Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria
Doctoral Programme in Information Technology

Online Learning Methods for Pricing and Advertising

Doctoral Dissertation of:
Marco Mussi

Supervisor:
Prof. Marcello Restelli

Co-supervisor:
Dott. Alberto Maria Metelli

Tutor:
Prof. Nicola Gatti

2023 – Cycle XXXVI

Abstract

NOWADAYS, when it comes to selling a product online, there are several key factors that require careful consideration. Two of the most significant factors are the pricing strategy and the investments in advertising. When determining the price of a product, it is essential to strike a balance. The price should neither be set too low, as this would result in a reduced revenue from the single sale, nor too high, as it may deter potential buyers. The amount of money we invest in advertising should be balanced to let people know our offer without overspending or reaching people who are not interested. These two aspects are usually handled disjointedly by humans, but this, even if we proceed to optimize for the two components individually, may lead to a suboptimal solution. In this thesis, we focus on the adoption of online learning to solve the task of finding the optimal price for a product and how to advertise it properly. This thesis encompasses various facets of pricing and advertising, offering both theoretical frameworks and practical solutions for addressing the associated challenges. The first part of the thesis faces pricing methods. Initially, we introduce a practical and efficient approach tailored to e-commerce pricing. This method empowers e-commerce businesses to price properly the long tail. Subsequently, our focus shifts to theoretical aspects of pricing, particularly emphasizing the problem of learning temporal dynamics. In the second part, we discuss the theoretical aspects of advertising, with a particular focus on marketing mix models and how to handle them in a tractable way. In the third and final part, we bring together the problems of pricing and advertising, presenting a unified approach to address both aspects

concurrently. This integrated approach allows us to strive for efficient and optimal solutions in the complex landscape of online product sales.

Sommario

Al giorno d'oggi, un e-commerce che vuole commercializzare un prodotto online deve tenere in considerazione diversi aspetti per poterlo fare al meglio. Due dei fattori più significativi sono la strategia di selezione del prezzo ottimale e gli investimenti pubblicitari. Nel determinare il prezzo per un prodotto, è essenziale trovare un giusto equilibrio. Il prezzo non dovrebbe essere troppo basso, poiché ciò comporterebbe una riduzione del ricavo dalla singola vendita, né troppo alto, poiché potrebbe scoraggiare potenziali acquirenti. L'importo in denaro che investiamo nella pubblicità dovrebbe essere bilanciato per far conoscere la nostra offerta senza spendere eccessivamente ed evitando di raggiungere persone non interessate. Solitamente, questi due aspetti sono gestiti separatamente dagli esseri umani, ma ottimizzare i due componenti individualmente, potrebbe portare a una soluzione non ottimale. In questa tesi, ci concentriamo sull'adozione dell'apprendimento online per risolvere il problema di trovare il prezzo ottimale per un prodotto e di come pubblicizzarlo correttamente. Questa tesi comprende vari aspetti relativi a *pricing* e *advertising*, offrendo sia prospettive teoriche che soluzioni pratiche per affrontare le sfide associate. La prima parte della tesi approfondisce i metodi di *pricing*. Inizialmente, presentiamo un approccio pratico ed efficiente per il *pricing* ottimale dei prodotti in vendita su siti di e-commerce che consente di prezzare coerentemente la long tail. Successivamente, l'attenzione si sposta sugli aspetti teorici del *pricing*, con particolare enfasi sul problema dell'apprendimento delle dinamiche temporali. Nella seconda parte, discutiamo gli aspetti teorici dell'*advertising*, con particolare attenzione al

marketing mix model, il cui scopo è trovare il giusto *mix* di campagne di diverso tipo per massimizzare le vendite. Nella terza e ultima parte, uniamo i problemi dell'*optimal pricing* e dell'*advertising*, presentando un approccio unificato per affrontare entrambi gli aspetti contemporaneamente. Questo approccio integrato ci consente di cercare soluzioni efficienti ed ottimali nel complesso panorama delle vendite online dei prodotti.

Contents

List of Symbols and Notation

Symbols

Symbol	Meaning
x	Constants
$\mathbf{x}$	Vectors
$\mathbf{A}$	Matrices
$\mathcal{A}$	Sets
$\mathbf{I}_n$	Identity matrix of dimension n
$\mathbf{0}_n$	Vector of all zeros of dimension n
$\mathbb{1}\{e\}$	Indicator function for event e

Notation

Notation	Meaning
$\llbracket a \rrbracket$	$\{1, 2, \ldots, a\}$
$\llbracket a, b \rrbracket$	$\{a, a+1, \ldots, b\}$ (with $a \leqslant b$)
$\llbracket a, \infty \rrparenthesis$	$\{a, a+1, \ldots\}$
$\lvert\mathcal{A}\rvert$	Cardinality of set $\mathcal{A}$
$\langle \mathbf{x}, \mathbf{y} \rangle$	$\mathbf{x}^{\mathsf{T}}\mathbf{y} = \sum_{i=1}^{n} x_i y_i$ (with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ real-valued vectors)
$\lVert\mathbf{x}\rVert_{\mathbf{A}}^2$	$\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x}$ (with $\mathbf{A} \in \mathbb{R}^{n \times n}$ positive semidefinite)
$\boldsymbol{\lambda}(\mathbf{A})$	Vector of the eigenvalues of $\mathbf{A}$
$\rho(\mathbf{A})$	$\max\{\lvert\boldsymbol{\lambda}(\mathbf{A})\rvert\}$ (*spectral radius* of $\mathbf{A}$)
$\lvert\lvert\mathbf{A}\rvert\rvert_2$	$\sqrt{\max\{(\boldsymbol{\lambda}(\mathbf{A}^{\mathsf{T}}\mathbf{A})\}}$ (*spectral norm* of $\mathbf{A}$)
$\Phi(\mathbf{A})$	$\sup_{\tau \geqslant 0} \lVert\mathbf{A}^{\tau}\rVert_2 / \rho(\mathbf{A})^{\tau}$ (see Oymak and Ozay, 2019)

Preface

This thesis collects several works made during these three years of doctorate. All the original contributions presented in this dissertation were developed in joint work with Marcello Restelli together with other co-authors.

The *Dynamic Pricing for the Long Tail*, discussed in Chapter 4, is a joint work with Gianmarco Genalti, Francesco Trovó, Alessandro Nuara, and Nicola Gatti. The work presented in this chapter (Mussi et al., 2022a) is published at the *ACM Conference on Knowledge Discovery and Data Mining (KDD)* and also appeared (Genalti et al., 2022) at the *European Workshop on Reinforcement Learning (EWRL)*.

The *Autoregressive Bandits*, discussed in Chapter 5, is a joint work with Francesco Bacchiocchi, Gianmarco Genalti, Davide Maran, Nicola Gatti, and Alberto Maria Metelli. This work (Bacchiocchi et al., 2024) is published at the *International Conference on Artificial Intelligence and Statistics (AISTATS)* and a preliminary version (Bacchiocchi et al., 2023) appeared at the *European Workshop on Reinforcement Learning (EWRL)*.

The *Dynamical Linear Bandits*, discussed in Chapter 7, is a joint work with Alberto Maria Metelli. This work (Mussi et al., 2023a) is published at the *International Conference on Machine Learning (ICML)* and a preliminary version (Mussi et al., 2022b) appeared at the *Complex Feedback in Online Learning Workshop*.

The *Factored Rewards Bandits*, proposed in Chapter 8, is a joint work with Simone Drago and Alberto Maria Metelli. This work (Mussi et al., 2024) is published at the *International Conference on Machine Learning (ICML)* and a preliminary version (Drago et al., 2024) appeared at the *Adaptive and Learning Agents Workshop*.

CHAPTER 1

Introduction

Motivated by the rapid increase in the quantity of data and the exponential growth of online platforms, companies are continually seeking innovative strategies to enhance their market presence, capture consumer attention, and optimize their pricing models. Machine Learning (ML) has emerged in recent years as a groundbreaking transformative force, empowering organizations to revolutionize the way they price products and promote them through advertising. The traditional paradigms of pricing and advertising, once reliant on static models and generalized strategies, are rapidly giving way to data-driven, adaptive approaches powered by ML algorithms.

1.1 What is Machine Learning?

Machine Learning (ML, Bishop, 2006) is a field within the broader world of Artificial Intelligence (AI, Russell and Norvig, 2021). The term "Machine Learning" is an "*umbrella*" term that encompasses a wide range of techniques and methodologies designed to enable machines to learn from data and make predictions or decisions without being specifically programmed. This term is used to enclose various learning paradigms, including *supervised learning*, *unsupervised learning*, and *reinforcement learning*.

Supervised learning is the first branch of ML. It involves training models using *labeled* data. A supervised learning algorithm learns to map inputs and outputs. This branch encloses regression and classification algorithms. It relies on a structured dataset for training, making it a powerful tool for solving specific problem domains.

Unsupervised learning, on the other hand, deals with *unlabeled* data and focuses on discovering patterns and structures within the data. This branch of ML includes techniques such as clustering and dimensionality reduction. Unsupervised learning is especially valuable when the data lack clear labels or when exploring large and complex datasets.

Reinforcement Learning (RL) is the last and the most complex branch of ML. It revolves around training *agents* to make *sequential decisions* in an unknown *environment*. RL poses additional challenges w.r.t. supervised and unsupervised learning as we usually have to understand how to actively collect data to properly learn what is around our agent.

1.2 Why Online Learning and Multi-Armed Bandits?

In this thesis, we face the challenge of solving sequential decision-making problems, specifically in the context of dynamic pricing and advertising budget optimization. These problems are a specific kind of RL problems in which we want to make decisions online, at every step, on the action to perform (e.g., a price to set, a budget to invest). These scenarios in which we make decisions step-by-step are referred as *online learning* problems. In making these decisions, we have to take into account two different objectives at each time step. Indeed, we want both to use (*exploit*) the information we have to select the most profitable actions, but also discover (*explore*) new options that may be more profitable w.r.t. the ones known until now. Given that our knowledge is due to the actions we perform, online learning algorithms have the difficult task of balancing this trade-off called *exploration-exploitation dilemma*. Within the domain of online learning, we encounter two primary categories of algorithms: online reinforcement learning algorithms, which are grounded in the framework of *Markov Decision Processes* (MDPs, Puterman, 2014), and Multi-Armed Bandits (MABs, Lattimore and Szepesvári, 2020). Online RL is well-suited for handling more complex scenarios, where these algorithms can optimize a specific metric, typically referred to as the *reward*, over *states/actions trajectories*, and in which the evolution of the state is influenced by our action. In contrast, the conventional definition of MAB lacks the concept of states, meaning that actions impact only the rewards.

In this thesis, we choose to adopt MABs for several reasons, both due to the problems we face and the characteristics of the bandits that make them suited for us. The main advantages of MABs w.r.t. reinforcement learning methods are:

- *Simplicity:* MAB algorithms are simpler and computationally lighter w.r.t. online RL ones. They are particularly effective when limited computational resources are available, or we need to make quick decisions in real time. Online reinforcement learning can be more complex and requires training a full reinforcement learning agent, which can be an overkill for simpler decision-making problems.
- *Explainability:* MABs behavior is more prone to be understood by human actors due to their simplicity. Indeed, the exploration strategy is designed with a clear trade-off between exploring new actions and exploiting known actions. The agent's choice to explore new actions or exploit a known arm is easier to explain and understand compared to the exploration strategies in RL, which can be more dynamic and complex.
- *Theoretical Guarantees:* MABs are designed to handle the exploration-exploitation trade-off efficiently. Their theoretical guarantees can be studied to characterize and optimize their behavior, thanks to the immediate feedback that simplifies the learning process and facilitates the design of algorithms with provable guarantees. On the other hand, rewards in RL may be delayed, sparse, or noisy, making it more challenging to attribute rewards to specific actions and assess the agent's performance in the short term.

1.3 Original Contributions and Overview

In this dissertation, we face the problem of online decision-making in the context of dynamic pricing and advertising budget optimization. These two topics are, indeed, two sides of the same coin. In order to sell a product, one must be able to select both the price that is proper for the reference market and advertise it properly. The price should neither be set too low, as this would result in reduced revenue from the single sale, nor too high, as it may deter potential buyers. The amount of money we invest in advertising should be balanced to let the people know of us without overspending and reaching people who are not interested in what we are selling. The goal, indeed, is to maximize the combination of pricing and advertising policies to increase our revenue.

We focus on three main topics that correspond to the three parts in which this thesis is structured: dynamic pricing, advertising optimization, and joint optimization of pricing and advertising. The three parts are all binded each other from (i) the scope of the algorithms proposed, whose final goal in all the cases is to improve the revenues due to the sales we perform, and (ii) the methodology used to pursue the goal, as all the algorithms presented in this thesis are based on Multi-Armed Bandits.

Before diving into the details of pricing and advertising, we start the thesis by summarizing in Chapter 2 all the fundamental notions of online learning and MABs that we need in order to understand the content of the next chapters.

1.3.1 Part I: Dynamic Pricing

In Part I, we first present, in Chapter 3, an overview of dynamic pricing, and we introduce all the fundamental notions on this topic. In Chapter 4, we present (Mussi et al., 2022a) a practical algorithm to perform dynamic pricing in the scenario of an e-commerce website that wants to price both popular and long-tail product in order to maximize its turnover. The algorithm we propose is tested on a large e-commerce selling a wide range of products with different kinds of customers. The experimental campaign we conducted demonstrates the empirical soundness of the proposed solution. Then, we move to a more theoretical aspect of pricing. In Chapter 5, we go beyond and formulate a new approach (Bacchiocchi et al., 2024) for handling dynamic pricing using MAB methods that take into account temporal dependences through the introduction of autoregressive processes to model such a dependency. Such processes are useful to represent the smooth trends that are not captured by standard MABs, but represent a phenomenon that cannot be ignored. We theoretically characterize our solution, and we discuss its soundness. Even in this case, we conducted simulations by applying our solution to a real-world dataset that has been generalized in order to be used to test this environment.

1.3.2 Part II: Advertising

In Part II, we first overview, in Chapter 6, all the scenarios in which machine learning can be adopted in advertising. Then, we focus on the problem of budget optimization in online advertising, we revise the fundamental notions in this field, and in particular, we focus our attention on the problem of optimal budget allocation for Marketing Mix Models (MMMs). Then, in Chapter 7, we go to the core of the contribution for this part. We propose (Mussi et al., 2023a) a theoretical framework to face the problem of optimizing the budget in MMM online. The method learns the best combination of campaigns in order to optimize the target metric. For this setting, we provide a lower bound on the regret, and we provide an optimistic algorithm with regret guarantees, balancing the customary exploration-exploitation trade-off. The algorithm we developed is demonstrated to be efficient under several KPI under analysis, and is tested in a simulation environment based on real-world data from an MMM use case.

1.3.3 Part III: Joint Pricing and Advertising

In Part III, we face the problem of jointly optimizing the price at which we want to sell an item and the expenditure to advertise it. This part presents a unique chapter, Chapter 8, as the fundamentals of pricing and advertising are already provided in Chapters 3 and 6, respectively. We propose a new setting for handling the problem in which the reward can be factorized and intermediate observations are available. After having theoretically characterized the setting, we provide two algorithms with different peculiarities, and we studied their theoretical guarantees. We test our solutions to verify the goodness and the results are in favor of our solution w.r.t. the state of the art in this field.

Finally, in Chapter 9, we summarize all the results presented in this dissertation, and we draw some possible future directions.

CHAPTER 2

Foundations of Online Decision-Making

In Chapter 1, we mentioned online learning and MABs in particular as the class of approaches that we consider in this thesis to handle sequential decision-making problems. With online learning, we usually refer to the branch of machine learning facing the problem of statistical learning in an online manner. Compared to supervised learning, this kind of algorithms presents additional challenges, as we have to understand how to (i) properly learn from data and (ii) how to actively collect new samples in order to improve our knowledge. While we perform our actions, we have to deal with the so-called *exploration-exploitation dilemma*. This implies that we have to select actions balancing the knowledge we have and the consequent willingness to exploit it with the request for more knowledge that we intrinsically must have in order to reach the optimal solution.

Sequential decision-making and online learning in particular, contains within it two classes of algorithms: Reinforcement Learning (RL, Sutton and Barto, 2018) and Multi-Armed Bandits (MABs, Lattimore and Szepesvári, 2020). In Reinforcement Learning, we have to choose a path, a sequence of actions, that leads us to the optimal solution. In Multi-Armed Bandits, the rounds are usually isolated and the effect of an action usually holds only for one round. Given that the literature on online decision-

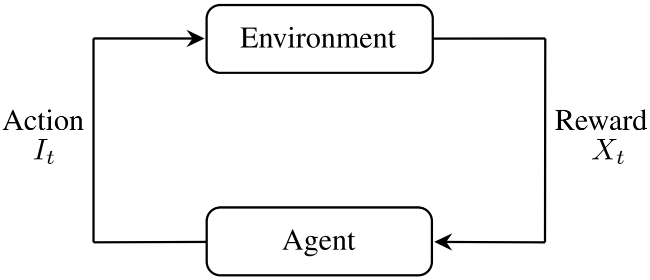


Figure 2.1: *MAB interaction scheme.*

making is extremely wide, in this chapter, we will present only a selection of the fundamental notions useful to understand the works that will be discussed in the next chapters of this thesis.

Chapter Outline This chapter is structured as follows. First, in Section 2.1, we present the MABs' basic notions and the interaction scheme. Then, in Section 2.2, we classify the most famous kinds of MABs given their characteristics. In Section 2.3, we present how to evaluate the performance of an online decision-making algorithm and the possible objectives. Finally, in Section 2.4, we present the more widely adopted algorithms, and we briefly discuss their characteristics and theoretical guarantees.

2.1 Overview on MABs

Multi-Armed Bandits (Lattimore and Szepesvári, 2020) are a class of algorithms in which we consider an agent performing action, which interacts with an environment. The latter, at every time step $t \in [\![T]\!]$ (where $T \in \mathbb{N}$ is the *time horizon* or *time budget*), takes as input the action $I_t \in [\![k]\!]$ (also called *arm*), and gives back a *reward* $X_t \in \mathbb{R}$. A representation of this simple interaction scheme is provided in Figure 2.1. The peculiarity of this framework stands in the fact that we receive the feedback (i.e., the reward) only for the action we are performing, and we get no information about what would have happened if we had chosen another action.[1] This makes it necessary to manage the exploration-exploitation trade-off discussed above. Indeed, with no information on the arms not pulled, in a noisy environment, if we wrongly choose the optimum, and if we exploit it forever, we will constantly pay a penalty due to the suboptimal choice.

