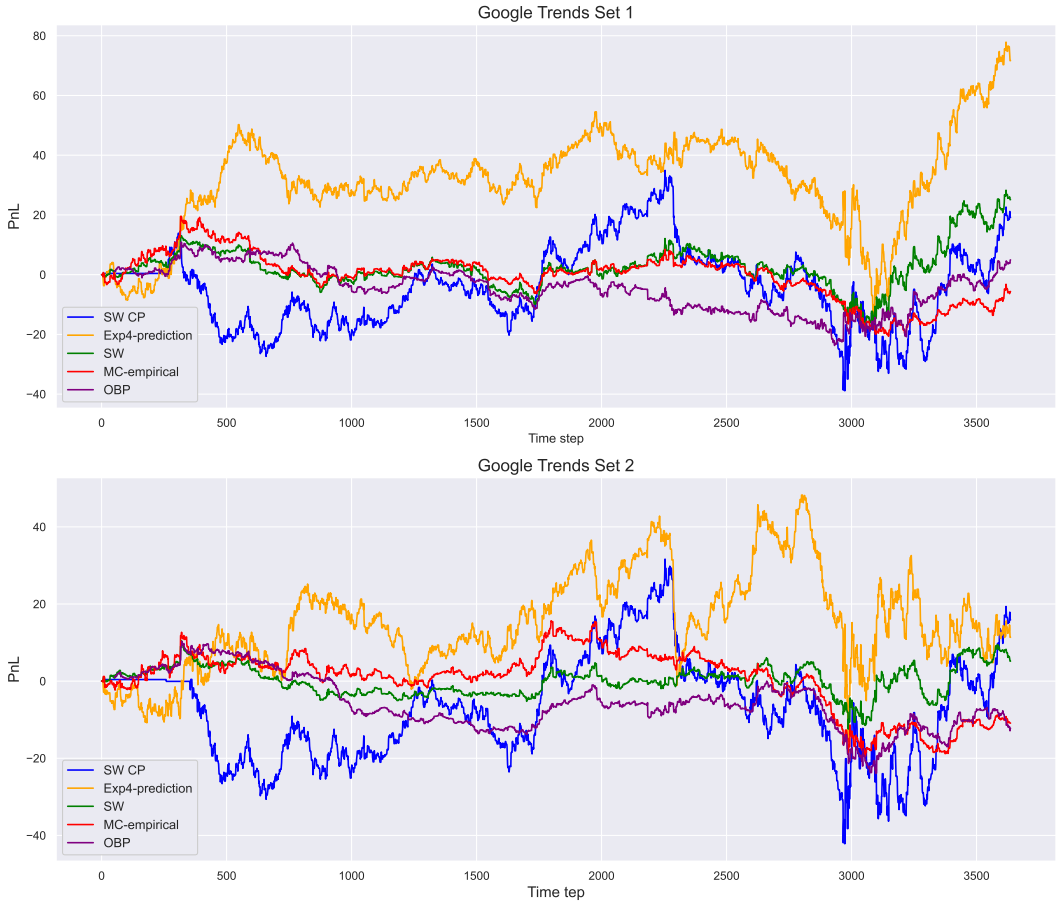

Figure 4.8: Experiment 3 – Google Trends Data

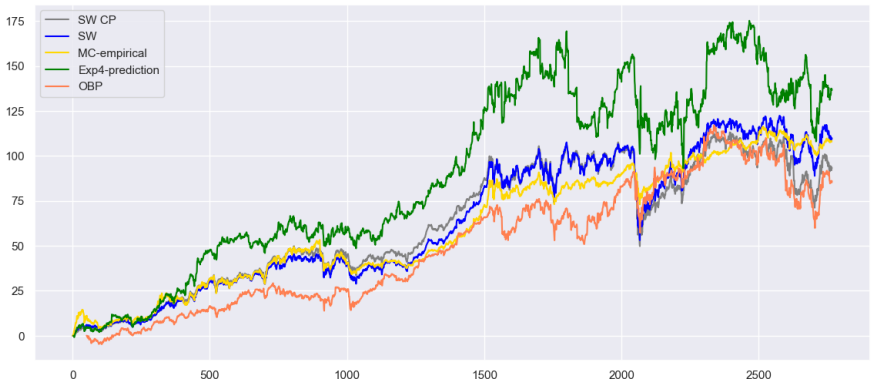

Figure 4.9: Experiment 3 – Stock Data

Conclusion

We have introduced `PSR-UCB`, a multi-armed bandit algorithm designed to maximise returns using the Probabilistic Sharpe ratio as a proxy for constructing confidence bounds for which a logarithmic regret is achieved. Furthermore, we empirically showed the greater advantage when compared with the current benchmark `UCB-RSSR` that aims to maximise the Sharpe ratio to build the confidence bounds. We have achieved theoretical guarantees on the asymptotic regret bounds and conjecture that they can be further improved, judging from the empirical results.

Furthermore, we extended the framework to changepoint detection tests to account for nonstationarity in the time series of rewards, which could be a promising avenue to pursue for future research, as the asymptotic nature of the estimator used in 2 requires a large dataset to achieve logarithmic regret and, as a result, repeated restarts of the arm counters currently hinder the algorithm's capabilities.

The reasoning behind choosing the Sharpe ratio is the extensive study towards it, being one of, if not the most, important metrics in portfolio management. Less is known about, for example, the different variants of the Calmar ratio, although even if the estimator's distribution is analytically intractable, it should be straightforward to come up with a polynomial approximation, which could contribute an upper bound.

Regarding the adversarial setting, we have analysed how multi-collinearity can impact the overall portfolio and proposed a tractable expert selection scheme. While our proposal doesn't account for risk and it provides volatile results, it still performs very well against the `MC-Empirical` [38] and `OBP` [99] benchmarks.

Appendix A

Technical Proofs

A.1 Mean and Variance Estimators

In Section 1.2 we introduced the following two estimators:

$$\hat{\mu} = \frac{1}{N\delta t}\sum_{i=0}^{N-1} R_i, \quad \hat{\sigma}^2 = \frac{1}{(N-1)\delta t}\sum_{i=0}^{N-1}(R_i - \hat{\mu}\delta t)^2,$$

for which we also stated two lemmas, which we prove here.