[1] This kind of feedback is also called *bandit feedback*. If we get the feedback also on the non-performed actions, we talk about *full* or *expert feedback*.

2.2 Fundamental Dichotomies of MABs

In the previous section, we talked about the concept of reward in an informal way. Now, we provide a more precise classification of the various classes of MABs given their reward, action space, and the presence of a state.

2.2.1 Classification due to the Rewards

Formally, the reward at time t is a scalar $X_t \in \mathbb{R}$. Given the peculiarities of the *entity* generating it, we can classify MABs as follows. The first important dichotomy about the reward is the one between *stochastic* and *adversarial* MABs. The second classification we analyze is due to the dependency of the reward on *time*.

Stochastic and Adversarial Rewards In the *stochastic* setting, when an arm $i \in [\![k]\!]$ is played, the agent observes a feedback $X \sim \nu_i$ (i.e., the reward) sampled from the probability distribution ν_i with expected value μ_i (i.e., the expected reward). The rewards are usually classified into two categories: *subgaussian* (unbounded) and *bounded* random variables. In the former, we assume an additive noise model, i.e., $X = \mu_i + \epsilon$, where μ_i is the expected return, usually assumed to be bounded (e.g., in $[0, 1]$) and ϵ is a zero-mean σ^2-subgaussian random variable, independent conditioned to the past.[2] In the latter, we assume that when we pull arm i we get $X \in [0, 1]$ drawn from a bounded (e.g., Bernoulli) distribution with expected value μ_i.

In the *adversarial* setting, when an arm $i \in [\![k]\!]$ is played, the agent observes a feedback that is chosen a priori from an adversary, which is supposed to select all the values of the reward before the interaction starts.

Stationary and Non-stationary Rewards The reward that an agent observes after the pull of a specific arm can be based on a distribution which can be fixed over time, or time-dependent. In the case in which for every arm the reward distribution does not change over time, we can talk about *stationary* setting, while we talk about *non-stationary* or *time-dependent* setting if at least one arm changes the distribution generating its payoffs over time.

In this thesis, we focus on stationary stochastic σ^2-subgaussian MABs.

[2] We recall that a zero-mean random variable ϵ is σ^2-subgaussian if it holds that $\mathbb{E}[\exp(\xi\epsilon)] \leqslant \exp(\sigma^2\xi^2/2)$ for every $\xi \in \mathbb{R}$.

2.2.2 Classification due to the Action Spaces

In the customary multi-armed bandit setting, we consider a *finite* number of arms that are assumed to be isolated, so no relation between arms is present (Lattimore and Szepesvári, 2020). However, in the last two decades, several approaches have been proposed to handle bandits with *infinitely many* arms. In these scenarios of infinite arms, the first point we have to face is the way in which we want to impose a correlation between the arms.

A first example of how we can consider such correlation is by imposing structure between arms using *non-parametric* methods (Srinivas et al., 2010) where a *kernel* defines the distance betweeen two actions and the influences that they have each other. Another widely known example of that is the *Linear Bandit* setting (Abbasi-Yadkori et al., 2011). In this setting, the agent chooses an action vector $\mathbf{a} \in \mathcal{A} \subset \mathbb{R}^d$ and receives a reward $X = \langle \boldsymbol{\theta}, \mathbf{a} \rangle + \epsilon$ where $\boldsymbol{\theta} \in \mathbb{R}^d$ is unknown to the learner, and ϵ is a zero-mean σ^2-subgaussian random variable, independent conditioned to the past. In this case, the structure between arms is a consequence of the linear structure of the rewards.

2.2.3 Classification due to the Presence of a State/Context

Another dichotomy in MABs is due to the presence of a state, also called *context* in MAB scenarios. The context is a vector embedding information about the situation in which the environment we are facing is in. It influences, together with the action we choose, the reward. In this interaction framework, before we are asked to pick an action from our action space, we receive information about the state in which we are. The goal is, given the state/context provided, to select the best action for such a context.

At this point, with this notion of state, we may ask ourselves what is the difference between a contextual MAB and a reinforcement learning problem. The difference lies in the fact that in RL, we are concerned about optimizing the entire trajectory, since the next state (as well as the reward) depends on the action we choose (and the current state) (Sutton and Barto, 2018). In contextual MAB, on the other hand, the next state (context) does not depend on the current state and action but is assumed to be sampled from a certain distribution, independent from the choices of the learner which is interacting with the environment.

2.3 Performance Evaluation in Stochastic MAB

MAB algorithms can have two different objectives, namely *best-arm identification* (a.k.a. *pure exploration*) and *regret minimization*. These two objectives are in contrast to each other and require different solutions (Bubeck et al., 2009).

2.3.1 Best-Arm Identification

The goal of Best-Arm Identification (BAI) is to find the best arm i^*, i.e., the one with the highest expected value. We have two possible scenarios in this case: *fixed-budget* BAI and *fixed-confidence* BAI.

Fixed-Budget BAI In this scenario, we are provided with a time budget T for our learning process. The goal is to minimize the probability of indicating at the end of such a time budget the wrong arm as the best arm. Formally, given an algorithm that recommends $\hat{I}^* \in [\![k]\!]$ at the end of the learning process, we measure its performance with the *error probability*, i.e., the probability of recommending a suboptimal arm at the end of the time budget T:

$$e_T := \mathbb{P}(\hat{I}^* \neq i^*).$$

Fixed-Confidence BAI In this scenario, we are required to provide the best arm $\hat{I}^* \in [\![k]\!]$ with a confidence $\delta \in (0, 1)$. The goal is to minimize the number of samples (i.e., the required time budget T) to indicate at the end of the learning process the best arm with a given confidence of at least $1 - \delta$.

2.3.2 Regret Minimization

The goal of regret minimization in the case of stochastic payoffs is, given a time horizon, to keep the cumulative loss w.r.t. the best possible action as low as possible.[3] Defining μ^* as the expected value of the optimal action, the *cumulative regret* over a time horizon T is:

$$R(T) := \sum_{t=1}^{T} (\mu^* - \mu_{I_t}) = \sum_{t=1}^{T} \Delta_{I_t}, \tag{2.1}$$

where $\Delta_{I_t} = \mu_1 - \mu_{I_t}$ is the so called *suboptimality gap*.

[3] For the adversarial setting, the regret is computed w.r.t. the *best fixed action,* i.e., the one with the highest cumulative reward.

More formally, to evaluate a policy $\boldsymbol{\pi}$ induced by an algorithm given a bandit instance $\boldsymbol{\nu} = (\nu_i)_{i=1}^k$, we want to look at its expected cumulative regret:

$$\mathop{\mathbb{E}}_{\boldsymbol{\nu}}[R(\boldsymbol{\pi}, T)] := \mathop{\mathbb{E}}_{\boldsymbol{\nu}}\left[\sum_{t=1}^{T}(\mu^* - \mu_{I_t})\right] = \mathop{\mathbb{E}}_{\boldsymbol{\nu}}\left[\sum_{t=1}^{T}\Delta_{I_t}\right], \tag{2.2}$$

where the expectation is taken w.r.t. the randomness of the rewards and the possible randomness of the policy/algorithm.

Given the definition of expected cumulative regret, we can now analyze decision-making algorithms to assess their performances. The goal is to find an upper bound on the cumulative regret that holds in expectation and compare it with the lower bound on the expected cumulative regret for a given setting.[4] The goal is to assess if the upper bound of an algorithm matches the lower bound for the setting, at least in the more relevant quantities (e.g., the time horizon T, the number of arms k). For what concerns the most important quantity, i.e., the time horizon T, an algorithm is consistent over a class of bandits if, for every bandit $\boldsymbol{\nu}$ in the class, it holds that:

$$\lim_{T \to +\infty} \frac{\mathbb{E}_{\boldsymbol{\nu}}[R(\boldsymbol{\pi}, T)]}{T^p} = 0,$$

for some $p > 0$ (Lattimore and Szepesvári, 2020).

To assess the performance of an algorithm or the complexity of a setting, there are two kinds of bounds that we can take into account, namely *instance-dependent* and *minimax* (also called *instance-independent* or *worst-case*) bounds. In the instance-dependent bounds, we consider a specific instance $\boldsymbol{\nu}$ of the class of MABs under analysis, and we characterize the upper and lower bounds by presenting the results w.r.t. the properties of a specific instance, e.g., the suboptimality gaps Δ_i. In the minimax bounds, we search inside the class of MABs under analysis, the most challenging instance $\boldsymbol{\nu}$, and we evaluate the performance of such instance. These bounds classify the complexity of a given MAB class, so the result does not include the quantities of a specific instance.

2.4 Widely Adopted Algorithms

In this section, we present three representative types of MAB algorithms. First, we present `UCB1` (Auer et al., 2002a; Bubeck, 2010), the most known

[4] Regret bounds can also be defined in high probability: after having fixed $\delta \in (0, 1)$, we can find a bound holding with probability $1 - \delta$. However, this formulation is weaker if compared to results in expectation. For this reason, in this thesis, we consider results that hold in expectation.

algorithm to handle stochastic MAB, which considers a finite set of arms and σ^2-subgaussian or bounded rewards. Then, we present the `Lin-UCB` algorithm (Abbasi-Yadkori et al., 2011), the most known solution for handling linear bandits with continuous arm d-dimensional space and linear stochastic payoffs in the performed actions. Finally, we present `GP-UCB` (Srinivas et al., 2010), an algorithm widely used in practice to handle continuous action space in a non-parametric manner using kernel methods.

Bachmann-Landau Notation Before presenting the algorithms described above and briefly characterizing their performances, we need to introduce some notation to present the bounds in a simplified way. This notation is due to Bachmann and Landau and allows us to present the results in a simple manner, avoiding constants and other quantities that are not of interest. Given a regret upper bound $R(T)$, we can write that the bound $R(T)$ is $\mathcal{O}(f(T))$ if:

$$R(T) = \mathcal{O}(f(T)) \iff \limsup_{T\to\infty} \frac{R(T)}{f(T)} < \infty.$$

In the same way, given a regret lower bound $R(T)$, we can write that the bound $R(T)$ is $\Omega(f(T))$ if:

$$R(T) = \Omega(f(T)) \iff \liminf_{T\to\infty} \frac{R(T)}{f(T)} > 0.$$

The same results can be simplified by also omitting logarithmic dependences using the notations $\widetilde{\mathcal{O}}(\cdot)$ for the upper bounds.

2.4.1 `UCB1`

The first algorithm we present is `UCB1`, whose pseudocode is provided in Algorithm 2.1. The first finite-time analysis of its theoretical guarantees is due to Auer et al. (2002a). This algorithm selects at each time step $t \in [\![T]\!]$ an action in an optimistic way. The optimistic estimate is composed of the sum of two components: the empirical mean of the rewards retrieved by pulling an arm, and a bound (i.e., the Hoeffding bound) to the quantity of how much we are uncertain about such an estimate. The goal is to let us guarantee that the real value is below the optimistic bound with high probability. The version presented in Algorithm 2.1 is a more efficient Hoeffding confidence bound which does not require the knowledge of the optimization horizon T, and it is first presented and analyzed by Bubeck (2010).

Algorithm 2.1: `UCB1` (Auer et al., 2002a; Bubeck, 2010).

```
Input: number of arms $k$, exploration parameter $\alpha > 2$,
       subgaussianity parameter $\sigma$
Initialize $N_i \leftarrow 0,\ \widehat{\mu}_i \leftarrow 0, \quad \forall i \in [\![k]\!]$
for $t \in [\![T]\!]$ do
    Compute $\mathrm{UCB}_i \leftarrow \widehat{\mu}_i + \sigma\sqrt{\frac{\alpha \log t}{N_i}}, \quad \forall i \in [\![k]\!]$
    Select $I_t \in \arg\max_{i \in [\![k]\!]} \mathrm{UCB}_i$
    Play $I_t$ and observe reward $X_t$
    Update $\widehat{\mu}_{I_t} \leftarrow \frac{\widehat{\mu}_{I_t} N_{I_t} + X_t}{N_{I_t} + 1}$
        $N_{I_t} \leftarrow N_{I_t} + 1$
end
```

`UCB1` presents a regret an instance-dependent upper bound on expected cumulative regret in the order of $\mathcal{O}(\log T)$ (Bubeck, 2010), and matches the lower bound for this setting up to constants factors (Lai and Robbins, 1985). From the minimax perspective, `UCB1` presents a worst-case upper bound on expected cumulative regret in the order of $\widetilde{\mathcal{O}}(\sqrt{T})$, and matches the lower bound for this setting up to logarithmic factors.[5]

2.4.2 `Lin-UCB`

The second algorithm we present is `Lin-UCB` (Abbasi-Yadkori et al., 2011), whose pseudocode is presented in Algorithm 2.2. `Lin-UCB` is the most famous and widely adopted algorithm for Linear Bandits. The main difference from the algorithmic point of view is that the estimate of the empirical mean is substituted by a Ridge-regularized regression. The optimistic bound models how the values estimated through the regression concentrate to the real values. Also here, the goal is to define an optimistic confidence area in which the real values of $\boldsymbol{\theta}$ are contained in high probability. `Lin-UCB` is optimal and presents an expected cumulative regret upper bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ and is optimal up to logarithmic factors (Lattimore and Szepesvári, 2020).

[5] For what concerns the minimax optimality, `MOSS` (Audibert and Bubeck, 2009, 2010) is minimax optimal up to constant factors.

Algorithm 2.2: `Lin-UCB` (Abbasi-Yadkori et al., 2011).

```
Input: Regularization parameter $\lambda > 0$, exploration bounds $(\beta_{t-1})_{t \in [\![T]\!]}$
Initialize $t \leftarrow 1$, $\mathbf{V}_0 = \lambda \mathbf{I}_d$, $\mathbf{b}_0 = \mathbf{0}_d$, $\widehat{\boldsymbol{\theta}}_0 = \mathbf{0}_d$
for $t \in [\![T]\!]$ do
    Compute $\mathbf{a}_t \in \arg\max_{\mathbf{a} \in \mathcal{A}} \mathrm{UCB}_t(\mathbf{a}) := \langle \widehat{\boldsymbol{\theta}}_{t-1}, \mathbf{a} \rangle + \beta_{t-1} \|\mathbf{a}\|_{\mathbf{V}_{t-1}^{-1}}$
    Play action $\mathbf{a}_t$ and observe $X_t$
    Update $\mathbf{V}_t = \mathbf{V}_{t-1} + \mathbf{a}_t \mathbf{a}_t^{\mathsf{T}}$, $\mathbf{b}_t = \mathbf{b}_{t-1} + \mathbf{a}_t X_t$
    Compute $\widehat{\boldsymbol{\theta}}_t = \mathbf{V}_t^{-1} \mathbf{b}_t$
end
```

2.4.3 `GP-UCB`

The last algorithm we present is `GP-UCB` (Srinivas et al., 2010), whose pseudocode is reported in Algorithm 2.3. The main characteristic of this algorithm is that it is based on Gaussian Process (GP) Regression (Rasmussen and Williams, 2006). This employment of kernels due to the GP allows us to establish a concept of similarity between arms. The algorithm runs a non-parametric regression using standard Gaussian process techniques, and add, as in the previous cases, a confidence interval to take care of uncertainty. This bound is multiplied by the epistemic uncertainty directly retrieved by the Gaussian Process.

Algorithm 2.3: `GP-UCB` (Srinivas et al., 2010).

```
Input: Input space $\mathcal{A}$, GP prior $\hat{\mu}_0 = 0$ and $\sigma_0$
for $t \in [\![T]\!]$ do
    Compute $\mathbf{a}_t \in \arg\max_{\mathbf{a} \in \mathcal{A}} \mathrm{UCB}_t(\mathbf{a}) := \hat{\mu}_{t-1}(\mathbf{a}) + \sqrt{\beta_t} \hat{\sigma}_{t-1}(\mathbf{a})$
    Play action $\mathbf{a}_t$ and observe $X_t$
    Perform a Bayesian update to obtain $\hat{\mu}_t$ and $\hat{\sigma}_t$
end
```

`GP-UCB` with Radial Basis Function kernel presents a cumulative regret upper bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ that holds in high probability (Chowdhury and Gopalan, 2017). This result is optimal up to logarithmic factors (Cai and Scarlett, 2021).

Part I

Dynamic Pricing

CHAPTER 3

Introduction on Dynamic Pricing

Most international economic forecasts agree that nearly 50% of the annual value unlocked by the adoption of Artificial Intelligence (AI) from 2030 on will be in marketing and sales (Chui et al., 2018). Examples of activities in which AI tools can play a central role in marketing and sales include attracting and acquiring new customers, suggesting and recommending products, and optimizing customers' retention and loyalty. In particular, AI can effectively automate these processes so as to increase their efficiency dramatically. The most important choice that AI can help us make when we want to sell an item on an e-commerce website is the price at which sell it. In recent years, plenty of e-commerce are adopting the practice of dynamic pricing to define the optimal pricing of the products through AI methods.

With dynamic pricing, we refer to the practice of keeping the pricing schedule dynamic in time. In dynamic pricing, the optimal price is defined through several factors, that depend on the specific instance we are considering. The most important factor we have to consider when performing dynamic pricing is the conversion rate curve, i.e., the relation between the price we set and the probability of a user buying a given item at such a price.[1] Several other factors can be taken into account to find the best price.

[1] In several parts of this thesis, we talk about *conversion rate* and *demand* curves in a comparable way. We

Examples are the availability of a given item, the expected request in the future, the price of the competitors, the types of customers (if we are allowed to display different prices for different users).

The main advantage of dynamic pricing is the extreme flexibility that is enabled in managing the prices, deciding the objective (e.g., net worth, turnover) and the time horizon we want to optimize.

In the last years, most of the tools performing dynamic pricing are enabled by AI techniques that are mainly used to make both sales forecasting and demand estimation. The customary AI technique chosen to perform dynamic pricing tasks is multi-armed bandits, which, thanks to their ability to efficiently use samples and efficiently exploit acquired knowledge during the learning process, are the best-suited tool to perform this task (Den Boer, 2015).

3.1 Foundations of Dynamic Pricing

In this section, we first discuss the various scenarios and assumptions on the product, customers, and competition that are usually considered in dynamic pricing settings. Then, we present what practically implies the estimate of a demand curve, and the common problem in its estimate. Finally, we present a simple example of an objective function.

Standard Assumptions Most of the works in dynamic pricing make use of the same standard assumptions. The first assumption, which is valid for almost all the products usually sold by e-commerce, is about the demand, which is usually assumed to be decreasing in price. This assumption holds for all products different from *Veblen*, *Giffen* or *luxury* items. The other assumptions about the products are that they are not perishable and with virtually unlimited availability. The latter assumption is usually adopted (as it is virtually true) when we perform dynamic pricing for e-commerce websites adopting the *dropshipping* (Singh et al., 2018) paradigm. If it does not hold, the objective function will be radically different to take into account the availability, and the main challenges in dynamic pricing will be related to demand forecasting.

Demand Curve One of the most important challenge of dynamic pricing is the task of demand (or conversion rate) estimation. Suppose we want to estimate the conversion rate for the sake of simplicity. Such a conversion rate is a function $\mathcal{D} : \mathcal{P} \rightarrow [0, 1]$, where $\mathcal{P}$ is the set of possible prices. It

refer to the *conversion rate* curve as the normalized version of the *demand* curve. We often adopt the term *demand* due to its generality.

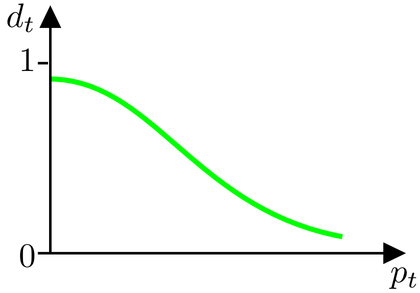


Figure 3.1: *Example of a conversion rate d_t w.r.t. price p_t at time t.*

represents the *willingness* (i.e., probability) of a customer buying an item at a given price. This task is fundamental in almost every dynamic pricing algorithms, no matter the assumptions we are considering (e.g., limited or unlimited availability) that can be integrated into the objective function (see below). Given a fixed cost c, we can estimate the conversion rate curve model at time t w.r.t. both margin m_t and prices p_t, and the two will be equally informative. Given a time-varying acquisition cost c_t, computing the conversion rate w.r.t. margin and price leads to different results. This requires to understand if the elasticity of the customers is on the margin or on the price. Assuming that the acquisition cost changes due to market dynamics for all the players (i.e., for our competitors and us), we can consider the margin. Otherwise, it is convenient to select the price. An example of conversion rate curve $d_t(p_t)$ w.r.t. price p_t at time t is provided in Figure 3.1. We can notice how the conversion rate is a probability, so it is bounded in $[0, 1]$. As stated above, the conversion rate curve's goal is to measure customers' willingness to buy an item. Such a conversion rate may vary significantly or be almost stable over the prices. In the first case, we talk about *elastic* customers (e.g., the ones of Figure 3.1), while in the second, we talk about *unelastic* customers (e.g., the ones of Figure 3.2).

Demand Estimation The estimate of the demand curve model can be performed in several ways, given the type of transactional data we have. Suppose to be able to observe (in addition to the purchases) when a customer visits a page and does not purchase the item. We have two ways to perform demand estimation in this scenario. The first way is to see every purchase or give up from the purchase at a given price as the realization of a *Bernoulli Random Variable* where the expected value is the demand at such a price. An example of that is provided Figure 3.3 (considering continuous price space $\mathcal{P}$). The second way, useful in the case in which we maintain the prices for time periods of comparable length, is to draw the curve using

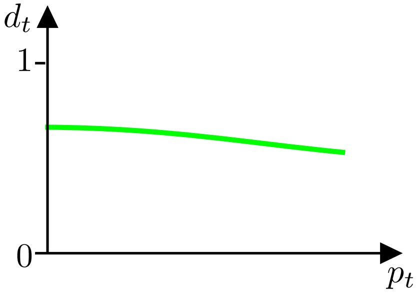


Figure 3.2: *Example of a unelastic conversion rate d_t w.r.t. price p_t at time t.*

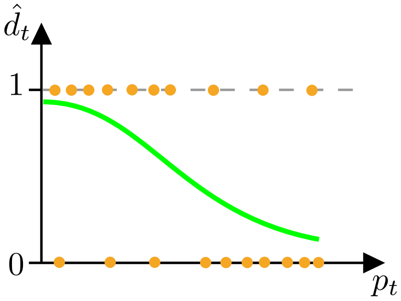


Figure 3.3: *Example of an estimate of the demand $\hat{d}_t$ w.r.t. price p_t at time t using data as Bernoulli random variables.*

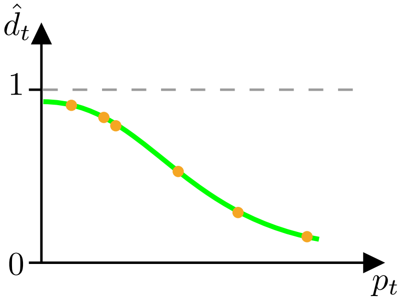


Figure 3.4: *Example of an estimate of the demand $\hat{d}_t$ w.r.t. price p_t at time t using data as normalized Binomial random variables.*

the number of purchases normalized by the number of purchases in addition to the number of give-ups. In this way, we get a buying probability, which can be seen as a (normalized) realization of a *Binomial Random Variable*. An example of this procedure is depicted in Figure 3.4 (even in this case, we consider a continuous price space $\mathcal{P}$). The main difficulty in demand estimation is due to the fact that, often, we have no access to the number of give ups, and so it is difficult to get the zero values in Figure 3.3 or define normalization factors in Figure 3.4.[2] A naive solution is, at least for the second case, to ignore the denominator, model the volumes, and then normalize these values. However, the sample in this way will be extremely noisy due to external factors such as seasonality, trend effects, temporary promo, and changes in competitors and markets.

Objective Function Objective functions may vary depending on the time horizon defined for our strategy by the business unit. However, given a time horizon T, our goal is usually to maximize some function of the volumes v_t and the selling price p_t, given the acquisition cost c_t. This correspond to find the set of prices maximizing $(p_1, \dots, p_T)$:

$$(p_1^*, \dots, p_T^*) \in \underset{(p_1,\dots,p_T) \in \mathcal{P}^T}{\arg\max} \sum_{t=1}^{T} f_t.$$

f_t represents our objective function, which, in the case of unlimited availability of the item under consideration, will be like:

$$f_t = (p_t - \alpha c_t)\ v_t(p_t),$$

with $\alpha \in [0, 1]$ a coefficient allowing to select as an objective generic convex combination of *net worth* (that can be obtained by selecting $\alpha = 1$) and *turnover* (that instead can be obtained by selecting $\alpha = 0$). One of the goals of a dynamic pricing algorithm is to replace v_t, which is unknown a priori, with an estimate of the demand $\widehat{d}_t$ (demand and volumes, once we remove seasonality, trends, and exogenous factors, are binded by a constant, which does not impact in the price we select). Our empirical objective function, when we replace the volumes with the demand becomes:

$$(p_1^*, \dots, p_T^*) \in \underset{(p_1,\dots,p_T) \in \mathcal{P}^T}{\arg\max} \sum_{t=1}^{T} \widehat{f}_t,$$

where:

$$\widehat{f}_t = (p_t - \alpha c_t)\ \widehat{d}_t(p_t).$$

A graphical representation of such a process for a given time instant t is provided in Figure 3.5. In such a figure, we can observe an intuitive tradeoff in the price we set. On the one hand, if the price is too low, the objective

[2] This is due to the fact that, for example, we cannot see the reaction of the customers which observes the prices outside our e-commerce, e.g., in a price comparison tool.

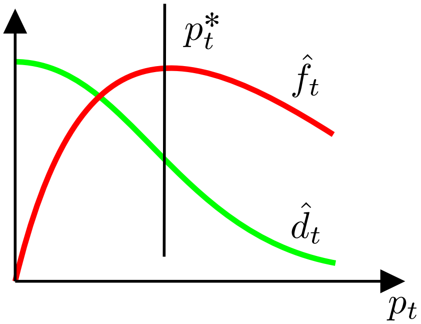


Figure 3.5: *Example of an estimated demand $\widehat{d}_t$ (green) and objective function $\widehat{f}_t$ (red) at time t.*

function will suffer a low per-item margin, even if we will sell a lot of items. On the other side, when the price is high, we will have a large per-item margin, but we will sell fewer items. So, we need to balance this tradeoff. In the case of limited availability of the products, the objective function may be revisited to take into account the availability constraints.

CHAPTER 4

Dynamic Pricing for the Long Tail

In this chapter, we propose an approach to manage pricing strategies in e-commerce websites. The goal of this algorithm, namely *DynaLT* (*Dynamic pricing for the Long Tail*), is to propose a unified approach for pricing, able to price all kinds of product, from the ones with very high volumes, requiring precise pricing strategies, to the ones with very low volumes, which instead requires to be properly managed to face the problem of data scarcity. In particular, we propose an online learning data-efficient algorithm able to define prices using a data aggregation strategy for the products with few sales in the past, in order to face the problem of data scarcity.

This chapter presents (Mussi et al., 2022a) a joint project with Gianmarco Genalti, Alessandro Nuara, Francesco Trovó, Nicola Gatti and Marcello Restelli, published at the *ACM Conference on Knowledge Discovery and Data Mining (KDD)*. A preliminary version of this work (Genalti et al., 2022) appeared at the *European Workshop on Reinforcement Learning (EWRL)*.

4.1 Introduction

The long-tail business model is pervasive in e-commerce. In particular, the long tail (Anderson, 2006) is a business strategy allowing companies to get a significant profit by selling low volumes of *hard-to-find* items to many customers instead of selling exclusively large volumes of a small set of *popular* items (Brynjolfsson et al., 2011). On the one hand, dealing effectively with the long tail is technically challenging as data per product are extremely scarce. Most importantly, such a data scarcity precludes the adoption of several Artificial Intelligence (AI) tools of great success, such as, e.g., deep learning, thus leaving the problem of designing suitable tools open. On the other hand, effective long-tail optimization is crucial for a company. Indeed, the revenue from the long tail usually represents a significant portion of the company's revenue. Furthermore, the competition with other companies on the long tail is weaker than that on the popular products due to the difficulties in optimizing the pricing. Therefore, an effective optimization of the long tail can lead to a significant increase in revenue.

Original Contribution In this chapter, we focus on real-world long-tail scenarios that are usually *non-stationary* due to the *seasonality* and/or competitors' *adaptive* behaviors. We design an *online learning algorithm* for dynamic pricing, which updates the estimates on the demand curve sample by sample and makes decisions to balance the customary machine learning trade-off between *exploitation* and *exploration*. We assume that the process to learn is stochastic. Such an assumption is reasonable even in the presence of adaptive competitors since, as we observed in our experimental analysis, the competitors rarely change the prices of their product. Technically speaking, we use historical data to capture seasonality, and we combine them with a sliding window to forget old data.

The main challenges due to the long tail we face are two. The first challenge concerns the design of learning algorithms that are robust and efficient when data are scarce. More precisely, when a small amount of data are available, the observation of a new sample can dramatically change the shape of the estimated demand curve. In this case, robustness is crucial to avoid significant variations of the algorithm outputs. Similarly, data efficiency is of paramount importance in non-stationary settings to effectively track the changes and limit the delay in the learning process. We force the *monotonicity* of the demand curve learned by the algorithm to address this challenge. Remarkably, this assumption commonly holds with long-tail products and allows better robustness (as new samples cannot dramatically change the shape of the demand curve learned by the algorithm) and data efficiency (as a sample at a given price provides information to many other prices). We force monotonicity by resorting to a specific class of Bernstein polynomials when estimating the demand curve. The second challenge concerns the design of algorithms capable of clustering the products such that every product of the same cluster will be priced with the same policy. In this case, there are two critical issues. The former is that the clustering cannot be based only on transaction data as data are too scarce. The second is that the common approaches assign some long-tail products to a popular product, which may be inefficient in practice due to the different market dynamics. The peculiarity of our clustering algorithm resides in exploiting similarities among products discovered from textual data describing the products, and it provides an explainable clustering by decision-tree approaches.

We first evaluate our algorithms in an offline synthetic setting, comparing their performance with the state-of-the-art and showing that our algorithms are more robust and data-efficient in the long-tail settings, thus supporting the need to adopt monotonicity in practice. Subsequently, we evaluate our algorithms in a real-world online setting with more than 8,000 products, including popular and long-tail, in an A/B test with humans for about two months. In this experiment, we obtain a revenue increase of about 18% for the popular products and 90% for the long-tail products.

4.2 Application Domain and Motivation

4.2.1 Industrial Context

Business Scenario Our work has been conducted in collaboration with an Italian e-commerce website selling more than $20,000$ different products (in the specific case, we consider non-perishables consumables). Notice that in this case the assumption of monotonicity of the demand curve trivially holds as these products are not luxury, Veblen, or Giffen. The e-commerce website adopts the drop-shipping business model. Thus, it is not subject to warehousing costs and can suggest/recommend many different products to the customers, including long-tail products leading to rare yearly transactions. In particular, 75% of the products provide about 590 KEuros corresponding to about 10% of the total e-commerce turnover, and the number of units sold for these products in 2021 is smaller than 10. Furthermore, about 50% of the products available in the catalog present no order in 2021. By a simple analysis of the transactions carried out in 2021, it can be ob-

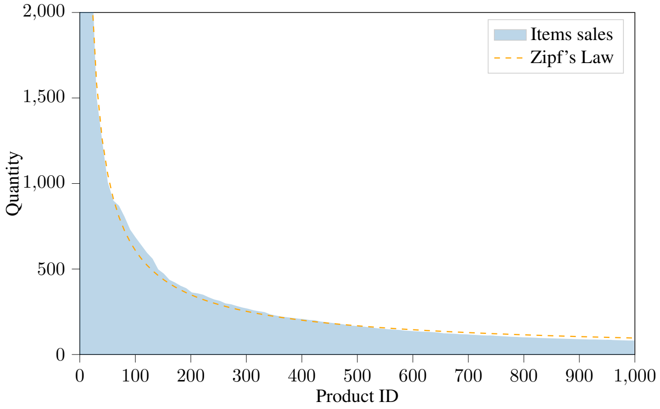


Figure 4.1: *Units sold per product in* 2021 *by the e-commerce for the top* $1,000$ *products, compared with (long-tail) Zipf's Law* $z(x) = \frac{c}{x^{0.8}}$.

served that the products, once re-ordered according to the number of units sold, satisfy the well-known (long-tail) *Zipf's Law* (Zipf, 1949), as shown in Figure 4.1.

To simplify the business processes, the e-commerce website management required the design of a single algorithm to perform pricing on long-tail and popular products coherently. The adoption of a single algorithm for both kinds of products is due to the simplicity in its management and to maintain fairness w.r.t. customers. Moreover, adopting different pricing policies for different products could be perceived as unfair by customers. The objective function to maximize is the *total profit*.

Market Landscape The e-commerce website with which we collaborate works in a market presenting a significant seasonality. We study it as follows. For every year, we count the number of sales per week and then we normalize them by the total number of the annual sales. In this way, we obtain the percentage of the annual sales distributed over 52 weeks. Figure 4.2 shows such a distribution once averaged over 5 years. The same analysis has been conducted to understand the buyers' trend over the different days of the week. The result is presented in Figure 4.3, where we can observe a strong intra-week periodic behavior. The size of the market, not reported in the figures, changes year by year, dramatically shrinking in 2020 due to the COVID-19 pandemic outbreaks. Notice that such a non-stationary behavior of the environment is also due to the presence of competitors whose share is significant w.r.t. the total market. Customarily, to monitor and compensate such effects, companies exploit data-scraping services to monitor competitors' pricing. Notice that this approach is not economically sustainable in

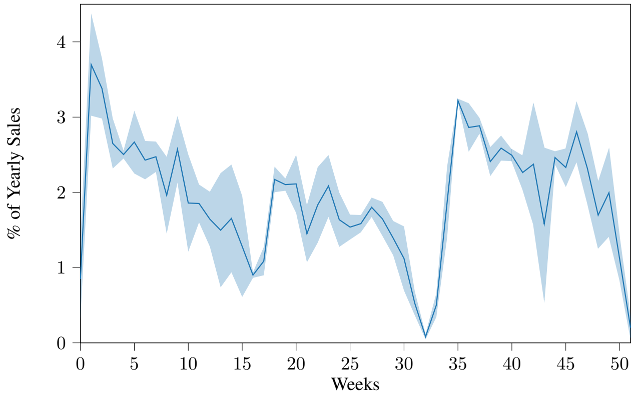


Figure 4.2: *Percentage of sales over the* 52 *weeks in a year (standard deviation is depicted as semitransparent areas).*

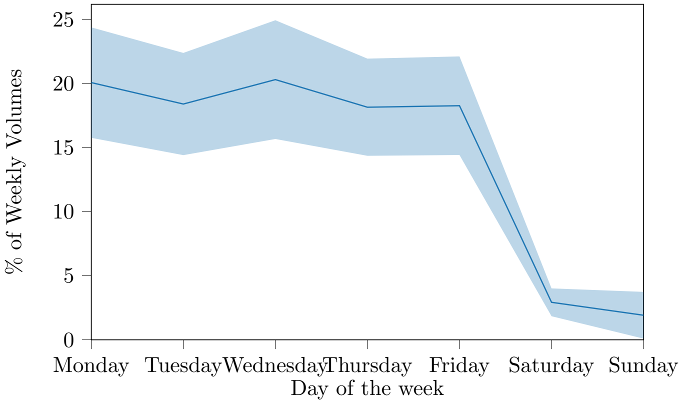


Figure 4.3: *Seasonality over a single week (standard deviation is depicted as semitransparent areas).*

the case of long-tail products due to the large number of different products to monitor which in its turn would require significant expenses in terms of scraping. However, a preliminary set of experiments on a few popular products shows that the competitors do not behave adversarially, i.e., no prompt reaction to price changes by the analyzed e-commerce website triggers a prompt response of the competitors. To assert this fact, we performed a Spearman's rank correlation test (see Kokoska and Zwillinger, 2000) to assess if the competitors react to our volumes' changes, i.e., if the variation of the prices we applied is correlated to theirs. The test is conducted over 35 best seller product, for which we have reliable data. Formally, the null hypothesis of the test is "The two variables are uncorrelated". The tests set with a significance level of 0.05 do not provide any strong statistical evidence that the two prices variation are correlated. In practice, the competitors change prices with a low frequency and disregard the specific changes performed by our algorithms. Therefore, in what follows, we model the competitors by including them as one of the effects present in the non-stationary stochastic environment.