A.1.1 Proof of Lemma 1.2.1

Proof. We start with the mean estimator. Given that we considered $R_i = \frac{S_{i+1}-S_i}{S_i}$, where S_i follows a GBM process, we have the closed-form solution for the stock price at any given time (insert reference) to be:

$$S_t = S_0 \exp\left((\mu - \frac{\sigma^2}{2})t + \sigma W_t \right) \tag{A.1.1}$$

Using the closed-form solution in the definition of R_i, which corresponds to the return at time t_i, we obtain

$$\begin{aligned} R_i &= \frac{S_{i+1} - S_i}{S_i} = \frac{S_{i+1}}{S_i} - 1 \\ &= \frac{S_0 \exp\left((\mu - \frac{\sigma^2}{2})t_{i+1} + \sigma W_{t_{i+1}} \right)}{S_0 \exp\left((\mu - \frac{\sigma^2}{2})t_i + \sigma W_{t_i} \right)} - 1 \\ &= \exp\left((\mu - \frac{\sigma^2}{2})\delta t + \sigma\sqrt{\delta t} Z_{t_i} \right) - 1. \end{aligned}$$

In the last equality we have used the fact that $t_{i+1} - t_i = \delta t$, as mentioned in Section 1.2. Moreover, note the Z_{t_i} term, which is a standard normal random variable, derived from the well-known property of Brownian Motion increments:

$$W_{t_{i+1}} - W_{t_i} \sim \mathcal{N}(0, \delta t^2)$$

We can now compute the mean of the returns as

$$\begin{aligned}
\mathbb{E}[R_i] &= \mathbb{E}\left[\exp\left((\mu - \frac{\sigma^2}{2})\delta t + \sigma\sqrt{\delta t}Z_{t_i}\right) - 1\right] \\
&= \mathbb{E}\left[\exp\left((\mu - \frac{\sigma^2}{2})\delta t\right) \cdot \exp\big(\sigma\sqrt{\delta t}Z_{t_i}\big)\right] - 1 \\
&= \exp\left((\mu - \frac{\sigma^2}{2})\delta t\right)\mathbb{E}\big[\exp\big(\sigma\sqrt{\delta t}Z_{t_i}\big)\big] - 1 \\
&= \exp\left((\mu - \frac{\sigma^2}{2})\delta t\right)\exp\left(\frac{\sigma^2}{2}\delta t\right) - 1 \\
&= \exp(\mu\delta t) - 1. \qquad \text{(A.1.2)}
\end{aligned}$$

For the first 3 equalities we have used the linearity and homogenity of expectation, whereas for the fourth one we have used the definition of the moment generating function for standard random normal variables, giving us the final result. Now we proceed to show that $\hat{\mu}$ is unbiased.

$$\begin{aligned}
\mathbb{E}[\hat{\mu}] &= \mathbb{E}\left[\frac{1}{N\delta t}\sum_{i=0}^{N-1} R_i\right] \\
&= \frac{1}{N\delta t}\sum_{i=0}^{N-1}\mathbb{E}[R_i] \\
&\overset{A.1.2}{=} \frac{1}{N\delta t}\sum_{i=0}^{N-1}\big(\exp(\mu\delta t) - 1\big) \\
&= \frac{N\exp(\mu\delta t) - N}{N\delta t} \\
&= \mu \cdot \frac{\exp(\mu\delta t) - 1}{\mu\delta t} \underset{\delta t\to 0}{\longrightarrow} \mu \cdot 1 = \mu.
\end{aligned}$$

For the last line of the equality we have used the well-known limit

$$\lim_{x\to 0}\frac{\exp(x) - 1}{x} = 1.$$

Now, for the volatility estimator we have

$$\begin{aligned}
\mathbb{E}[\hat{\sigma}^2] &= \frac{1}{(N-1)\delta t}\,\mathbb{E}\left[\sum_{i=0}^{N-1}(R_i - \hat{\mu}\delta t)^2\right] \\
&= \frac{1}{(N-1)\delta t}\,\mathbb{E}\left[\sum_{i=0}^{N-1} R_i^2 - 2\hat{\mu}\delta t\cdot\sum_{i=0}^{N-1} R_i + N\cdot\hat{\mu}^2\delta t^2\right] \\
&= \frac{1}{(N-1)\delta t}\,\mathbb{E}\left[\sum_{i=0}^{N-1} R_i^2 - 2N\hat{\mu}\delta t^2\cdot\frac{1}{N\delta t}\sum_{i=0}^{N-1} R_i + N\cdot\hat{\mu}^2\delta t^2\right] \\
&= \frac{1}{(N-1)\delta t}\mathbb{E}\left[\sum_{i=0}^{N-1} R_i^2 - N\hat{\mu}^2\delta t^2\right] \\
&= \frac{1}{(N-1)\delta t}\left(\sum_{i=0}^{N-1}\mathbb{E}[R_i^2] - N\delta t^2\mathbb{E}[\hat{\mu}^2]\right). \qquad \text{(A.1.3)}
\end{aligned}$$

The first term inside the brackets is computed as

$$
\begin{aligned}
\mathbb{E}\big[R_i^2\big] &= \mathbb{E}\left[\left(\frac{S_{i+1}}{S_i} - 1\right)^2\right] \\
&= \mathbb{E}\left[\left(\frac{S_{i+1}}{S_i}\right)^2 - 2 \cdot \frac{S_{i+1}}{S_i} + 1\right] \\
&\overset{(A.1.1)}{=} \mathbb{E}\left[\exp\left(2(\mu - \frac{\sigma^2}{2})\delta t + 2\sigma\sqrt{\delta t} Z_{t_i}\right)\right] - 2\mathbb{E}[R_i] - 1 \\
&\overset{(A.1.2)}{=} \exp(2\mu\delta t + \sigma^2\delta t) - 2\exp(\mu\delta t) + 1. \qquad \text{(A.1.4)}
\end{aligned}
$$