4.3 Relevant Literature of Dynamic Pricing

In this section, provide an overview of the relevant works in dynamic pricing. We divide this section in two parts. The first provides an overview of *learning for dynamic pricing* (Section 4.3.1), while the second focuses on *long tail* (Section 4.3.2). For a comprehensive analysis of the dynamic-pricing literature, refer to (Narahari et al., 2005; Bertsimas and Perakis, 2006; Den Boer, 2015).

4.3.1 Learning for Dynamic Pricing

Rothschild (1974) presents one of the seminal works on the adoption of MAB algorithms for dynamic pricing. This algorithm has been subsequently extended in several directions to capture the characteristics of different pricing settings. Kleinberg and Leighton (2003) study the problem of dealing with continuous-demand functions and proposes a discretization of the price values to provide theoretical guarantees on the regret of the algorithm. This approach suffers from the drawback that the reward is assumed to have a unique maximum in the price. Such an assumption is hard to be verified in practice. Trovò et al. (2015, 2018) relax this assumption, assuming that the demand function is monotonically decreasing and exploiting this assumption in the learning algorithm to provide uncertainty bounds tighter than those of classical frequentist MAB algorithms. However, the model formulation explicitly imposes neither monotonicity nor weak monotonicity on the estimated demand functions, so decisions that violate business logic can be allowed during the learning process. The authors show how the monotonicity assumption does not improve the asymptotic bound of regret provided by the MAB theory. On the other hand, exploiting monotonicity allows for an empirical improvement in performance (see Section 4.7 and Mussi et al. 2023b). The same argument also holds for the work proposed by Misra et al. (2019), where the monotonicity property of the demand function is used to ensure faster convergence. However, monotonicity is not forced as a model-specific feature. Besbes and Zeevi

(2015) show that linear models are a suitable and efficient tool for modeling a demand function. Other works that adopt a parametric formulation of the demand function are by Besbes and Zeevi (2009) and Broder and Rusmevichientong (2012). These works assume stationary customer behavior. Cope (2007) and Bauer and Jannach (2018) are two of the main works on Bayesian inference applied to dynamic pricing. They both do not impose monotonic constraints on the model. Interestingly, Bauer and Jannach (2018) take into account non-stationary features (e.g., competitors' prices). Araman and Caldentey (2009) use a Bayesian approach to dynamic pricing using a prior belief on the parameters to capture market-related information and force the model to be monotonic. Wang et al. (2021a) investigates non-parametric models for demand function estimation. In this case, the authors assume that the demand function is smooth. Finally, Nambiar et al. (2019) propose a model to deal with both the non-stationarity data and the model misspecification. However, the required contextual knowledge at a product-wise level is not usually available in practice.

4.3.2 Long-Tail Pricing

A few recent works have been proposed in the dynamic pricing literature to deal with the long tail. In particular, Gandhi et al. (2020) and Adam et al. (2020) provide parametric models for the demand curve estimation in a setting where many products may present no transactions in the historical data. However, these works merely estimate the demand curve without addressing the exploration/exploitation dilemma, and thus no guarantees can be provided. Miao et al. (2019) propose a pricing algorithm where an online clustering is performed to deal with long-tail products. The authors perform dynamic pricing in a context-based fashion where clustering is based only on transaction data. This requires observing at least one transaction per product, which is rarely met in real-world long-tail settings. In Ye et al. (2018), products are clustered through contextual information. However, due to the unique nature of the involved products, this approach cannot be adopted in several scenarios as this information are often not provided.

4.4 Problem Formulation

We study the scenario in which an e-commerce website sells a set $\mathcal{J}$ of non-perishable products with unlimited availability. We assume that a textual description and transaction data are available for all the products $j \in \mathcal{J}$ sold in the past.

At every time $t \in [\![T]\!]$, we aim to set a percentage margin (from now on, the margin) $\overline{m}_{jt} \in \mathcal{M}_j$, where $\mathcal{M}_j$ is the finite set of feasible values for the margin of an item $j \in \mathcal{J}$. Such a margin $\overline{m}_{jt}$ is defined as:

$$\overline{m}_{jt} := \frac{p_{jt} - c_j}{c_j}, \tag{4.1}$$

where p_{jt} and c_j are the selling price and the acquisition cost for product j at time t, respectively. Finally, we denote with $v_{jt}(\overline{m}_{jt})$ the actual number of sales (volumes) for an item j at time t when choosing margin $\overline{m}_{jt}$.

Learning Problem The objective function the e-commerce website aims to maximize is the *total profit*. Formally, the maximization problem is:[1]

$$m^*_{jt} = \arg\max_{\overline{m}_{jt} \in \mathcal{M}_j} \; f_{jt}(\overline{m}_{jt}), \tag{4.2}$$

where:

$$f_{jt}(\overline{m}_{jt}) := \overline{m}_{jt} \, c_j \, v_{jt}(\overline{m}_{jt}). \tag{4.3}$$

Given a policy $\boldsymbol{\pi}$ returning at day t a margin value $\overline{m}_{jt}$ for each product $j \in \mathcal{J}$, we define the pseudo-regret over time $t \in [\![T]\!]$ as:

$$R(\boldsymbol{\pi}, T) := \sum_{t \in [\![T]\!]} f_{jt}(m^*_{jt}) - \sum_{t \in [\![T]\!]} f_{jt}(\overline{m}_{jt}),$$

where $f_{jt}(m^*_{jt})$ is the expected value provided by a clairvoyant algorithm choosing the optimal margin for each product. Intuitively, the notion of regret provides a measure of the cumulative loss of our policy $\boldsymbol{\pi}$ w.r.t. the (clairvoyant) policy choosing at each time t the optimal margin maximizing $f(\cdot)$. Thus, our goal is the minimization of the pseudo-regret $R(\boldsymbol{\pi}, T)$, which is equivalent to the task of maximizing of the cash flow margin accumulated over time.

4.5 Pricing Single Products

To model the demand curve for each product j, we use the transaction data aggregated over a time interval of one week. These data consist in the aggregated average margin $\overline{m}_{j\tau}$ and amount of units sold $v_{j\tau}$, for every product j and each week τ.

Seasonality Motivated by the seasonality analysis depicted in Figure 4.2, we factorize the dependence of the volumes on seasonality and margin with

[1] Let us remark that this problem also applies to a generic convex combination of turnover and total profit.

two different, independent functions. In particular, we define the *adjusted volumes* $\bar{v}_{j\tau}$ of product j at time τ as follows:

$$\bar{v}_{j\tau} := v_{j\tau}\ s_{j\tau},$$

where $s_{j\tau}$ is a coefficient, independent of the chosen margin, representing the seasonality for product j at time τ. This coefficient is estimated from historical data as discussed in Appendix A.1. The above factorization allows a dramatic reduction of the samples needed to have a stable estimate of the demand curve.

Non-stationary demand In addition to seasonality effects, the market can be non-stationary due to trends (e.g., contractions or expansions) and adaptive behaviors of the competitors. These effects change the dependency of volumes on margin over time. We deal with these kinds of non-stationarity sources by adopting a sliding window that discards outdated data for the estimation of the volumes. More precisely, we use data coming from the last Y years for the estimation of the seasonality coefficients $s_{j\tau}$, while we adopt a sliding window of N weeks for the estimation of the adjusted volumes. Notice that, when N is small, the trend effect can be considered to be negligible, and the demand curve is sufficiently stable. In particular, the sliding window size is chosen to find the best trade-off to balance issues due to the non-stationary environment and trend w.r.t. the model's sample request to face noise and outliers (see Section 4.7.1).

4.5.1 Bayesian Estimation of the Demand Curve

We aim to find the best margin for a product j using its transaction data. Our estimation algorithm is based on a *Bayesian Linear Regression* (BLR, Tipping, 2001). In such a regression model, we build an estimate $\hat{d}_j(\cdot)$ of the demand function for product j as a linear combination of the basis function taken as input, formally:

$$\hat{d}_j(m) = \sum_{h=0}^{Z} \theta_h\ \phi_h(m), \tag{4.4}$$

where θ_h represents the h-th weight distribution and $\phi_h(m)$ represents the h-th basis function of the margin $m \in \mathcal{M}_j$. Notice that the BLR method returns a distribution $\hat{d}_j(m)$ for each margin $m \in \mathcal{M}_j$, which allows the adoption of a MAB approach in the following step of the algorithm. To increase data efficiency and robustness of the learning process in long-tail scenarios, we force our regression to return a monotonic non-increasing demand curve in the margin. Such an assumption is reasonable in our setting, as the products are non-perishable consumables and, therefore, they are not luxury, Veblen, or Giffen (Dougan, 1982; Kemp, 1998).

The demand curve estimation is performed using data collected during $\tau \in \mathcal{T} := \{t-N, \dots, t-1\}$, i.e., pairs $(\overline{m}_{j\tau}, \bar{v}_{j\tau})$ of input margins $\overline{m}_{j\tau}$ and output seasonality-adjusted volumes $\bar{v}_{j\tau}$. To force the monotonicity of the estimated demand curve $\hat{d}(\cdot)$, we use a specific transformation of the standard *Bernstein polynomials* (Bernstein, 1912; Lorentz, 1953a) as basis functions in combination with a non-negative prior distribution for the θ_h parameters. Formally, the Bernstein polynomials of degree Z are composed by $Z+1$ functions, defined as:

$$b_{h,Z}(m) = \binom{Z}{h} m^h (1-m)^{Z-h}, \quad h = \{0, \dots, Z\}, \tag{4.5}$$

where $\binom{Z}{h}$ is the binomial coefficient. Notice that the choice of Bernstein polynomials allows us to model any demand function satisfying mild assumptions. More precisely, Bernstein polynomials converge to any function satisfying boundedness and continuity in a given range for a sufficiently large degree Z of the polynomials (Lorentz, 1953b). An example of Bernstein polynomials with $Z = 20$ is shown in Figure 4.4a. Defining the row vector $\boldsymbol{b}_Z(x)$ as:

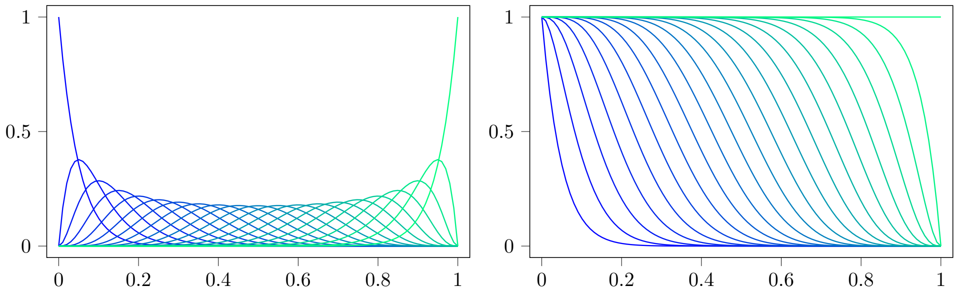
$$\boldsymbol{b}_Z(m) := \left[b_{0,Z}(m),\ b_{1,Z}(m),\ \dots,\ b_{Z,Z}(m)\right], \tag{4.6}$$

a monotonic version $\phi_h(m)$ of the original basis functions is obtained as follows (McKay and Ghosh, 2011; Wilson et al., 2020):

$$\phi_h(m) := \boldsymbol{b}_Z(m) \cdot \left(\mathbf{I}_{Z+1} - \boldsymbol{S}_{Z+1}\right)^{-1} \cdot \mathbb{1}_h, \quad h = \{0, \dots, Z\}, \tag{4.7}$$

where $\mathbf{I}_{Z+1}$ is the identity matrix of order $Z+1$, $\boldsymbol{S}_{Z+1}$ is the square matrix of dimension $Z+1$ with all 1 in the *superdiagonal* ($s_{i,i+1} = 1$ for each i, 0 otherwise, Olver and Shakiban, 2019), and $\mathbb{1}_h$ is the indicator operator selecting h-th element of the vector. An example of $\phi_h(m)$ obtained transforming the Bernstein's basis functions with $Z = 20$ is presented in Figure 4.4b. Since Bernstein polynomials are defined over the support $[0, 1]$, we rescale values over this range. From now on, for the sake of presentation and w.l.o.g., we assume that the margins are s.t. $\mathcal{M}_j \subseteq [0, 1]$.

To guarantee that the resulting demand function $\hat{d}_j(\cdot)$ is monotone, we use both the basis $\phi_h(\cdot)$ as defined in Equation (4.7) and a prior distribution for the θ_h parameters having a non-negative support. Therefore, we use the



(a) *Bernstein Polynomials.*

(b) *Basis Function using transformed Bernstein Polynomials.*

Figure 4.4: *Selected Basis Functions.*

Lognormal distribution (Wilson et al., 2020) to model the parameters θ_h. Formally, we have the following:

$$\theta_h \sim \mathcal{LN}(\mu_h, \sigma_h), \quad \forall h \in \{0, \dots, Z\},$$

where $\mathcal{LN}(\mu_h, \sigma_h)$ denotes the *Lognormal* distribution with mean μ_h and standard deviation σ_h. Finally, the model fitting is performed using the last N available samples, i.e., relying on the data $\{(\overline{m}_{j\tau}, \bar{v}_{j\tau})\}_{\tau \in \mathcal{T}}$.

4.5.2 Exploration Strategy

Our problem can be naturally formulated as an online learning problem, where the goal is to balance the acquisition of information on the stochastic functions properly while, at the same time, maximizing the cumulative reward. The procedure addressing the exploration/exploitation at best is summarized in Figure 4.5. In particular, we resort to a sampling procedure similar to *Thompson Sampling* (TS, Agrawal and Goyal, 2012; Kaufmann et al., 2012; Chowdhury and Gopalan, 2017). By construction, a Bayesian model provides in output a probability distribution of the posteriors on the weights, which can be used to drive the exploration in the learning process. Formally, we sample from the posterior distribution of BLR weights, retrieving a single realization of the posterior binding margins with the demand curve ($\hat{d}(\overline{m}_{jt})$).

According to the MAB framework, we choose the best arm over a finite set of possible margins (representing the arms) $\mathcal{M}_j$. We can compute the value of the expected objective function $\hat{f}(\overline{m}_{jt}), \forall \overline{m}_{jt} \in \mathcal{M}_j$, which is the counterpart of Equation 4.3 computed with the estimated demand function

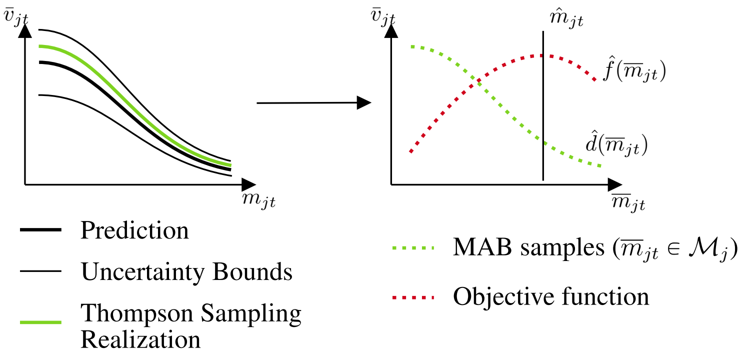


Figure 4.5: *Optimal margin $\hat{m}_{jt}$ estimation process.*

$\hat{d}(\cdot)$:[2]

$$\hat{f}(\overline{m}_{jt}) = \overline{m}_{jt}\ \hat{d}(\overline{m}_{jt}). \tag{4.8}$$

The optimal margin $\hat{m}_{jt}$ is the best arm, corresponding to:

$$\hat{m}_{jt} = \underset{\overline{m}_{jt} \in \mathcal{M}_j}{\arg\max}\ \hat{f}(\overline{m}_{jt}), \tag{4.9}$$

where $\hat{f}(\overline{m}_{jt})$ is the objective function estimated using demand curve $\hat{d}(\cdot)$, the latter coming from TS sample over the model.

4.6 Pricing Long-Tail Products

The algorithms proposed in Section 4.5 cannot be directly applied to long-tail products since the available data are not sufficient to produce a reliable estimate of the demand curve. The commonly adopted approach to applying to a long-tail product the same margin used for a popular product presenting similar characteristics may lead to wrong business decisions. This is mainly because the competition over long-tail and popular products is different, which, in its turn, can lead to different optimal margins.

We deal with long-tail products by aggregating *similar* products subject to the constraint that the aggregated data are sufficient to produce an accurate estimation of the corresponding demand curve. Then, we apply our bandit pricing algorithm to each aggregation of products singularly. In the following sections, we describe the steps of our algorithm.

[2] Recall that the demand curve is no longer the expected volumes curve due to aggregation (see Section 4.6.4) and seasonality adjustment process.

4.6.1 Distance Estimation

In this step, we exploit textual information to estimate the similarities among the products. This kind of information is indeed the only information available in large-scale e-commerce websites regarding the products. Initially, our algorithm removes the stop-words from the textual description (i.e., recurring words such as, e.g., "the" and "that"), as done in the work by Wilbur and Sirotkin (1992). Subsequently, the algorithm encodes into vectors the products' textual descriptions using *Term Frequency - Inverse Document Frequency* (TF-IDF, Luhn, 1957; Jones, 1972). After that, it computes a distance matrix $\mathcal{D} = [d_{jk}]_{j,k \in \mathcal{J}}$, in which every entry provides the distance d_{jk} between each pair of vectors obtained using TF-IDF. Such a matrix expresses the similarities among the products. Additional technical details are provided in Appendix A.2.