The second expectation term is expressed as

$$
\begin{aligned}
\mathbb{E}[\hat{\mu}^2] &= \frac{1}{N^2\delta t^2}\mathbb{E}\left[\sum_{i=0}^{N-1} R_i^2 + 2\sum_{0\le i<j\le N-1} R_i R_j\right] \\
&= \frac{1}{N^2\delta t^2}\left(\sum_{i=0}^{N-1}\mathbb{E}[R_i^2] + N(N-1)\mathbb{E}[R_i]^2\right) \\
&= \frac{1}{N^2\delta t^2}\left(N\big(\exp(2\mu\delta t + \sigma^2\delta t) - 2\exp(\mu\delta t) + 1\big) + N(N-1)\big(\exp(\mu\delta t) - 1)\big)^2\right) \\
&= \frac{1}{N^2\delta t^2}\left(N\exp(2\mu\delta t + \sigma^2\delta t) - 2N^2\exp(\mu\delta t) + N^2 + N(N-1)\exp(2\mu\delta t)\right). \qquad \text{(A.1.5)}
\end{aligned}
$$

Now, combining the equalities from (A.1.4) and (A.1.5) into (A.1.3), we get

$$
\begin{aligned}
\mathbb{E}\big[\hat{\sigma}^2\big] &= \frac{1}{(N-1)\delta t}\left(\sum_{i=0}^{N-1}\mathbb{E}[R_i^2] - N\delta t^2\mathbb{E}[\hat{\mu}^2]\right) \\
&= \frac{1}{(N-1)\delta t}\Bigg(N(\exp(2\mu\delta t + \sigma^2\delta t) - 2\exp(\mu\delta t) + 1) \\
&- \exp(2\mu\delta t + \sigma^2\delta t) + 2N\exp(\mu\delta t) - N - (N-1)\exp(2\mu\delta t)\Bigg) \\
&= \frac{1}{(N-1)\delta t}\Big((N-1)\exp(2\mu\delta t + \sigma^2\delta t) - (N-1)\exp(2\mu\delta t)\Big) \\
&= \frac{1}{\delta t}\Big(\exp(2\mu\delta t + \sigma^2\delta t) - \exp(2\mu\delta t)\Big) \\
&= \sigma^2\exp(2\mu\delta t)\cdot\frac{\exp(\sigma^2\delta t) - 1}{\sigma^2\delta t} \underset{\delta t\to 0}{\longrightarrow} \sigma^2\cdot 1\cdot 1 = \sigma^2.
\end{aligned}
$$

□

Now we proceed to show the bounds of these estimators in the following lemma.

A.1.2 Proof of Lemma 1.2.2

Proof. We start with the mean estimator. As the variance is invariant at additive operations with constants, and thus we have

$$
\begin{aligned}
\mathrm{Var}(\hat{\mu}-\mu) &= \mathrm{Var}(\hat{\mu}) \\
&= \mathbb{E}[\hat{\mu}^2] - \mathbb{E}[\hat{\mu}]^2 \\
&= \frac{1}{N\delta t^2}\Big(\exp(2\mu\delta t+\sigma^2\delta t) - 2N\exp(\mu\delta t) + N + (N-1)\exp(2\mu\delta t)\Big) \\
&- \left(\frac{\exp(\mu\delta t)-1}{\delta t}\right)^2 \\
&= \frac{1}{\delta t^2}\left(\frac{1}{N}\exp(2\mu\delta t+\sigma^2\delta t) - 2\exp(\mu\delta t) + 1 + \frac{N-1}{N}\exp(2\mu\delta t)\right) \\
&- \frac{\exp(2\mu\delta t) - 2\exp(\mu\delta t) + 1}{\delta t^2} \\
&= \frac{1}{\delta t^2}\left(\frac{1}{N}\exp(2\mu\delta t+\sigma^2\delta t) - \frac{1}{N}\exp(2\mu\delta t)\right) \\
&= \frac{1}{N\delta t}\cdot\frac{\exp(2\mu\delta t)\big(\exp(\sigma^2\delta t)-1\big)}{\delta t} \\
&= \frac{\sigma^2}{T}\cdot\exp(2\mu\delta t)\frac{\exp(\sigma^2\delta t)-1}{\sigma^2\delta t} \underset{\delta t\to 0}{\longrightarrow} \frac{\sigma^2}{T}
\end{aligned}
$$

Now, for the variance estimator we can use the results from [81] to conclude that

$$\mathrm{Var}(\hat{\sigma}^2) = \left(\kappa - \frac{N-3}{N-1}\right)\frac{\sigma^4}{N},$$

where κ is the kurtosis of the returns. □

A.2 The Delta Method

The Delta method is a classic technique used in statistic to analyse the asymptotic behaviour of estimators.

Theorem A.2.1. *(Delta method) Let $X_1, X_2, \dots X_n$ be a sequence of random variables satisfying*

$$\sqrt{n}(X_n-\theta) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\sigma^2),$$

where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution.

Then, for any function g for which $g'(\theta)$ exists and it's finite, we have

$$\sqrt{n}(g(X_n)-g(\theta)) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\sigma^2\cdot[g'(\theta)]^2).$$

Now, suppose we have an estimator $\hat{\theta}_n$ for parameter θ and that we also have the **consistent** estimator for the variance $\hat{\sigma}_n^2$, i.e. $\hat{\sigma}_n^2 \longrightarrow \sigma^2$ at least in probability. If the following condition holds

$$\sqrt{n}(\hat{\theta}_n-\theta) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\sigma^2),$$

then we can construct a confidence interval by firstly considering the fact that

$$\frac{\sqrt{n}(\hat{\theta}_n-\theta)}{\hat{\sigma}_n} = \frac{\sqrt{n}(\hat{\theta}_n-\theta)}{\sigma}\cdot\frac{\sigma}{\hat{\sigma}_n} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1),$$

where the first fraction converges to the normal distribution via the Delta method and the second fraction converges to 1 because $\hat{\sigma}_n^2$ is consistent. Thus, it is now clear where that the $n-1$ term that is due to Bessel's correction is already contained withing the variance estimator and the additional $\sqrt{n}$ factor is added such that the estimator is asymptotically normally distributed.