4.6.2 Tree Structure Generation

In this step, we generate a binary tree structure based on the products' similarities by applying the *hierarchical clustering* approach proposed by (Murtagh, 1983) to the products and the corresponding distance matrix $\mathcal{D}$.[3] In this tree structure, every terminal node (i.e., leaf) corresponds to a product $j \in \mathcal{J}$, and each non-terminal node corresponds to an aggregation of products, which we call *meta-products*. More precisely, a meta-product is the aggregation of those products whose leaves are reachable in the subtree whose root is the meta-product. Formally, we define a meta-product $\mathcal{K}$ as the set of products j present in the corresponding subtree. Figure 4.6 depicts an example of the tree structure resulting from the application of the above clustering approach over 6 products, in which products (terminal nodes) are depicted as squares, and meta-products (non-terminal nodes) are depicted as circles. In this example, the meta-product $\alpha = \{1, 2, 3, 4\}$ is the aggregation of the products 1, 2, 3, and 4. We remark that such a tree structure provides an explainable way to describe the similarities among the products whose interpretation is crucial in real-world applications, as it directly shows which products and aggregations are similar. Notice that, while all the non-terminal nodes are in principle meta-products, our algorithm works with only a subset of them chosen as discussed in the next step and discards the remaining ones.

[3] We remark that the application of the hierarchical clustering algorithm by (Murtagh, 1983) requires the choice of a distance metric and a linkage method, e.g., a method to compute the distance of two clusters. We adopt the metric induced by $\mathcal{D}$, and we opt for the use of the single linkage, as suggested by Ding and He (2002).

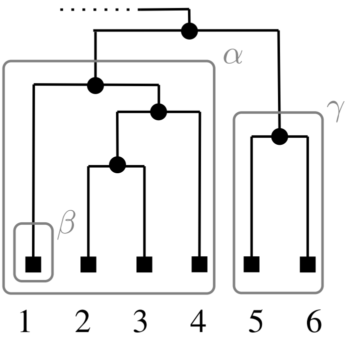


Figure 4.6: *Example of a tree structure.*

4.6.3 Product Aggregation Strategy

In this step, the algorithm chooses the specific subset of meta-products to be priced. The rationale is to return a set of *minimal* meta-products, each populated with a sufficient amount of data to obtain an accurate demand curve estimation.

For every product j, we define a vector $\mathrm{s}_j := (s_{jt-N}, \ldots, s_{jt-1})$, whose elements $s_{j\tau} = 1$ if at least a unit of the product j has been sold at time τ, $s_{j\tau} = 0$ otherwise. Similarly, given a meta-product α, we define a vector $\mathrm{s}_\alpha := (s_{\alpha t-N}, \ldots, s_{\alpha t-1})$, obtained as $\mathrm{s}_\alpha := \oplus_{j \in \alpha} \mathrm{s}_j$, where $\oplus$ is the bit-wise "or" operation of the vectors corresponding to the products j belonging to α. Notice that $s_{\alpha\tau} = 1$ if at time τ at least a unit of at least one product belonging to the meta-product α has been sold. The condition stating that the amount of data for a meta-product α are sufficient is that the number of time points for which there is at least a sale of meta-product α is at least $q\,N$, where $q \in (0, 1]$ is a parameter that we can tune. The above condition can be evaluated by computing the sum of the elements of s_α and comparing it with $q\,N$.

Finally, the choice of the meta-products is performed as follows. Starting from each product j we check the above condition, and if it is satisfied, the product j is chosen as a meta-product. An example of this case is represented by the product 1 in Figure 4.6. If the condition does not hold on the single product, we traverse upward the nodes of the aforementioned tree structure, and stop as soon as the above condition is satisfied. In this case, the meta-product corresponding to the non-terminal node is selected to build the demand model for the product j. Notice that the minimality principle we adopt is motivated by the need for balancing between the bias and variance of the demand curve estimates. Indeed, merging additional products to a minimal meta-product would most likely increase the bias of the demand curve estimated for each product therein, while providing only a negligible benefit in terms of variance reduction.

4.6.4 Meta-product Demand Estimation and Pricing

In this step, the algorithm estimates the demand curve of each selected meta-product and prices the corresponding products. Let us consider a meta-product aggregating the set of products $\mathcal{K} \subseteq \mathcal{J}$ and the corresponding sale statistics pairs $(\overline{m}_{k\tau}, \bar{v}_{k\tau})$ for all products $k \in \mathcal{K}$ and time $\tau \in \mathcal{T}$. We compute the demand curve of a meta-product using the overall volume $\bar{v}_{\mathcal{K}\tau}$ for a specific time τ and the corresponding average margin $\overline{m}_{\mathcal{K}\tau}$ used to get such a volume at time τ. The above quantities are computed, for each $\tau \in \mathcal{T}$, as:

$$\begin{aligned} \bar{v}_{\mathcal{K}\tau} &:= s_{\mathcal{K}\tau} \sum_{k \in \mathcal{K}} v_{k\tau}, \\ \overline{m}_{\mathcal{K}\tau} &:= \sum_{k \in \mathcal{K}} \overline{m}_{k\tau} \cdot \frac{v_{k\tau}}{\sum_{h \in \mathcal{K}} v_{h\tau}}, \end{aligned}$$

where the average margin is computed by averaging the products' margins weighted by their seasonality-adjusted volumes, and the coefficient $s_{\mathcal{K}\tau}$ is computed similarly to its single-product counterpart (details are provided in Appendix A.1). The estimated demand function and the final selected margin $\hat{m}_{\mathcal{K}t}$ for the meta-product are computed using the same procedure described in Section 4.5, i.e., using margins $\overline{m}_{\mathcal{K}\tau}$ as input of the regression model and volumes $\bar{v}_{\mathcal{K}\tau}$ as output, as well as the selection of the margin. Indeed, once the above conversion has been applied, the meta-product data are of the same nature as the one of a single product j and, therefore, are processed in the same way. Finally, the selection of the margin for a product j is provided by the margin of the meta-product $\mathcal{K}$ including j with the smallest cardinality. For instance, in Figure 4.6, the margin of product 2 is selected using meta-product α, while product 1 using the margin corresponding to meta-product β.

A visual representation of the overall algorithm described in the previous sections is provided in Figure 4.7. The process starts from the textual description of each product and, thanks to these information, builds the tree structure. Subsequently, using the transaction data available, it builds the meta-products, estimates the corresponding demand functions, and, finally, it provides a margin to apply to each product in the catalog.

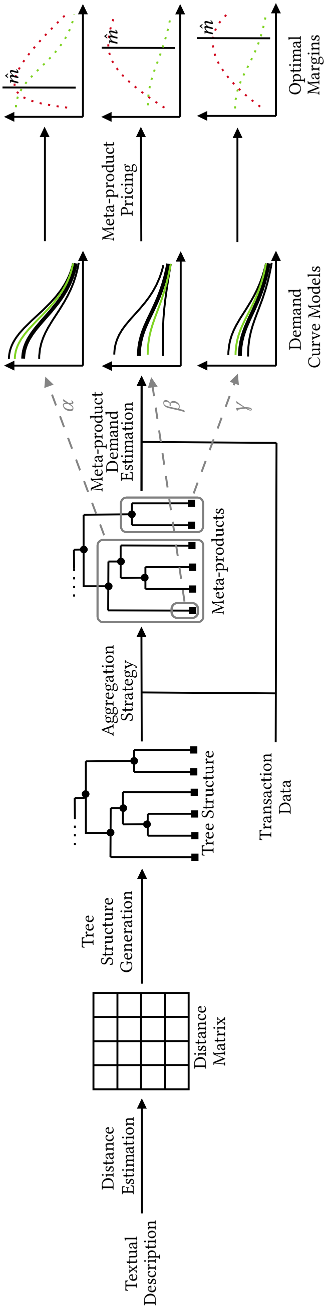


Figure 4.7: *Overview of the algorithm.*

4.7 Experimental Evaluation

In this section, we evaluate the empirical performance of our algorithms. Before doing that, we describe how we implement the actual choices described so far. Then, we show how the resort to monotonic bandits improves the pricing performance. To do that, we use an offline setting whose optimal solution is known. Subsequently, we describe the application of our algorithm to a real-world long-tail setting.

4.7.1 Pricing Single Products

We compare our algorithm with a BLR approach not exploiting the monotonicity, denoted with *NM-BLR*, where we use the Normal prior for the parameters θ_h and Bernstein's polynomials in Equation (4.7) as basis functions. A detailed description of the setting is deferred to Appendix A.3. We compare the two algorithms in terms of empirical regret $\hat{R}(\boldsymbol{\pi}, T)$, i.e., the empirical counterpart of the regret $R(\boldsymbol{\pi}, T)$. Results are averaged over 15 independent runs for each algorithm (standard deviation is reported in brackets).

Noise and Outliers

First, we study how our solution and the *NM-BLR* method are affected by the variation of the standard deviation of the noise of the volumes $v_{j\tau}$ and the introduction of outliers, i.e., the presence of customers performing significantly larger orders than usual. The time-stationary volume function we use is:

$$v(x) = 2e^{-(x+1.2)^{\frac{5}{2}}} + \epsilon,$$

where prices $x \in [0.32, 1.00]$ and $\epsilon \sim \mathcal{N}(0, \sigma)$ is a Gaussian zero-mean noise with standard deviation σ. The product had a unitary cost $c = 0.3$. In what follows, outliers are modeled as using, with probability o, a different noise term $\epsilon' \sim \mathcal{N}(0, \sigma')$ having $\sigma' = 10\,\sigma$. In particular, we investigate scenarios with $\sigma \in \{0.001, 0.005, 0.01\}$ and $o \in \{0\%, 10\%, 20\%\}$. The algorithms have been run over a time horizon of $T = 100$ weeks.

Results The empirical regret $\hat{R}(\boldsymbol{\pi}, T)$ obtained with the two methods are summarized in Table 4.1 (the smaller, the better). On average, *DynaLT* outperforms its non-monotone counterpart *NM-BLR* on every setting. Overall, as expected, the performance of the two algorithms degrades as the standard deviation of the noise σ and the outlier percentage o increase. Without outliers, *DynaLT* is significantly better than *NM-BLR* for each value of the

Table 4.1: *$\hat{R}(\boldsymbol{\pi}, T)$ in the presence of noise and outliers (15 runs, standard deviation in brackets).*

			Outlier percentage o		
			0%	10%	20%
Noise std σ	0.001	*DynaLT*	6.05 (0.12)	7.92 (0.2)	8.91 (0.25)
		NM-BLR	7.51 (0.04)	10.43 (0.08)	11.61 (0.14)
	0.005	*DynaLT*	9.36 (0.29)	18.17 (0.81)	22.09 (1.09)
		NM-BLR	16.34 (0.42)	21.88 (0.55)	25.15 (0.68)
	0.01	*DynaLT*	16.0 (0.27)	35.51 (1.74)	36.75 (1.56)
		NM-BLR	27.34 (0.6)	37.71 (1.64)	37.12 (1.21)

noise, and the improvement increases as the noise gets larger, with an improvement in terms of regret from $\approx 20\%$ for $\sigma = 0.001$ to $\approx 41\%$ for $\sigma = 0.01$. Conversely, with outliers, the advantage of using *DynaLT* is significant only for small values of noise standard deviation, e.g., $\sigma = 0.001$ and $\sigma = 0.05$, leading to a reduction of the regret in the range $[12\%, 23\%]$. Finally, with both a large noise standard deviation ($\sigma = 0.01$) and outliers ($o = 10\%$ and $o = 20\%$), the performance of the two techniques are comparable.

Non-stationarities

Second, we evaluate our algorithm in a non-stationary setting. To do that, we simulate some changes in the environment due to, e.g., a new competitor or a new product. This is done by abruptly changing the product volume function at specific time points. The specific shapes of the different volume functions are provided in Appendix A.3. In particular, we introduce $c \in \{1, 2, 3\}$ abrupt changes occurring at evenly spaced time points (over the entire time horizon). Notice that, since the underlying demand functions are different for different values of c, the regrets corresponding to these scenarios cannot be directly compared. Even in this case, we evaluate the impact of exploiting the monotonicity w.r.t. a traditional demand function estimation method (*NM-BLR*). Furthermore, we analyze the impact of changing the sliding window length $N \in \{20, 30, 40\}$. In this experiment, the empirical regret $\hat{R}(\boldsymbol{\pi}, T)$ is computed w.r.t. a clairvoyant policy that knows when the changes in the environment would occur, and its value has been averaged over 15 independent runs. The algorithms have been run over a time horizon of $T = 120$ weeks.

Results The empirical regrets are reported in Table 4.2 (the smaller, the

Table 4.2: *$\hat{R}(\pi, T)$ in the presence of non-stationarities (15 runs, standard deviation in brackets).*

			Number of Changes c		
			1	2	3
Window Size N	20	*DynaLT*	$4.16\ (0.54)$	$8.86\ (1.85)$	$6.51\ (0.56)$
		NM-BLR	$4.82\ (0.8)$	$12.8\ (2.17)$	$10.27\ (0.23)$
	30	*DynaLT*	$4.55\ (0.8)$	$9.49\ (2.15)$	$6.82\ (0.49)$
		NM-BLR	$4.84\ (0.84)$	$12.45\ (1.75)$	$12.75\ (0.27)$
	40	*DynaLT*	$3.95\ (0.54)$	$7.46\ (1.54)$	$6.14\ (0.65)$
		NM-BLR	$4.32\ (0.22)$	$9.87\ (1.69)$	$11.85\ (0.69)$

better). Even in this scenario, the performance provided by *DynaLT* is, on average, better than those of *NM-BLR*. However, the difference is significant only in the setting with $c = 3$. This suggests that monotonicity allows for a better estimate of the demand functions, especially if the environment changes frequently. Moreover, the performance for *DynaLT* achieved with different window sizes do not change significantly, suggesting that this method is less sensitive to changes in the window size.

4.7.2 Pricing Long-tail Products

The *DynaLT* has been used for two months on a real-world e-commerce website, comparing its performance with the pricing strategy previously used by the business managers.[4]

Setting In this test, we have a catalog $\mathcal{J}$ of $7,826$ products, with a turnover of 2.50 MEuros per year and a cumulative net margin of 0.53 MEuros (according to 2021 statistics). We divide the products into two sets defined by e-commerce specialists, according to both technical and market aspects, to set a proper A/B testing procedure. Specifically, $2,132$ products have been priced by the company's experts, while $5,694$ have been priced by *DynaLT*. From now on, we refer to the former one as the *control set* B and to the latter as the *test set* A. As showed previously in Figure 4.1, only $\approx 3\%$ of the products got on average at least 1 sale per week during 2021. We refer to this subset of products as *popular*, while the remaining ones are addressed as *long-tail*. Based on this classification, we further divide each one of the A and B sets into two subsets containing only *popular* products (A_P and B_P, respectively) and *long-tail* ones (A_{LT} and B_{LT}, respectively).

[4] Further details about the e-commerce website have been retained due to NDA.

The pricing process for the two above tests has been conducted over a test period $T = 8$ weeks, from November 22, 2021, to January 16, 2022. Since it is not possible to compute the regret $\hat{R}(\boldsymbol{\pi}, T)$ of the strategies in a real-world scenario, we evaluate the different pricing schemes in terms of their profits. Formally, the performances of the two strategies have been evaluated using the *rate of the profits* between the analyzed period and the one obtained in a control period C, from November 23, 2020, to January 17, 2021. Let us define the total profit achieved by *DynaLT* as:

$$M(\mathsf{A}, T) := \sum_{t \in T} \sum_{j \in \mathsf{A}} v_{jt} \overline{m}_{jt} c_j, \tag{4.10}$$

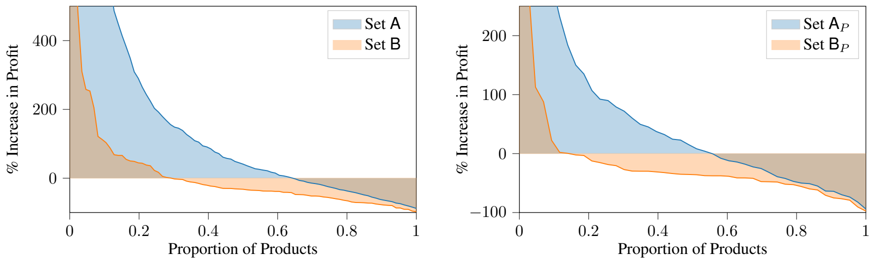
where $\overline{m}_{jt}$ is chosen by *DynaLT*. Similarly, we can define the profit $M(\mathsf{B}, T)$ achieved by human experts in the period T over the set B, and the profits $M(\mathsf{A}, C)$ and $M(\mathsf{B}, C)$ of *DynaLT* on the set A and human expert on the set B, respectively, over the period C.

The performance metric we adopt is:

$$G := \frac{M(\mathsf{A}, T)}{M(\mathsf{A}, C)} \frac{M(\mathsf{B}, C)}{M(\mathsf{B}, T)}. \tag{4.11}$$

Intuitively, G is greater than 1 if *DynaLT* increases the profit obtained during the period T w.r.t. period C more than human experts did. The *DynaLT* hyperparameters are set using historical data from 2016 to 2021. The algorithm is implemented in Python $3.8.5$, relying on the $2.5.0$ version of TensorFlow for what concerns the BLR model implementation, and runs over a Windows 10 machine (Intel Core i7-8750H @ 2.20GHz CPU with 16 GB of DDR4 system memory). Despite *DynaLT* took in charge $\approx 5,700$ products to price every week, the BLR model for demand estimation only had to fit in $\approx 1,200$ different instances. This highlights that the aggregation step is crucial to decrease the computational load of the pricing process, which reduces of $\approx 80\%$ the number of models that need to be fit. Thanks to the use of such an approach, the *DynaLT* algorithm runs on this architecture in ≈ 20 minutes, where ≈ 2 minutes are required for the aggregation step and the remaining ≈ 18 minutes are used for the training of the demand curve model.

Global Performance The results, in terms of performance index G, of the real-world experiment are summarized in Table 4.3. *DynaLT* algorithm over the set A provides an increase of $\approx 5\%$ in terms of profit during the period T w.r.t. period C. Instead, the choice of the experts over the set B provided a reduction of the profit of $\approx 25\%$. Therefore, the performance index G is ≈ 1.4.



(a) *Profit increase for the products in set* A *(blue) and* B *(orange).* **(b)** *Profit increase per single popular product in set* A_P *(blue) and set* B_P *(orange).*

Figure 4.8: *Profit increase by product.*

Table 4.3: *Performance in the real-world experiment.*

Popular G_P	Long-tail G_{LT}	Overall G
1.18	1.91	1.4

Figure 4.8a represents the increases in profit for each product in settings A (blue area) and B (orange areas). While the set A records an increase in profit w.r.t. last year in $\approx 63\%$ of the products, set B achieves a positive performance in $\approx 29\%$ of the products. This suggests that the improvement provided by *DynaLT* is due to a better pricing strategy over a large number of products.