A.3 Proof of Theorem 2.2.7

Proof. The proof for the theorem will have a similar structure to the one used in [65] for Theorem 7.2.

First, we need to make some notations. We take $SR_i := \frac{\mu_i}{\sigma_i}$ be the true Sharpe ratio with its respective true mean and standard deviation for arm i. Similarly, we take $\hat{SR}_{iu_i} := \frac{\hat{\mu}_{iu_i}}{\hat{\sigma}_{iu_i}}$ to be the estimated Sharpe ratio for selecting arm i after u_i observations. Without loss of generality we assume arm 1 to be the optimal one, i.e. $\mu^* = \mu_1$ (and assume the highest Sharpe is correspondent to arm 1 as well).

Define in similar fashion (First unlabeled equation at page 106 from [65]) the sets

$$G_i := \left\{ SR_1 < \min_{t\in[n]} UCB_i^{PSR}(t,p) \right\} \cap \left\{ \hat{SR}_{iu_I} + Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} < SR_1 \right\}.$$

Then, we will bound each term of the regret as

$$\mathbb{E}[T_i(n)] = \mathbb{E}[\mathbf{1}_{\{G_i\}} T_i(n)] + \mathbb{E}[\mathbf{1}_{\{G_i^c\}} T_i(n)] \leq u_i + \mathbb{P}(G_i^c) n. \tag{A.3.1}$$

The bound on the first term follows the same reasoning as in the reference. For the second term, we need to take a look at the complement of G_i:

$$G_i^c = \left\{ SR_1 \geq \min_{t\in[n]} UCB_i^{PSR}(t,p) \right\} \cup \left\{ \hat{SR}_{iu_I} + Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \geq SR_1 \right\}.$$

For the first set of the two, an union bound argument is used to prove that

$$\mathbb{P}(SR_1 \geq \min_{t\in[n]} UCB_i^{PSR}(t,p)) \leq \sum_{s=1}^{n} \mathbb{P}\left(SR_1 \geq \hat{SR}_{1s} + Z_p \hat{\sigma}_{\hat{SR}_{1s}} \sqrt{\frac{1}{s}} \right) \overset{(2.2.3)}{\leq} np. \tag{A.3.2}$$

Now, for the second term we need to bound

$$\mathbb{P}\left(\hat{SR}_{iu_i} + Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \geq SR_1 \right). \tag{A.3.3}$$

Now, the sub-optimal gap is $\Delta_i = \mu_1 - \mu_i$ and the Sharpe ratios are $\frac{\mu_i}{\sigma_i}$. Thus,

$$SR_1 = \frac{\Delta_i + \mu_i}{\sigma_1} \tag{A.3.4}$$

This follows through as

$$\begin{aligned}
&\mathbb{P}\left(\hat{SR}_{iu_i} - \frac{\mu_1}{\sigma_1} \geq -Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \right) \\
&= \mathbb{P}\left(\hat{SR}_{iu_i} - \frac{\mu_i + \Delta_i}{\sigma_1} \geq -Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \right) \\
&= \mathbb{P}\left(\hat{SR}_{iu_i} - \frac{SR_i \sigma_i}{\sigma_1} \geq \frac{\Delta_i}{\sigma_1} - Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \right) \\
&= \mathbb{P}\left(\hat{SR}_{iu_i} - SR_i + \big(1 - \frac{\sigma_i}{\sigma_1}\big) SR_i \geq \frac{\Delta_i}{\sigma_1} - Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \right) \\
&= \mathbb{P}\left(\hat{SR}_{iu_i} - SR_i \geq \frac{\Delta_i}{\sigma_1} - \big(1 - \frac{\sigma_i}{\sigma_1}\big) SR_i - Z_p \hat{\sigma}_{\hat{SR}_{iu_i}} \sqrt{\frac{1}{u_i}} \right) \\
&= \mathbb{P}\left(\sqrt{u_i} \frac{\hat{SR}_{iu_i} - SR_i}{\hat{\sigma}_{\hat{SR}_{iu_i}}} \geq \frac{\Delta_i \sqrt{u_i}}{\sigma_1 \hat{\sigma}_{\hat{SR}_{iu_i}}} - \big(1 - \frac{\sigma_i}{\sigma_1}\big) \frac{SR_i \sqrt{u_i}}{\hat{\sigma}_{\hat{SR}_{iu_i}}} - Z_p \right)
\end{aligned}$$

To simplify the computations (and potentially limit the bounds), we need to make a few remarks and assumptions. Firstly, we need the bound to hold for a high confidence level, so p will need to be very low. Thus, $Z_p = \Phi^{-1}(1-p)$ will be a positive value almost surely. Secondly, since we assumed that arm 1 is the optimal arm, we'll also assume that it has the highest Sharpe ratio, i.e. $1 - \frac{\sigma_i}{\sigma_1} \geq 0$. Lastly, we will assume that the true Sharpe ratios of all arms are positive. With these considerations, we can further simplify the previous expression. We have

$$\begin{aligned}
&\mathbb{P}\bigg(\sqrt{u_i}\frac{\hat{SR}_{iu_i} - SR_i}{\hat{\sigma}_{\hat{SR}_{iu_i}}} \geq \frac{\Delta_i\sqrt{u_i}}{\sigma_1\hat{\sigma}_{\hat{SR}_{iu_i}}} - \big(1 - \frac{\sigma_i}{\sigma_1}\big)\frac{SR_i\sqrt{u_i}}{\hat{\sigma}_{\hat{SR}_{iu_i}}} - Z_p\bigg) \\
&\leq \mathbb{P}\bigg(\sqrt{u_i}\frac{\hat{SR}_{iu_i} - SR_i}{\hat{\sigma}_{\hat{SR}_{iu_i}}} \geq \frac{\Delta_i\sqrt{u_i}}{\sigma_1\hat{\sigma}_{\hat{SR}_{iu_i}}}\bigg).
\end{aligned}$$

We denote the right hand side term of the probability as Q_i. Because the Sharpe ratio estimator is asymptotically normal via the delta method, if we take an u_i (and implicitly an n) large enough, then the left hand quantity will become a standard normal random variable Z. Thus, we can approximate the resulted quantity as

$$\begin{aligned}
\mathbb{P}(Z \geq Q_i) &= \int_{Q_i}^{\infty} \frac{1}{\sqrt{2\pi}}\exp\bigg(-\frac{x^2}{2}\bigg)dx \\
&\leq \frac{1}{Q_i\sqrt{2\pi}}\int_{Q_i}^{\infty} x\exp\bigg(-\frac{x^2}{2}\bigg)dx \\
&\leq \frac{1}{Q_i\sqrt{2\pi}}\exp\bigg(-\frac{Q_i^2}{2}\bigg). \qquad \text{(A.3.5)}
\end{aligned}$$

Thus, we managed to bound the probability defined at (A.3.3) by the quantity derived in (A.3.5). We must now choose an u_i such that is very low relative to n, preferrably to a logarithmic or lesser scale. Note that the Q_i term is very similar to the one derived in the proof from [65], although now we have an additional term, which tightens the bound. To be more specific, the bound derived from literature scales by

$$\exp\bigg(-\frac{u_i\Delta_i^2}{2}\bigg)$$

up to some constants, whereas this bound is scaled by approximatively

$$\frac{1}{\Delta_i\sqrt{u_i}}\exp\bigg(-\frac{u_i\Delta_i^2}{2}\bigg).$$

As u_i increases, our bound decreases faster, so our intuition tells us the overall PSR-derived algorithm should perform better. This is further discussed in Chapter 4.

Putting (A.3.5) and (A.3.2) into (A.3.1), we get

$$\mathbb{E}[T_i(n)] \leq u_i + n\bigg(np + \frac{\sigma_1\hat{\sigma}_{\hat{SR}_{iu_i}}}{\Delta_i\sqrt{u_i}}\exp\bigg(-\frac{\Delta_i^2 u_i}{2\sigma_1^2\hat{\sigma}^2_{\hat{SR}_{iu_i}}}\bigg)\bigg). \qquad \text{(A.3.6)}$$

Notice that similarly, when u_i gets large enough, then

$$\hat{\sigma}_{\hat{SR}_{iu_i}} \longrightarrow \sigma_{SR_i}.$$

The previous convergence will surely happen in this context, since otherwise our estimator wouldn't converge to a normal distribution if u_i wasn't large enough, so we can take equality between the variance estimator and the true variance.

Using the previous expression, we can choose $u_i = \left\lceil \frac{\log(n^2)\,\sigma_1^2\sigma_{SR_i}^2}{\Delta_i^2} \right\rceil$, $p = \frac{1}{n^2}$ and if we substitute it in (A.3.6) and get

$$\begin{aligned}
\mathbb{E}[T_i(n)] &\leq u_i + n\left(np + \frac{\sigma_1 \hat{\sigma}_{\hat{SR}_{iu_i}}}{\Delta_i\sqrt{u_i}}\exp\left(-\frac{\Delta_i^2 u_i}{2\sigma_1^2\hat{\sigma}^2_{\hat{SR}_{iu_i}}}\right)\right) \\
&\leq \left\lceil \frac{\log(n^2)\,\sigma_1^2\sigma_{SR_i}^2}{\Delta_i^2} \right\rceil + 1 + \frac{n}{\sqrt{2\pi\log(n^2)}}\exp\left(-\frac{\log(n^2)}{2}\right) \\
&\leq \frac{\log(n^2)\,\sigma_1^2\sigma_{SR_i}^2}{\Delta_i^2} + 2 + \frac{1}{\sqrt{2\pi\log(n^2)}} \\
&\leq \frac{2\log(n)\,\sigma_1^2\sigma_{SR_i}^2}{\Delta_i^2} + 3
\end{aligned}$$

Thus,

$$\begin{aligned}
R_n &= \sum_{i=1}^{k} \Delta_i \mathbb{E}[T_i(n)] \\
&\leq \sum_{i=1}^{k} \Delta_i \left(\frac{\log(n)\,\sigma_1^2\sigma_{SR_i}^2}{\Delta_i^2} + 3\right) \\
&\leq \sum_{\Delta_i>0} \frac{2\sigma_1^2\sigma_{SR_i}^2\log(n)}{\Delta_i} + 3\sum_{i=1}^{k}\Delta_i \\
&\leq 2\sigma_1^4 \sum_{\Delta_i>0} \frac{\log(n)}{\Delta_i} + 3\sum_{i=1}^{k}\Delta_i
\end{aligned}$$

□

Remark A.3.1. Even though we use a very similar method (Theorem 7.2 [65]) of proving Theorem 2.2.7, there are 2 key differences:

- The asymptotic normality of the Sharpe ratio estimator even in the absence of normal returns tightens the bound because we use a Gaussian random variable in the concentration inequalities.
- We bound the returns sub-optimal gaps, showcased by equation (A.3.4).