Long-Tail and Best-Sellers Comparison As mentioned before, both products' sets A and B are mostly constituted by long-tail products. Indeed, $5,481$ out of $5,694$ products in A, and $2,078$ out of $2,132$ products in B are long-tail products. We will refer to the aforementioned subsets of long-tail products A_{LT} and B_{LT}, respectively. The performance index G computed over the new sets (using A_{LT} and B_{LT} in the definition in place of A and B, respectively) is $G_{LT} = 1.91$, suggesting that the pricing of long-tail products is significantly improved thanks to *DynaLT*. In the case of popular products, the improvement is smaller as we have $G_P = 1.18$. Nonetheless, in Figure 4.8b the profit increment for the popular products in the set A_P occurs for $\approx 55\%$ of the popular products, while in set B_P only in 14% of the cases we have an improvement.

4.8 Discussion and Conclusions

In this chapter, we faced the complex task of pricing products in an e-commerce scenario in the presence of both popular and long-tail products. Long-tail product commonly constitutes the majority of the ones present in a catalog, but automatic pricing methods are usually unable to handle them due to the scarcity of their transaction data. We proposed a modeling approach based on the demand curve's properties, which can speed up the demand curve learning process, and an aggregation strategy to automatically group products with too little data. The modeling approach has been tested on synthetically generated data to show the advantages of including the monotonicity property, and the overall *DynaLT* has been implemented in a real-world e-commerce website, showing that its application increases the profits on average of 18% w.r.t. what is gained by manually pricing the products.

CHAPTER 5

Autoregressive Bandits for Temporal Structures in Pricing

In this chapter, we propose a novel online decision-making setting, namely, Autoregressive Bandits (ARBs), in which the observed rewards are governed by an autoregressive process, whose parameters depend on the chosen action. We show that, under mild assumptions on the reward process, the optimal policy can be conveniently computed. Then, we propose `AutoRegressive Upper Confidence Bound` (`AR-UCB`) a new optimistic regret minimization algorithm, suffering regret in the order of $\widetilde{\mathcal{O}}\left(\frac{(n+1)^{3/2}\sqrt{kT}}{(1-\Gamma)^2}\right)$, where T is the time horizon, k is the number of actions, n is the order of the AR process, and $\Gamma < 1$ is an index characterizing the stability of the process.

This chapter presents (Bacchiocchi et al., 2024), a joint work with Francesco Bacchiocchi, Gianmarco Genalti, Davide Maran, Marcello Restelli, Nicola Gatti and Alberto Maria Metelli published at the *International Conference on Artificial Intelligence and Statistics (AISTATS)*. A preliminary version of this work (Bacchiocchi et al., 2023) appeared at the *European Workshop on Reinforcement Learning (EWRL)*.

5.1 Introduction

In a large variety of sequential decision-making problems, a learner is required to choose an action that, when executed, determines: (i) the immediate reward and (ii) the behavior of an underlying process that will influence, in some unknown manner, the future rewards. This process is influenced by the course of actions the agent performs and generates a temporal dependence between the sequence of observed rewards. A class of stochastic processes widely employed to model the temporal dependencies in real-world phenomena are the *autoregressive* (AR) processes (Hamilton, 2020). In this chapter, we model the reward of a sequential decision-making problem as an AR process whose parameters depend on the action selected by the agent at every round. This scenario can be represented as a particular class of continuous *reinforcement learning* problems (Sutton and Barto, 2018) where an AR process governs the temporal structure of the observed rewards through the action-dependent AR parameters that are unknown to the agent. It is worth mentioning that such a scenario displays notable differences compared to more traditional *non-stationary* learning problems. Indeed, in the scenario we address, the environment does not change, and the reward dynamics depend on the agent's course of actions only. Let us consider the pricing problem we want to address within this chapter.

Pricing Application *Consider the problem of finding the optimal price for a given product. A pricing strategy aims at maximizing a certain index, e.g., volumes, turnover, or profit. Usually, pricing algorithms focus on the* one-step *performance (Mueller et al., 2019). These solutions, however, fail in modeling the* long-term *phenomena that a pricing strategy inherently presents. Indeed, with one-step pricing algorithms, we fail (i) to model the long-term effect of our pricing strategy on customer loyalty and (ii) to capture the different demands of loyal and not-loyal customers. This problem, even if ubiquitous in the real world (Bowen and Chen, 2001), is unexplored in the literature, as existing approaches struggle to correctly deal with these autoregressive dynamics.*

Contributions In this chapter, motivated by the real-world problem presented above, we propose a novel setting, named *AutoRegressive Bandit* (ARB), in which the reward follows an AR process of order n whose parameters depend on the agent's actions. Importantly, we show that the optimal policy, differently from many bandit models, is *stationary* and *closed-loop*, as the optimal action depends on the previously observed rewards (Section 5.2). Then, we devise a new optimistic algorithm, namely

`AutoRegressive Upper Confidence Bound` (`AR-UCB`), able to learn the optimal policy in an online fashion (Section 5.3), and we show that it suffers sublinear regret of order $\widetilde{\mathcal{O}}\left(\frac{(n+1)^{3/2}\sqrt{kT}}{(1-\Gamma)^2}\right)$, where T is the optimization horizon, k is the number of actions, and $\Gamma < 1$ is a stability index of the process (Section 5.4). Finally, we empirically evaluate `AR-UCB` on both synthetic and real-world data, comparing its performance with several bandit baselines with competitive results and illustrating its notable robustness w.r.t. the misspecification of key parameters (Section 5.5).

5.2 Problem Formulation

In this section, we introduce the ARB setting, formalize the learning problem, how the learner interacts with the environment, assumptions, policies and definition of regret (Section 5.2.1). Subsequently, we derive a closed-form solution for the optimal policy of an ARB (Section 5.2.2).

5.2.1 Setting

We study the sequential interaction between a learner and an environment. Let $T \in \mathbb{N}$ be the learning horizon. At every round $t \in [\![T]\!]$, the learner chooses an action $a_t \in \mathcal{A} := [\![k]\!]$, among the $k \in \mathbb{N}$ available ones. In the ARB setting, the reward evolves according to an *autoregressive process* of order n (AR(n), Hamilton, 2020). Thus, the learner observes a noisy reward x_t of the form:

$$x_t = \gamma_0(a_t) + \sum_{i=1}^{n} \gamma_i(a_t) x_{t-i} + \xi_t, \tag{5.1}$$

where $x_t \in \mathcal{X}$ ($\mathcal{X} \subseteq \mathbb{R}$ is the reward space), $\gamma_0(a_t) \in \mathbb{R}$ and $(\gamma_i(a_t))_{i \in [\![n]\!]} \in \mathbb{R}^n$ are the unknown *parameters* depending on chosen action a_t, and ξ_t is a zero-mean σ^2-subgaussian random noise, independent conditioned to the past. The reward evolution can be expressed in an alternative form as follows:[1]

$$x_t = \langle \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle + \xi_t, \tag{5.2}$$

where $\mathbf{z}_{t-1} := (1, x_{t-1}, \dots, x_{t-n})^{\mathrm{T}} \in \mathcal{Z} := \{1\} \times \mathcal{X}^n$ is the *vector of past rewards* expressing past history, and $\boldsymbol{\gamma}(a) := (\gamma_0(a), \dots, \gamma_n(a))^{\mathrm{T}} \in \mathbb{R}^{n+1}$ is the *parameter vector*, defined for all the actions $a \in \mathcal{A}$. It is worth noting

[1] Although the linear structure might resemble the *contextual linear bandits* (Chu et al., 2011), the two settings are non-comparable. Indeed, in our ARBs the vector $\mathbf{z}_{t-1}$ is not sampled independently at every round, but, instead, follows a sequential process depending on the past, making the decision problem way more challenging.

that when $\gamma_i(a) = 0$ for all $i \in [\![n]\!]$ and $a \in \mathcal{A}$, the ARB setting reduces to a standard MAB (Auer et al., 2002a).

Assumptions We introduce the assumption that we employ in this chapter and comment on its role.

Assumption 5.1. *The parameters $(\gamma_i(a))_{i \in [\![0,n]\!]}$ fulfill the following conditions:*

a. (Non-negative coefficients) $\gamma_i(a) \geqslant 0$ for every $a \in \mathcal{A}$, $i \in [\![0, n]\!]$;
b. (Stability) $\Gamma := \max_{a \in \mathcal{A}} \sum_{i=1}^{n} \gamma_i(a) < 1$;
c. (Boundedness) $m := \max_{a \in \mathcal{A}} \gamma_0(a) < +\infty$.

Some comments are in order. Assumption 5.1.a requires that the coefficients of the AR process are non-negative. This scenario is ubiquitous in real-world AR phenomena (e.g., pricing, stock markets, digital advertising), where processes violating such an assumption will generate unrealistic sign alternation behaviors. Indeed x_t, in practice, represents the sales volume in the case of pricing, the value of a stock in the stock market, the number of customers that an e-commerce website may have, and so on. In all these real-world scenarios, the quantity x_t is meaningful whenever we consider non-negative values that we want to maximize, and when Assumption 5.1.a is not fulfilled (i.e., at least one $\gamma_i(a)$ is negative), the positivity of x_t is no longer ensured. Consider the example presented in Figure 5.1, where we present a general scenario in which, at time τ, we are in a given with a certain positive x_τ. Consider, for the sake of simplicity, a noiseless setting with $n = 1$ (i.e., an AR(1) process) and, for a given action i, we have $\gamma_0(a) = 0$. Consider now $\gamma_1(a) < 0$. Figure 5.1 shows what will happen in this case. The value of x_t continuously changes its sign at each time step, and this behavior is not compatible with the real-world phenomena of our interest. This is even more unrealistic if we think about the scenario in which we have another value of the state $\overline{x}_\tau > x_\tau$. In this scenario, after performing the same action i, we will observe that the best-starting state $\overline{x}_\tau$ leads to a worst next state $\overline{x}_{\tau+1} < x_{\tau+1}$. This behavior has no practical meaning in the applications of our interest. Given these considerations, we can derive that the worst possible effect of a given action is to *reset* the state, which corresponds to have $\gamma_1(a) = 0$. A representation of this phenomenon is drawn in Figure 5.2.

Assumption 5.1.b requires that the sum of $(\gamma_i(a))_{i \in [\![n]\!]}$ is limited to a value $\Gamma \in [0, 1)$ and Assumption 5.1.c enforces the boundedness of $\gamma_0(a)$, a standard assumption in stochastic MABs. These latter assumptions guarantee that the AR process does not diverge in expectation regardless of the sequence of the actions played.

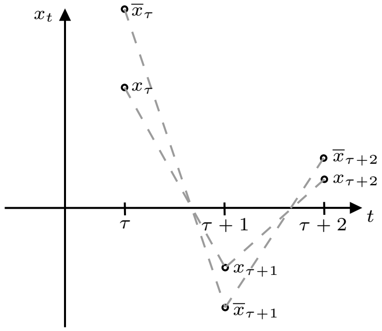


Figure 5.1: *An illustration of the effect of a negative $\gamma_1(a)$ over time.*

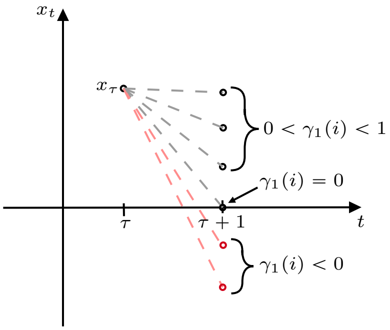


Figure 5.2: *The effect of $\gamma_1(a)$ in the evolution of the state x_t, in the case of a non-negative one (black), and a negative one (red).*

Policies and Regret The learner's behavior is modeled by a deterministic policy $\boldsymbol{\pi} = (\pi_t)_{t\in\mathbb{N}}$ defined, for every round $t \in \mathbb{N}$ as $\pi_t : \mathcal{H}_{t-1} \to \mathcal{A}$, mapping the history of observations $H_{t-1} = (x_0, a_1, x_1, \ldots, a_{t-1}, x_{t-1}) \in \mathcal{H}_{t-1}$ to an action $a_t = \pi_t(H_{t-1}) \in \mathcal{A}$ where $\mathcal{H}_{t-1} = \mathcal{X} \times (\mathcal{A} \times \mathcal{X})^{t-1}$ is the set of histories of length $t-1$. The performance of a policy $\boldsymbol{\pi}$ is evaluated in terms of the *expected cumulative reward* over the horizon $T \in \mathbb{N}$, defined as:

$$J_T(\boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=1}^{T} x_t\right] \qquad \text{with} \qquad \begin{cases} x_t = \langle \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle + \xi_t \\ a_t = \pi_t(H_{t-1}) \end{cases}, \tag{5.3}$$

where the expectation is taken w.r.t. the randomness of the reward noise ξ_t. A policy $\boldsymbol{\pi}^*$ is *optimal* if it maximizes the expected average reward, i.e., $\boldsymbol{\pi}^* \in \arg\max_{\boldsymbol{\pi}} J_T(\boldsymbol{\pi})$, whose performance is denoted as $J_T^* := J_T(\boldsymbol{\pi}^*)$. The goal of the learner is to minimize the *expected cumulative (policy) regret* by playing a policy $\boldsymbol{\pi}$, competing against the optimal policy $\boldsymbol{\pi}^*$ over a *learning horizon* $T \in \mathbb{N}$:

$$R(\boldsymbol{\pi}, T) = J_T^* - J_T(\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=1}^{T} r_t\right], \tag{5.4}$$

where $r_t := x_t^* - x_t$ is the instantaneous policy regret and $(x_t^*)_{t\in[\![T]\!]}$ is the sequence of rewards observed by playing the optimal policy $\boldsymbol{\pi}^*$.

Pricing Application (cont.) *The problem of optimal pricing presented in Section 5.1 can be easily mapped to the ARB setting. Consider the scenario* *in which we want to maximize the volumes over time. The volumes are our* reward x_t*, and the history of our rewards* $x_{t-1}, \dots, x_{t-n}$ *provides an indication of the loyal customer pool over* n *units of time (e.g., days, weeks). The ARB setting allows modeling a reward which is the contribution of both new customers (via* γ_0*) and the loyal customer pool (via* $\gamma_1, \dots, \gamma_n$*). Specifically, the price, our* action a_t*, induces different values of the coefficient* $\boldsymbol{\gamma}(a_t)$*, to represent the different demand curves that loyal and new customers might have.*

5.2.2 Optimal Policy

In this section, we derive a closed-form expression for the optimal policy $\boldsymbol{\pi}^*$ for the expected cumulative reward of Equation (5.3), under Assumption 5.1.a.

Theorem 5.2.1 (Optimal Policy)**.** *Under Assumption 5.1.a, for every round* $t \in \mathbb{N}$*, the optimal policy* $\pi_t^*(H_{t-1})$ *satisfies:*

$$\pi_t^*(H_{t-1}) \in \arg\max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \mathbf{z}_{t-1} \rangle. \tag{5.5}$$

This result deserves some comments. First, the optimal action depends on the vector of past rewards $\mathbf{z}_{t-1}$ and, thus, on the most recent n rewards $x_{t-1}, \dots, x_{t-n}$ only. Thus, the optimal policy $\boldsymbol{\pi}^*$ is non-Markovian with memory n or, equivalently, Markovian w.r.t. the state representation $\mathbf{z}_{t-1}$.[2] Second, the optimal action maximizes, at every round $t \in \mathbb{N}$, the *expected instantaneous reward* $\mathbb{E}[x_t | H_{t-1}] = \langle \boldsymbol{\gamma}(a), \mathbf{z}_{t-1} \rangle$. This is a consequence of the non-negativity of the parameters $\gamma_i(a)$ (Assumption 5.1.a), which enforces a meaningful evolution of the AR process, compatible with our real-world motivating scenarios. This way, the action maximizing the expected *immediate* reward (i.e., a *myopic* policy) is optimal for the expected *cumulative* reward too. The proof can be found in Appendix B.

5.3 AutoRegressive Upper Confidence Bound

In this section, we present `AutoRegressive Upper Confidence Bound` (`AR-UCB`), an optimistic regret minimization algorithm for the ARB setting whose pseudo-code is reported in Algorithm 5.1. `AR-UCB` leverages the myopic optimal policy for ARBs (Theorem 5.2.1) and implements an incremental regularized least squares procedure to estimate the

[2] We can look at the ARB as a particular *Markov Decision Processes* (Puterman, 2014) with $\mathbf{z}_{t-1} \in \mathcal{Z}$ as state representation.

unknown parameters $\boldsymbol{\gamma}(a)$, for every action $a \in \mathcal{A}$ independently. The algorithm requires the knowledge of the order n of the AR process, although this knowledge can be replaced with the one of an upper bound $\overline{n} > n$ of the AR order.[3]

AR-UCB starts by initializing for all the actions $a \in \mathcal{A}$ the Gram matrix $\mathbf{V}_0(a) = \lambda \mathbf{I}_{n+1}$, where $\lambda > 0$ is the Ridge regularization parameter, the vectors $\mathbf{b}_0(a) = \widehat{\boldsymbol{\gamma}}_0(a) = \mathbf{0}_{n+1}$, and the observations vector $\mathbf{z}_0 = (1, 0, \dots, 0)^T$ (Line 1).[4] Then, for each round $t \in [\![T]\!]$, AR-UCB computes the *Upper Confidence Bound* (UCB) index (Line 3) for every $a \in \mathcal{A}$ and the optimistic action a_t:

$$a_t \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \ \mathrm{UCB}_t(a) := \langle \widehat{\boldsymbol{\gamma}}_{t-1}(a), \mathbf{z}_{t-1} \rangle + \beta_{t-1}(a) \left\| \mathbf{z}_{t-1} \right\|_{\mathbf{V}_{t-1}(a)^{-1}}, \tag{5.6}$$

where $\widehat{\boldsymbol{\gamma}}_{t-1}(a)$ is the most recent estimate of the parameter vector $\boldsymbol{\gamma}(a)$, $\mathbf{z}_{t-1} = (1, x_{t-1}, \dots, x_{t-n})^{\mathrm{T}}$ is the observations vector, and $\beta_{t-1}(a) \geqslant 0$ is an exploration coefficient that will be defined later (Section 5.4). The index $\mathrm{UCB}_t(a)$ is designed to be optimistic, i.e., $\langle \boldsymbol{\gamma}(a), \mathbf{z}_{t-1} \rangle \leqslant \mathrm{UCB}_t(a)$ with high probability for all $a \in \mathcal{A}$. Then, action a_t is executed (Line 4) and the new reward x_t is observed. This sample is employed to update the Gram matrix estimate $\mathbf{V}_t(a_t)$, the vector $\mathbf{b}_t(a_t)$, and the estimate $\widehat{\boldsymbol{\gamma}}_t(a_t)$ (lines 6-8).

5.4 Regret Analysis

In this section, we present the analysis of the regret of AR-UCB. We start providing a self-normalized concentration inequality for estimating the AR parameters $\boldsymbol{\gamma}(a)$ (Section 5.4.1). Then, we derive a decomposition of the regret (Section 5.4.2) that is useful to complete the analysis and, finally, we present the bound on the expected cumulative (policy) regret (Section 5.4.3). The complete proofs of the theorems stated in this section can be found in Appendix B.

5.4.1 Concentration Inequality for Parameter Vectors

We start by providing a concentration result for the estimates $\widehat{\boldsymbol{\gamma}}_t(a)$ of the true parameter vector $\boldsymbol{\gamma}(a)$, for every action $a \in \mathcal{A}$, as performed in Algorithm 5.1. At the end of each round $t \in \mathbb{N}$, where the chosen action

[3] Indeed, any AR process of order n can be regarded as an AR process of order $\overline{n} > n$ setting $\gamma_i(a) = 0$ for $i \in [\![n+1, \overline{n}]\!]$. An empirical validation of the AR-UCB performances in the case of a misspecified n is provided in Section 5.5.4.

[4] We assume to know the initial observations vector $\mathbf{z}_0$. If this is not the case, we can play an arbitrary action for the first n rounds to observe $(x_t)_{t \in [\![n]\!]}$ with just an additional constant loss term.

Algorithm 5.1: AR-UCB.

```
Input: Regularization parameter $\lambda > 0$, autoregressive order $n$, exploration
       coefficients $(\beta_{t-1})_{t\in[\![T]\!]}$
Initialize $t \leftarrow 1$, $\mathbf{V}_0(a) = \lambda \mathbf{I}_{n+1}$, $\mathbf{b}_0(a) = \mathbf{0}_{n+1}$, $\widehat{\gamma}_0(a) = \mathbf{0}_{n+1}$, $\forall a \in \mathcal{A}$,
  $\mathbf{z}_0 = (1, 0, \dots, 0)^T$
for $t \in [\![T]\!]$ do
    Compute
      $a_t \in \arg\max_{a\in\mathcal{A}} \mathrm{UCB}_t(a) := \langle \widehat{\gamma}_{t-1}(a), \mathbf{z}_{t-1} \rangle + \beta_{t-1}(a) \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(a)^{-1}}$
    Play action $a_t$ and observe $x_t = \langle \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle + \xi_t$
    Update $\forall a \in \mathcal{A}$:
        $\mathbf{V}_t(a) = \mathbf{V}_{t-1}(a) + \mathbf{z}_{t-1}\mathbf{z}_{t-1}^{\mathrm{T}} \mathbb{1}_{\{a=a_t\}}$
        $\mathbf{b}_t(a) = \mathbf{b}_{t-1}(a) + \mathbf{z}_{t-1} x_t \mathbb{1}_{\{a=a_t\}}$
        $\widehat{\gamma}_t(a) = \mathbf{V}_t(a)^{-1} \mathbf{b}_t(a)$
    Update $\mathbf{z}_t = (1, x_t, \dots, x_{t-n+1})^T$
    $t \leftarrow t + 1$
end
```

is $a_t \in \mathcal{A}$, we solve the Ridge-regularized linear regression problem and update the coefficient vector estimate $\widehat{\gamma}_t(a_t)$ associated to a_t:

$$\widehat{\gamma}_t(a_t) = \operatorname*{arg\,min}_{\widetilde{\gamma}\in\mathbb{R}^{n+1}} \sum_{l\in\mathcal{O}_t(a_t)} (x_l - \langle \widetilde{\gamma}, \mathbf{z}_{l-1} \rangle)^2 + \lambda \|\widetilde{\gamma}\|_2^2 = \mathbf{V}_t(a_t)^{-1} \mathbf{b}_t(a_t),$$

where $\mathcal{O}_t(a)$ is the set of rounds where action a has been chosen, i.e., $\mathcal{O}_t(a) := \{\tau \in [\![t]\!] : a_\tau = a\}$. The following result shows how the estimate $\widehat{\gamma}(a)$ concentrates around the true parameters $\boldsymbol{\gamma}(a)$ over the rounds.

Lemma 5.4.1 (Self-Normalized Concentration). *Let $a \in \mathcal{A}$ be an action, let $(\widehat{\gamma}_t(a))_{t\in\mathcal{O}_\infty(a)}$ be the sequence of solutions to the Ridge regression problems computed by Algorithm 5.1. Then, for every regularization parameter $\lambda > 0$, confidence $\delta \in (0, 1)$, simultaneously for every round $t \in \mathbb{N}$ and action $a \in \mathcal{A}$, with probability at least $1 - \delta$ it holds that:*

$$\|\widehat{\gamma}_t(a) - \boldsymbol{\gamma}(a)\|_{\mathbf{V}_t(a)} \leqslant \sqrt{\lambda} \|\boldsymbol{\gamma}(a)\|_2 + \sigma \sqrt{2 \log\left(\frac{k}{\delta}\right) + \log\left(\frac{\det \mathbf{V}_t(a)}{\lambda^{n+1}}\right)}.$$

Lemma 5.4.1 resembles the self-normalized concentration inequality from (Abbasi-Yadkori et al., 2011, Theorem 1). However, contrary to Lin-UCB (Abbasi-Yadkori et al., 2011), the exploration coefficients $\beta_t(a)$ are different for every action $a \in \mathcal{A}$. Lemma 5.4.1 allows properly defining the exploration coefficients $\beta_t(a)$ employed in Algorithm 5.1, defined for every action $a \in \mathcal{A}$ and round $t \in [\![0, T-1]\!]$:

$$\beta_t(a) := \sqrt{\lambda(m^2+1)} + \sigma\sqrt{2\log\left(\frac{k}{\delta}\right) + \log\left(\frac{\det \mathbf{V}_t(a)}{\lambda^{n+1}}\right)}. \tag{5.7}$$

This formula contains two terms. The first one is a *bias* term that increases with m (i.e., the maximum value of the largest $\gamma_0(a)$ over the actions $a \in \mathcal{A}$, see Assumption 5.1.c) and with the regularization parameter of the Ridge regression $\lambda > 0$. The second one is the *concentration* term and increases with the subgaussian parameter σ of the noise, the number of actions k, and the determinant of the design matrix $\mathbf{V}_t(a)$, but decreases in λ. It is worth noting that $\beta_t(a)$ is obtained from Lemma 5.4.1, by observing that, under Assumptions 5.1.b and 5.1.c, we have $\|\boldsymbol{\gamma}(a)\|_2 \leqslant \sqrt{m^2+\Gamma^2} \leqslant \sqrt{m^2+1}$. Thus, the exploration coefficient $\beta_t(a)$ ensures that, with probability $1-\delta$, the following inequality holds simultaneously for all actions $a \in \mathcal{A}$ and rounds $t \in [\![0, T-1]\!]$:

$$\|\widehat{\boldsymbol{\gamma}}_t(a) - \boldsymbol{\gamma}(a)\|_{\mathbf{V}_t(a)} \leqslant \beta_t(a). \tag{5.8}$$

We observe that $\beta_t(a)$ (see Equation 5.7) and AR-UCB do not require the knowledge of the maximum sum Γ of the parameters $\gamma_i(a)$ over the actions (see Assumption 5.1.b). This is a desirable feature of our algorithm as Γ is often unknown in practice and difficult to upper bound or estimate. Nevertheless, Γ appears in the regret analysis in Section 5.4.2. Differently, the value of m, needed to compute the optimistic coefficient $\beta_t(a)$, can be easily replaced with an upper bound $\overline{m} > m$ when unknown.[5]

5.4.2 Regret Decomposition

In this section, we present a novel *decomposition* of the regret that will be employed in the final bound of Section 5.4.3. The contents of this section are of independent interest and applicable to any learner's policy $\boldsymbol{\pi}$, beyond AR-UCB. From a technical perspective, the analysis is composed of two steps: (i) we decompose the instantaneous (policy) regret r_t in terms of the instantaneous *external regret* ρ_t (Lemma 5.4.2); (ii) we bound the cumulative expected (policy) regret $R(\boldsymbol{\pi}, T) = \mathbb{E}[\sum_{t=1}^{T} r_t]$ in terms of the expected cumulative external regret $\varrho(\boldsymbol{\pi}, T) = \mathbb{E}[\sum_{t=1}^{T} \rho_t]$ (Lemma 5.4.3).

We start with step (i), by recalling that the definition of cumulative expected (policy) regret $R(\boldsymbol{\pi}, T)$ in Equation (5.4) compares the sequence of rewards $(x_t^*)_{t\in[\![T]\!]}$ when executing the optimal policy $\boldsymbol{\pi}^*$ with the sequence

[5] An empirical analysis of the effect of the misspecification of such a parameter is provided in Section 5.5.3.

of rewards $(x_t)_{t \in [\![T]\!]}$ when executing the learner's policy $\boldsymbol{\pi}$. However, in our ARB setting, the observed reward x_t depends on the past history H_{t-1}. Thus, the instantaneous (policy) regret $r_t := x_t^* - x_t$ can be decomposed in two terms: (a) the dissimilarity between the past history H_{t-1}^* when executing the optimal policy and the learner's observed history H_{t-1}; (b) the instantaneous *external regret* (Dekel et al., 2012) $\rho_t := \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle$ representing the loss of executing the learner action a_t instead of the optimal one $a_t^* = \pi_t^*(H_{t-1}^*)$ *assuming* that such actions are applied to the observations vector $\mathbf{z}_{t-1}$ generated by the execution of the learner's policy. The following result formalizes the instantaneous regret decomposition.

Lemma 5.4.2 (Policy Regret Decomposition). *Let $(x_t^*)_{t \in [\![T]\!]}$ be the sequence of rewards by executing the optimal policy $\boldsymbol{\pi}^*$ and let $(x_t)_{t \in [\![T]\!]}$ be the sequence of rewards by executing the learner's policy $\boldsymbol{\pi}$. Then, for every $t \in [\![T]\!]$ it holds that:*

$$\begin{aligned} r_t &= x_t^* - x_t \\ &= \sum_{i=1}^{n} \gamma_i(a_t^*)(x_{t-i}^* - x_{t-i}) + \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle \\ &= \sum_{i=1}^{n} \gamma_i(a_t^*) r_{t-i} + \rho_t, \end{aligned} \tag{5.9}$$

where $r_t := x_t^* - x_t$ is the instantaneous policy regret, $\rho_t := \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle$ is the instantaneous external regret, $a_t^* = \pi_t^*(H_{t-1}^*)$, and $r_{t-i} = 0$ if $i \geqslant t$.

The decomposition in Equation (5.9) comprises two terms. The second one ρ_t is the instantaneous external regret discussed above. The first one defines a recurrence relation of order n on the instantaneous policy regret r_t. We now move to step (ii) with the following result that shows that the contribution of the recurrence can be reduced to a term depending on Γ and n that multiplies the cumulative external regret.

Lemma 5.4.3 (External-to-Policy Regret Bound). *Let $\boldsymbol{\pi}$ be the learner's policy and $T \in \mathbb{N}$ be the horizon. Under Assumptions 5.1.a and 5.1.b, it holds that:*

$$\mathbb{E}[R(\boldsymbol{\pi}, T)] = \mathbb{E}\left[\sum_{t=1}^{T} \left[\sum_{i=1}^{n} \gamma_i(a_t^*) r_{t-i} + \rho_t\right]\right] \leqslant \left(\frac{\Gamma n}{1-\Gamma} + 1\right) \varrho(\boldsymbol{\pi}, T), \tag{5.10}$$

where $\varrho(\boldsymbol{\pi}, T) := \mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right]$ is the cumulative expected external regret.

Lemma 5.4.3 provide us a bound on the cumulative expected (policy) regret $R(\boldsymbol{\pi}, T)$ achieved by `AR-UCB` (or any algorithm playing in an ARB) by bounding the cumulative expected external regret $\varrho(\boldsymbol{\pi}, T)$. The order of the regret bound w.r.t. T is governed by the external regret, while the effect of a *weaker* history (i.e., the sub-optimal actions of the past) emerges as an instance-specific constant. Such a constant is 1 whenever $n = 0$ or $\Gamma = 0$, i.e., when the ARB reduces to a standard MAB. In all other cases, the bigger the value of n or Γ, the more visible the AR effects are, and, consequently, the more the sub-optimal choices of the past get amplified. Finally, we point out that the multiplicative factor $\frac{\Gamma n}{1-\Gamma} + 1$ to pass from external to policy regret is tight since there exists a sequence of external regrets in which the inequality of Lemma 5.4.3 holds with equality (see Appendix B).

5.4.3 Regret Bound

In the following, we present a bound on the expected policy regret bound for `AR-UCB`.

Theorem 5.4.4. *Let $\delta = (2T)^{-1}$. Under Assumptions 5.1.a, 5.1.b, and 5.1.c, `AR-UCB` suffers a cumulative expected (policy) regret bounded by (highlighting the dependence on m, σ, n, Γ, k, and T only):*

$$\mathbb{E}[R(\texttt{AR-UCB}, T)] \leqslant \widetilde{\mathcal{O}}\left(\frac{(m+\sigma)(n+1)^{3/2}\sqrt{kT}}{(1-\Gamma)^2} \right).$$

Some observations are in order. First, when we set $n = 0$ and $\Gamma = 0$, i.e., we reduce the ARB to a standard MAB, we obtain a regret rate of $\widetilde{\mathcal{O}}((m+\sigma)\sqrt{kT})$, which is tight for standard MABs. The quantity $\frac{m+\sigma}{1-\Gamma}$ is the maximum value that rewards can achieve, as proven in Lemma B.0.1. As intuition suggests, the ARB learning problem becomes more challenging as the AR order n increases and when the bound on the sum of the parameters Γ approaches one. This is witnessed in Theorem 5.4.4 with the dependence of the regret on $(n+1)^{3/2}$ and $(1-\Gamma)^{-1}$. The interplay between n and $(1-\Gamma)^{-1}$ is showing that even if two instances have the same sum of parameters (i.e., Γ), the one with fewer coefficients (i.e., n) is more easily learnable. This is explained by the fact that our algorithm learns the individual parameters by means of a regression procedure learning to a $\sqrt{n+1}$ in the regret. Finally, suppose we run `AR-UCB` with a larger AR order $\overline{n} > n$. In such a case, the dependence on $(n+1)^{3/2}$ becomes $(n+1)(\overline{n}+1)^{1/2}$, since the factor due to passing from external to policy regret (Lemma 5.4.3) will always contain the true n, while $\overline{n}$ appears because of the estimation process. Similarly, if we execute AR-UCB with a value $\overline{m} > m$, the regret bound still holds by replacing m with $\overline{m}$.

5.5 Numerical Simulations

In this section, we first provide (Section 5.5.1) a numerical validation of AR-UCB compared with other bandit baselines in synthetically-generated domains. Then, we discuss (Section 5.5.2) the importance of exploiting the noise in this setting, and, subsequently, we analyze the sensitivity of AR-UCB to the misspecification of the three most important parameters, i.e., m (Section 5.5.3), n (Section 5.5.4), and σ (Section 5.5.5). In Sections 5.5.6 and 5.5.7, we provide experimental results in the particular case of processes of order 0 (i.e., a standard stochastic MAB) and 1 (i.e., AR(1)). Finally, in Section 5.5.8, we conduct validation in a setting generalized from real-world data. The code containing the AR-UCB algorithm, as well as the environements used in this section, can be found at https://github.com/marcomussi/ARB.

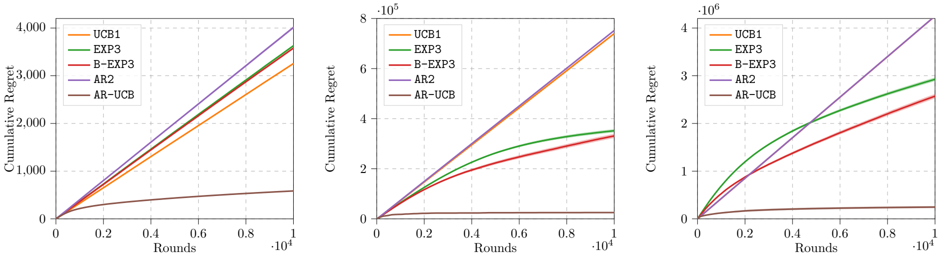
5.5.1 AR-UCB vs Bandit Baselines

Setting We evaluate AR-UCB in three scenarios that differ in the properties of the autoregressive processes that govern the rewards. The competing algorithms are evaluated in terms of cumulative regret w.r.t. the setting-specific clairvoyant. The three settings have their AR(n) process order $n \in \{2, 4\}$, number of actions $k \in \{2, 7\}$, and scale $m \in \{1, 20, 920\}$. The values of $\boldsymbol{\gamma}(a)$ have been sampled from uniform probability distributions for each action $a \in \mathcal{A}$ and for each setting. The environments are noisy with a standard deviation $\sigma \in \{0.75, 1.5, 10\}$. We chose to set the hyper-parameters of AR-UCB as follows: $\lambda = 1$, while $\overline{m} \in \{10, 100, 1000\}$, that is equivalent to chose $\overline{m}$ of the same magnitude of the true value m, in a pessimistic fashion. Table 5.1 summarizes the details of the three environments.

Baselines AR-UCB will compete with several bandit baselines. First, it is compared with UCB1 (Lai and Robbins, 1985; Auer et al., 2002a), a widely adopted solution for stochastic MABs. Second, we consider Exp3, designed for adversarial MABs (Auer et al., 1995, 2002b) and its extension to finite-memory adaptive adversaries B-Exp3 (Dekel et al., 2012). Lastly, we compare AR-UCB with AR2 (Chen et al., 2023), an algorithm for managing AR(1) processes. The hyper-parameters chosen for the baselines are the ones proposed in the original papers.