Appendix B

Changepoint Detection libraries

B.1 CDP Libraries

Several software packages implement some of the CPD methods. Some of these software packages are listed below:

- changepoint_online: A Collection of Methods for Online Changepoint Detection
  - Description: The changepoint_online package provides efficient algorithms for detecting changes in data streams based on the Focus algorithm. The Focus algorithm solves the CUSUM likelihood-ratio test exactly in $\mathcal{O}(\ln(n))$ time per iteration, where n represents the current iteration. The method is equivalent to running a rolling window (MOSUM) simultaneously for all sizes of windows or the Page-CUSUM for all possible values of the size of change (an infinitely dense grid).
  - Author: Gaetano Romano, Daniel Grose
  - URL: `https://pypi.org/project/changepoint-online/`
  - Citation: [89]
- ocpdet: A Python package for online changepoint detection, implementing state-of-the-art algorithms and a novel approach.
  - Description: OCPDet is an open-source Python package for online changepoint detection, implementing state-of-the-art algorithms and a novel approach, using a scikit-learn style API. Algorithms implemented in ocpdet are:
    - CUSUM: Cumulative Sum algorithm ([84]);
    - EWMA: Exponentially Weighted Moving Average algorithm ([88]);
    - Two Sample tests: Nonparametric hypothesis testing for changepoint detection ([90]);
    - Neural Networks: Novel approach based on sequentially learning neural networks proposed in [54] and extended to online context in [60]
  - Author: Victor Khamesi
  - URL: `https://pypi.org/project/ocpdet/`
  - Citation: [60]
- sdt.changepoint: A changepoint detection module of sdt-python
  - Description: The sdt.changepoint module provides algorithms for changepoin detection, i.e., for finding changepoints in a time series. There are several algorithms available:

    - PELT: a fast offline detection algorithm ([62]);
    - Offline Bayesian changepoint detection ([39]);
    - Online Bayesian changepoint detection ([3])
- Author: Lukas Schrangl
- URL: `https://schuetzgroup.github.io/sdt-python/changepoint.html`
- Citation: [93]

Bibliography

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, *Improved algorithms for linear stochastic bandits*, Advances in neural information processing systems, 24 (2011).

[2] R. P. Adams and D. J. MacKay, *Bayesian online changepoint detection*, arXiv preprint arXiv:0710.3742, (2007).

[3] R. P. Adams and D. J. C. MacKay, *Bayesian online changepoint detection*, 2007.

[4] J. M. Addoum, J. H. Van Binsbergen, and M. W. Brandt, *Asset allocation and managerial assumptions in corporate pension plans*, Available at SSRN 1710902, (2010).

[5] A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire, *Algorithms for portfolio management based on the newton method*, in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 9–16.

[6] R. Alami, *Bayesian change-point detection for bandit feedback in non-stationary environments*, in Asian Conference on Machine Learning, PMLR, 2023, pp. 17–31.

[7] Y. Amihud, H. Mendelson, and R. Wood, *Liquidity and the 1987 stock market crash*, Journal of Portfolio Management, 16 (1990), pp. 65–69.

[8] J.-Y. Audibert and S. Bubeck, *Minimax policies for adversarial and stochastic bandits*, in COLT, 2009, pp. 217–226.

[9] P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite-time analysis of the multi-armed bandit problem*, Machine learning, 47 (2002), pp. 235–256.

[10] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, *The nonstochastic multiarmed bandit problem*, SIAM journal on computing, 32 (2002), pp. 48–77.

[11] Y. Aït-Sahalia and J. Yu, *High frequency market microstructure noise estimates and liquidity measures*, The Annals of Applied Statistics, 3 (2009).

[12] J. Bai and P. Perron, *Computation and analysis of multiple structural change models*, Journal of applied econometrics, 18 (2003), pp. 1–22.

[13] D. Bailey and M. López de Prado, *The sharpe ratio efficient frontier*, The Journal of Risk, 15 (2012), pp. 3–44.

[14] L. Besson, E. Kaufmann, O.-A. Maillard, and J. Seznec, *Efficient change-point detection for tackling piecewise-stationary bandits*, Journal of Machine Learning Research, 23 (2022), pp. 1–40.

[15] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, Journal of Political Economy, 81 (1973), pp. 637–654.

[16] L. Bondesson, *When is the sample mean BLUE?*, Scandinavian Journal of Statistics, 3 (1976), pp. 116–120.

[17] G. Boracchi, D. Carrera, C. Cervellera, and D. Maccio, *Quanttree: Histograms for change detection in multivariate data streams*, in International Conference on Machine Learning, PMLR, 2018, pp. 639–648.

[18] A. Borodin, R. El-Yaniv, and V. Gogan, *Can we learn to beat the best stock*, Advances in Neural Information Processing Systems, 16 (2003).

[19] M. Broadie, *Computing efficient frontiers using estimated parameters*, Annals of operations research, 45 (1993), pp. 21–58.

[20] E. Brynjolfsson, *Artificial intelligence and the modern productivity paradox: a clash of expectations and statistics*, techreport 24001, National Bureau of Economic Research, 1050 Massachusetts Avenue, Cambridge, MA 02138, Nov. 2017. `https://www.nber.org/system/files/working_papers/w24001/w24001.pdf`.

[21] S. Bubeck, N. Cesa-Bianchi, et al., *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*, Foundations and Trends® in Machine Learning, 5 (2012), pp. 1–122.

[22] R. R. Bush and F. Mosteller, *A Stochastic Model with Applications to Learning*, The Annals of Mathematical Statistics, 24 (1953), pp. 559 – 585.

[23] A. Cassel, S. Mannor, and A. Zeevi, *A general framework for bandit problems beyond cumulative objectives*, Mathematics of Operations Research, 48 (2023), pp. 2196–2232.

[24] N. Cesa-Bianchi, K. Eldowa, E. Esposito, and J. Olkhovskaya, *Improved regret bounds for bandits with expert advice*, arXiv preprint arXiv:2406.16802, (2024).

[25] N. Cesa-Bianchi and G. Lugosi, *Combinatorial bandits*, Journal of Computer and System Sciences, 78 (2012), pp. 1404–1422.

[26] H. Chen, Y. He, and C. Zhang, *On interpolating experts and multi-armed bandits*, arXiv preprint arXiv:2307.07264, (2023).

[27] W. Chen, Y. Wang, and Y. Yuan, *Combinatorial multi-armed bandit: General framework and applications*, in International conference on machine learning, PMLR, 2013, pp. 151–159.

[28] W. Chu, L. Li, L. Reyzin, and R. Schapire, *Contextual bandits with linear payoff functions*, in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 208–214.

[29] G. Cornuéjols, J. Peña, and R. Tütüncü, *Optimization Methods in Finance*, Cambridge University Press, July 2018.

[30] D. Cortes, *Adapting multi-armed bandits policies to contextual bandits scenarios*, arXiv preprint arXiv:1811.04383, (2018).

[31] M. Csörgo, *Horváth., l.(1997) limit theorems in change-point analysis*, 1954.

[32] P. S. Dalton, V. H. Gonzalez Jimenez, and C. N. Noussair, *Exposure to poverty and productivity*, PLOS ONE, 12 (2017), p. e0170231.

[33] P. Das, *Online convex optimization and its application to online portfolio selection*, (2014).

[34] P. Das, N. Johnson, and A. Banerjee, *Online lazy updates for portfolio selection with transaction costs*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 27, 2013, pp. 202–208.

[35] R. Degenne and V. Perchet, *Combinatorial semi-bandit with known covariance*, Advances in Neural Information Processing Systems, 29 (2016).

[36] V. DeMiguel, L. Garlappi, F. J. Nogales, and R. Uppal, *A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms*, Management science, 55 (2009), pp. 798–812.

[37] J. Dodson, *Why is it so hard to estimate expected returns?* `https://www-users.cse.umn.edu/~dodso013/docs/dodson2012-lambda.pdf`, Apr. 2012.

[38] Y. Du, S. Wang, Z. Fang, and L. Huang, *Continuous mean-covariance bandits*, Advances in Neural Information Processing Systems, 34 (2021), pp. 875–886.

[39] P. Fearnhead, *Exact and efficient bayesian inference for multiple changepoint problems*, Statistics and Computing, 16 (2006), pp. 203–213.

[40] P. Fearnhead and Z. Liu, *On-line inference for multiple changepoint problems*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 69 (2007), pp. 589–605.

[41] C. Ferguson and J. R. Moroney, *The sources of change in labor's relative share: a neoclassical analysis*, Southern Economic Journal, (1969), pp. 308–322.

[42] A. Fomin, A. Korotayev, and J. Zinkina, *Negative oil price bubble is likely to burst in march - may 2016. a forecast on the basis of the law of log-periodical dynamics*, 2016.

[43] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of computer and system sciences, 55 (1997), pp. 119–139.

[44] A. Ghahtarani, A. Saif, and A. Ghasemi, *Robust portfolio selection problems: a comprehensive review*, Operational Research, 22 (2022), pp. 3203–3264.

[45] J. K. Ghosh, *Bayes solutions in sequential problems for two or more terminal decisions and related results*, Calcutta Statistical Association Bulletin, 13 (1964), pp. 101–122.

[46] C. Granger, *Long memory relationships and the aggregation of dynamic models*, Journal of Econometrics, 14 (1980), pp. 227–238.

[47] L. Györfi, G. Lugosi, and F. Udina, *Nonparametric kernel-based sequential investment strategies*, Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, 16 (2006), pp. 337–357.

[48] Z. Harchaoui, E. Moulines, and F. Bach, *Kernel change-point analysis*, Advances in neural information processing systems, 21 (2008).

[49] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth, *On-line portfolio selection using multiplicative updates*, Mathematical Finance, 8 (1998), pp. 325–347.

[50] D. V. Hinkley and E. A. Hinkley, *Inference about the change-point in a sequence of binomial variables*, Biometrika, 57 (1970), pp. 477–488.

[51] M. Hoffman, E. Brochu, N. De Freitas, et al., *Portfolio allocation for bayesian optimization.*, in UAI, 2011, pp. 327–336.

[52] P.-H. Hsu, Q. Han, W. Wu, and Z. Cao, *Asset allocation strategies, data snooping, and the 1 / n rule*, Journal of Banking Finance, 97 (2018), pp. 257–269.

[53] X. Huo and F. Fu, *Risk-aware multi-armed bandit problem with application to portfolio selection*, Royal Society open science, 4 (2017), p. 171377.

[54] M. Hushchyn and A. Ustyuzhanin, *Generalization of change-point detection in time series data based on direct density ratio estimation*, (2020).

[55] S. Ito, D. Hatano, H. Sumita, A. Yabe, T. Fukunaga, N. Kakimura, and K.-i. Kawarabayashi, *Regret bounds for online portfolio selection with a cardinality constraint*, Advances in Neural Information Processing Systems, 31 (2018).

[56] A. Kalai, S. Chen, A. Blum, and R. Rosenfeld, *On-line algorithms for combining language models*, in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258), vol. 2, IEEE, 1999, pp. 745–748.

[57] S. Kale, *Multiarmed bandits with limited expert advice*, in Conference on learning theory, PMLR, 2014, pp. 107–122.

[58] A. Kazerouni, M. Ghavamzadeh, Y. Abbasi Yadkori, and B. Van Roy, *Conservative contextual linear bandits*, Advances in Neural Information Processing Systems, 30 (2017).

[59] M. G. Kendall and A. B. Hill, *The analysis of economic time-series-part i: Prices*, Journal of the Royal Statistical Society. Series A (General), 116 (1953), p. 11.

[60] V. Khamesi, *ocpdet: A python package for online changepoint detection in univariate and multivariate data*, 2022.

[61] S. Khurshid, M. S. Abdulla, and G. Ghatak, *Optimizing sharpe ratio: Risk-adjusted decision-making in multi-armed bandits*, arXiv preprint arXiv:2406.06552, (2024).

[62] R. Killick, P. Fearnhead, and I. A. Eckley, *Optimal detection of changepoints with a linear computational cost*, Journal of the American Statistical Association, 107 (2012), pp. 1590–1598.

[63] H. Konno and H. Yamazaki, *Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market*, Management science, 37 (1991), pp. 519–531.

[64] T. Lai and H. Robbins, *Asymptotically efficient adaptive allocation rules*, Advances in Applied Mathematics, 6 (1985), pp. 4–22.

[65] T. Lattimore and C. Szepesvári, *Bandit algorithms*, Cambridge University Press, 2020.

[66] M. Lavielle and G. Teyssière, *Adaptive Detection of Multiple Change-Points in Asset Price Volatility*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 129–156.

[67] B. Li and S. C. Hoi, *Online portfolio selection: A survey*, ACM Computing Surveys (CSUR), 46 (2014), pp. 1–36.

[68] B. Li, P. Zhao, S. C. Hoi, and V. Gopalkrishnan, *Pamr: Passive aggressive mean reversion strategy for portfolio selection*, Machine learning, 87 (2012), pp. 221–258.

[69] N. Littlestone and M. K. Warmuth, *The weighted majority algorithm*, Information and computation, 108 (1994), pp. 212–261.

[70] A. W. Lo and A. C. MacKinlay, *Data-snooping biases in tests of financial asset pricing models*, Working Paper 3001, National Bureau of Economic Research, June 1989.

[71] G. Lorden, *Procedures for reacting to a change in distribution*, The annals of mathematical statistics, (1971), pp. 1897–1908.

[72] H. Luo, C.-Y. Wei, A. Agarwal, and J. Langford, *Efficient contextual bandits in non-stationary worlds*, in Conference On Learning Theory, PMLR, 2018, pp. 1739–1776.

[73] B. Mandelbrot, *The variation of certain speculative prices*, The Journal of Business, 36 (1963), p. 394.

[74] H. Markowitz, *Portfolio selection*, The Journal of Finance, 7 (1952), p. 77.

[75] D. S. Matteson and N. A. James, *A nonparametric approach for multiple change point analysis of multivariate data*, Journal of the American Statistical Association, 109 (2014), pp. 334–345.

[76] E. Mertens, *Comments on variance of the IID estimator in Lo (2002)*, techreport, University of Basel, Nov. 2002.

[77] R. C. Merton, *On estimating the expected return on the market*, Journal of Financial Economics, 8 (1980), pp. 323–361.

[78] W. C. Mitchell, *Making and Using of Index Numbers*, Harvard University Press, 1915.

[79] A. Mostafa and A. Ghorbal, *Bayesian and non-bayesian analysis for random change point problem using standard computer packages*, International Journal Of Mathematical Archive, 2 (1963), p. 2011.

[80] A. S. Nemirovskij and D. B. Yudin, *Problem complexity and method efficiency in optimization*, (1983).

[81] B. O'Neill, *Some useful moment results in sampling problems*, The American Statistician, 68 (2014), pp. 282–296.

[82] M. A. Osborne, R. Garnett, and S. J. Roberts, *Gaussian processes for global optimization*, (2009).

[83] G. Ozkaya and Y. Wang, *Multi-armed bandit approach to portfolio choice problem.* https://thevoice.bse.eu/2020/09/16/multi-armed-bandit-approach-portfolio-choice-problem/.

[84] E. S. Page, *Continuous inspection schemes*, Biometrika, 41 (1954), p. 100.

[85] P. Perrault, M. Valko, and V. Perchet, *Covariance-adapting algorithm for semi-bandits with application to sparse outcomes*, in Conference on Learning Theory, PMLR, 2020, pp. 3152–3184.

[86] R. Quinteiro, F. S. Melo, and P. A. Santos, *Limited depth bandit-based strategy for monte carlo planning in continuous action spaces*, arXiv preprint arXiv:2106.15594, (2021).

[87] H. Robbins, *Some aspects of the sequential design of experiments*, (1952).

[88] S. W. Roberts, *Control chart tests based on geometric moving averages*, Technometrics, 1 (1959), pp. 239–250.

[89] G. Romano, I. Eckley, P. Fearnhead, and G. Rigaill, *Fast online changepoint detection via functional pruning cusum statistics*, Journal of Machine Learning Research, 24 (2023), pp. 1–36.

[90] G. J. Ross and N. M. Adams, *Two nonparametric control charts for detecting arbitrary distribution changes*, Journal of Quality Technology, 44 (2012), pp. 102–116.

[91] D. Russo and B. Van Roy, *An information-theoretic analysis of thompson sampling*, Journal of Machine Learning Research, 17 (2016), pp. 1–30.

[92] Y. Saatçi, R. D. Turner, and C. E. Rasmussen, *Gaussian process change point models*, in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 927–934.

[93] L. Schrangl, *Quantifying conformational dynamics of biomolecules via single-molecule FRET*, PhD thesis, Technische Universität Wien, 2020.

[94] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al., *Mastering atari, go, chess and shogi by planning with a learned model*, Nature, 588 (2020), pp. 604–609.

[95] M. Sewell, *Characterization of financial time series*, (2006).

[96] W. F. Sharpe, *Mutual fund performance*, The Journal of Business, 39 (1966), p. 119.

[97] W. F. Sharpe, *The sharpe ratio*, The Journal of Portfolio Management, 21 (1994), pp. 49–58.

[98] W. Shen and J. Wang, *Portfolio selection via subset resampling*, Proceedings of the AAAI Conference on Artificial Intelligence, 31 (2017).

[99] W. Shen, J. Wang, Y.-G. Jiang, and H. Zha, *Portfolio choices with orthogonal bandit learning*, in Twenty-fourth international joint conference on artificial intelligence, 2015.

[100] A. Shiryayev, *Some precise formulas in change-point problems*, Theory of Probability and its Applications, 10 (1963), pp. 380–385.

[101] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., *Mastering the game of go with deep neural networks and tree search*, nature, 529 (2016), pp. 484–489.

[102] D. Smith, *Trends and Causes of Armed Conflict*, VS Verlag für Sozialwissenschaften, 2004, pp. 111–127.

[103] J. Sween and D. T. Campbell, *The interrupted time series as quasi-experiment: Three tests of significance. a fortran program for the cdc 3400 computer.*, (1965).

[104] W. R. Thompson, *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*, Biometrika, 25 (1933), pp. 285–294.

[105] E. O. Thorp, *Portfolio choice and the kelly criterion*, in Stochastic optimization models in finance, Elsevier, 1975, pp. 599–619.

[106] G. J. Van den Burg and C. K. Williams, *An evaluation of change point detection algorithms*, arXiv preprint arXiv:2003.06222, (2020).

[107] G. U. Yule, *Why do we sometimes get nonsense-correlations between time-series?–a study in sampling and the nature of time-series*, Journal of the Royal Statistical Society, 89 (1926), p. 1.