	Parameters			
Setting	n	k	m	σ
A	2	2	1	0.75
B	4	7	20	1.5
C	4	7	920	10

Table 5.1: *Settings description.*



(a) *Setting* A. **(b)** *Setting* B. **(c)** *Setting* C.

Figure 5.3: *Cumulative regret of* AR-UCB *and multiple baselines (100 runs, mean ± std).*

Results Figure 5.3 shows the average cumulative regrets for AR-UCB and the other bandit baselines. We observe that AR-UCB suffers the smallest cumulative regret in these scenarios, always displaying a sublinear behavior. Both Exp3 and B-Exp3 in two scenarios out of three (B and C) achieve sublinear regret. On the other hand, both UCB1 and AR2 are not able to achieve sublinear regret in the presented scenarios. This is not surprising since we require them to learn more complex processes than those they are designed for (i.e., models with $n = 0$ and $n = 1$ for UCB1 and AR2, respectively).

5.5.2 On the Effect of Stochasticity

The optimal policy (Theorem 5.2.1) for the ARB setting exploits the contribution of the noise to increase the collected reward. In this section, we provide experimental evidence of this phenomenon. We first introduce a notion of *optimal policy without noise*. Then, we conduct an experiment to highlight the variations between the two policies in environments presenting different noise magnitudes.

Optimal Policy without Noise The optimal policy, when no noise is involved, is *constant* and corresponds, for sufficiently large T, to playing the

σ	`Stochastic`	`Deterministic`
0	**19994 (0)**	**19994 (0)**
0.1	**20167 (0.20)**	19998 (2.04)
0.5	**22049 (1.02)**	20012 (1.02)
1.0	**24504 (2.04)**	20030 (2.04)
2.0	**29428 (4.09)**	20067 (4.08)

Table 5.2: *Cumulative reward of the Clairvoyant* `Stochastic` *and* `Deterministic` *policies (*100 *runs, mean (std)).*

action $a^+ \in \mathcal{A}$ that brings the system to the most profitable steady state.[6] Such an action a^+ is the one maximizing the *steady-state reward*, namely:

$$a^+ \in \arg\max_{a \in \mathcal{A}} \frac{\gamma_0(a)}{1 - \sum_{i=1}^{n} \gamma_i(a)}. \tag{5.11}$$

It is worth noting the role of Assumption 5.1.b which guarantees the existence of the inverse $(1 - \sum_{i=1}^{n} \gamma_i(a))^{-1} \geqslant (1 - \Gamma)^{-1}$ for each action $a \in \mathcal{A}$. The proof can be found in Appendix B.1.

Setting To demonstrate the importance of the noise in this setting, we consider the two clairvoyant policies defined above. We compare the optimal `Stochastic` policy (Equation 5.5) and the optimal policy for the `Deterministic` setting (Equation 5.11). The setting selected is challenging and made of $n = 2$ actions, a_1 and a_2, that are very close in terms of expected steady-state reward:

$$\boldsymbol{\gamma}(a_1) = (1,\ \rho,\ 0)^{\mathrm{T}} \qquad\qquad \boldsymbol{\gamma}(a_2) = (1,\ 0,\ \rho - \epsilon)^{\mathrm{T}},$$

where $\rho = 0.5$, $\epsilon = 0.02$ and the noise is Gaussian considering values of $\sigma \in \{0, 0.1, 0.5, 1.0, 2.0\}$.

Results Table 5.2 shows the performance of the two policies in terms of cumulative reward. First, with no noise (i.e., $\sigma = 0$), the performances of the two policies are equivalent. However, when we consider a stochastic setting (i.e., $\sigma > 0$), the `Stochastic` policy can exploit the beneficial effect of the noise in order to increase the average reward. Indeed, the optimal `Deterministic` policy retrieves almost the same reward for all

[6] The request for large T is to make transient effects neglectable.

the tested values of σ, while `Stochastic` policy increases its average reward as much as the system is noisy (since it can exploit it).

5.5.3 On the Knowledge of Parameter m

A fundamental parameter of `AR-UCB` is the value $m = \max_{a \in \mathcal{A}} \gamma_0(a)$. In this experiment, we empirically show that any choice in the same order of magnitude as the actual value will let the algorithm achieve a sublinear regret, while severe underestimation prevents the algorithm from achieving a sublinear cumulative regret.

Setting We run multiple simulations varying the value of parameter $\overline{m}$. We chose $k = 7$, $n = 4$ and $\gamma_0(a) = 500$ for every action $a \in \mathcal{A}$ (i.e., $m = 500$). The autoregressive parameters $\gamma_i(a)$ have been sampled from a uniform probability distribution with support in $[0, 1/4 - \epsilon]$, where $\epsilon > 0$ is an arbitrarily small value. For this experiment, we test values $\overline{m} \in \{1, 10, 100, 500, 1000, 2500\}$.

Results In Figure 5.4, we report the cumulative regrets of `AR-UCB` under different choices of $\overline{m}$. First, it is worth noting how choosing values of $\overline{m} \geqslant m$ always results in a sublinear cumulative regret, with a progressive increase as $\overline{m}$ gets larger. This is highlighted when comparing, for instance, the scenario where $\overline{m} = 2500$ to the one where $\overline{m} \in \{500, 1000\}$. When $\overline{m}$ is underestimated, we empirically observe two facts. When $\overline{m}$ is in the same order of magnitude as the true value m (e.g., $\overline{m} = 100$), we empirically get a smaller sublinear cumulative regret (even if no theoretical guarantees are present). Finally, a severe underestimation of the parameter leads to a linear cumulative regret, as clearly visible for $\overline{m} \in \{1, 10\}$, although, in these settings, the cumulative regret is lower w.r.t. the other settings in the very first stages of the simulations (due to a more limited exploration).

5.5.4 On the Knowledge of the Autoregressive order n

As stated in Section 5.4, `AR-UCB` can also run under a misspecified parameter $\overline{n} \neq n$. In this section, we provide an empirical analysis of the effect of misspecifying such a value.

Setting We consider a configuration with $k = 7$, $n = 10$, $\gamma_0(a) = 1$ and $\gamma_i(a)$ for $i \geqslant 1$ sampled from a uniform distribution having support in $[0, 10^{-2} \cdot 2i)$ for every action $a \in \mathcal{A}$. `AR-UCB` is run varying the parameter $\overline{n} \in \{1, 2, 4, 8, 10, 16\}$.

Results Figure 5.5 reports the average cumulative regret for the consid-

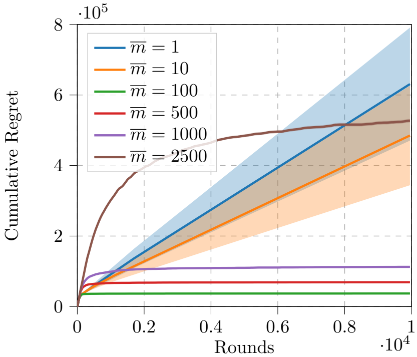


Figure 5.4: *Effect of the choice of parameter $\overline{m}$ on the AR-UCB cumulative regret (100 runs, mean ± std).*

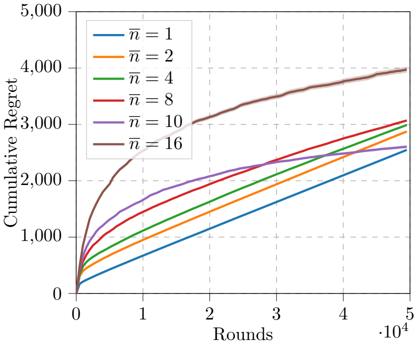


Figure 5.5: *Effect of the choice of parameter $\overline{n}$ on the AR-UCB cumulative regret (100 runs, mean ± std).*

ered values of $\overline{n}$. On the one hand, an underestimation of parameter n (i.e., $\overline{n} \in \{1, 2, 4\}$) results in an asymptotically linear cumulative regret. This effect is justified since AR-UCB is not able to learn the actual AR dynamics due to underfitting, i.e., the considered models are too simple. On the other hand, AR-UCB achieves sublinear cumulative regret when $\overline{n} \geqslant n$ (i.e., $\overline{n} \in \{10, 16\}$). In particular, when $\overline{n} > n$, the linear models use more parameters than required, resulting in slower learning. However, as the samples increase, the algorithm learns that the exceeding coefficients are not significant. A particular case is when $\overline{n}$ is close to n but strictly lower (i.e., $\overline{n} = 8$). Here, the cumulative regret degenerates to linear, but if the coefficients $\gamma_j(a)$ for $j \in [\![\overline{n} + 1, n]\!]$ are not very large, the performance of AR-UCB with misspecified $\overline{n}$ results, in practice, close to the one obtained with the true n.

5.5.5 On the Knowledge of Parameter σ

Another quantity required in order to execute AR-UCB is the process noise's standard deviation σ. In this experiment, we empirically show that our algorithm works in severe misspecification of this quantity. From now on, we will refer to this input parameter as $\overline{\sigma}$.

Setting We evaluate AR-UCB under different specifications of the value $\overline{\sigma} \in \{10^{-2}, 10^{-1}, 1, 15, 30\}$, when the true value is set to $\sigma = 15$. The other relevant experiment parameters are $m = 1200$, $n = 10$ and $k = 7$. The autoregressive parameters $\gamma_i(a)$ have been sampled from a uniform probability distribution.

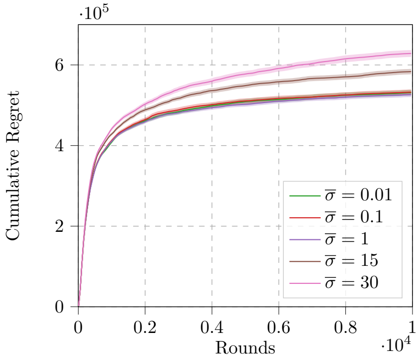


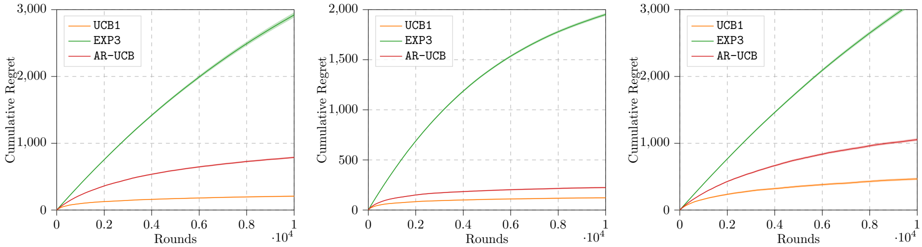
Figure 5.6: *Cumulative regret of* AR−UCB *in the case of misspecification of process noise parameter σ (100 runs, mean ± std).*

Results In Figure 5.6, we report the cumulative regrets of AR−UCB under different choices of $\overline{\sigma}$. First, it is worth noting how all the choices of $\overline{\sigma}$ always result in a sublinear cumulative regret, progressively increasing as $\overline{\sigma}$ gets larger. Even if there are no theoretical guarantees of expected sublinear regret when $\overline{\sigma}$ is misspecified from below, empirically, this type of misspecification is not so evident in the performance.

5.5.6 Stochastic Bandit Problem

Setting We evaluate AR−UCB in the special case $n = 0$. This problem is equivalent to solving a standard stochastic bandit problem. This experiment compares the performances of AR−UCB in this setting against well-known gold standards: UCB1 and Exp3. The competing algorithms are evaluated in terms of cumulative regret w.r.t. the setting-specific clairvoyant. The three settings differ in the values of $m \in \{2, 7.5\}$ (i.e., the maximum arms' expected reward) and the values of $\sigma \in \{0.9, 1.25, 2\}$, the noise's standard deviation. The number of actions is $k = 7$.

Results Figure 5.7 shows the average cumulative regrets for AR−UCB, UCB1, and Exp3. We immediately observe that all the algorithms suffer sublinear cumulative regret, as expected since they are all able to provide no-regret theoretical guarantees in this setting. In all the experiments, UCB1 outperforms all the other algorithms since it is specifically designed for the scenario under analysis. AR−UCB, as expected, performs properly in this setting since, as already discussed in Section 5.4.3, its regret is asymptotically optimal when $n = 0$.



(a) *Setting* A.

(b) *Setting* B.

(c) *Setting* C.

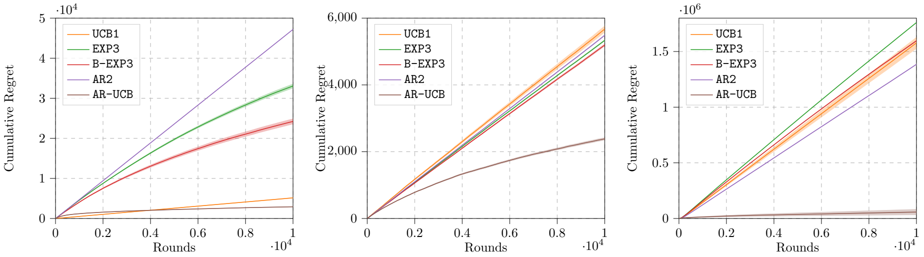
Figure 5.7: *Cumulative regret of* AR-UCB*,* UCB1*, and* Exp3 *in the case of* $n = 0$ *(100 runs, mean* ± *std).*

5.5.7 AR(1) Bandit Problem

AR(1) processes are the simplest autoregressive processes. Therefore, we will present a specific analysis of this setting to show how AR-UCB and the baselines perform when the complexity given by the dynamic temporal structure is minimal. Results show how even the minimal autoregressive contribution can lead all the algorithms (except for AR-UCB) to linear cumulative regret.

Setting We evaluate AR-UCB in the case $n = 1$. This is the simplest setting in which an autoregressive component contributes to the reward. This experiment compares the performances of AR-UCB in this setting against the same baselines as Section 5.5.1. The competing algorithms are evaluated in terms of cumulative regret w.r.t. the setting-specific clairvoyant. The three settings differ in the values of $m \in \{2, 8, 10\}$ (i.e., the maximum arms' expected reward) and the values of $\sigma \in \{1, 1.25, 2\}$, the noise's standard deviation. The values of the $\gamma_1(a)$ parameters have been sampled from uniform distributions having their sampling ranges inside $[0, 1)$. The number of actions is $k = 7$.

Results Figure 5.8 shows the average cumulative regrets for all the competing algorithms. We immediately observe that the only algorithms able to achieve sublinear regret are AR-UCB (in all three settings), B-Exp3 (first and third experiments), and Exp3 (first experiment only). Such a result is unsurprising since none of the baselines has specific theoretical guarantees in the Autoregressive Bandit problem, even in the simple scenario when $n = 1$. Even though, we decided to adopt these algorithms as baselines since they represent the gold standard algorithms in the bandit literature (UCB1, Exp3) and the algorithms that solve problems near to ours (B-Exp3 , AR2), respectively.



(a) *Setting* A.

(b) *Setting* B.

(c) *Setting* C.

Figure 5.8: *Cumulative regret of* AR−UCB *and the others bandit baselines in the case of* $n = 1$ *(100 runs, mean ± std).*

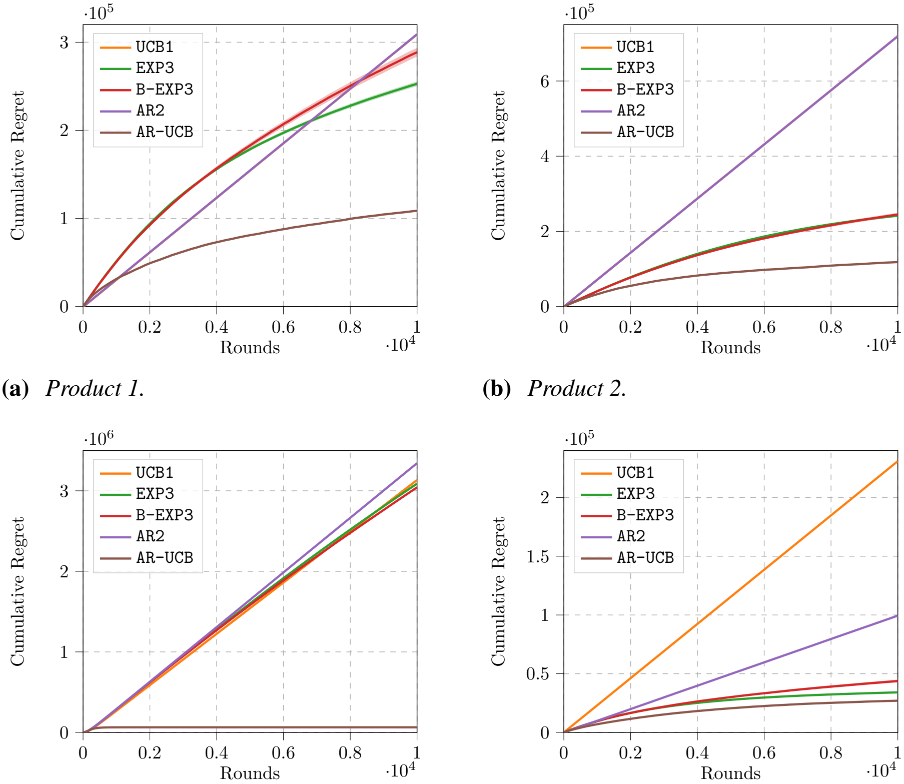
5.5.8 Real-World Data - Dynamic Pricing

We evaluate AR−UCB over the *dynamic pricing* task in e-commerce. The scenario we consider in this experiment is the one presented in Sections 5.1 and 5.2. The problem of sequentially choosing the price while dealing with exploration-exploitation dilemma is a well-known task in the literature (Kleinberg and Leighton, 2003). We show that AR−UCB is able to find the pricing schedule that maximizes the total sales while accounting for loyalty dynamics, using a simulation environment generated from real-world data.

Setting Configuration We have the possibility to access a dataset of transactions generated from a real e-commerce website selling consumables.[7] We focused on the top 4 best-selling products. For each product, we have weekly records of the number of units sold and the related price. We discretize the prices into $k = 8$ price bands (i.e., our actions) and we build the simulation environment considering a maximum horizon of the effect of the past prices on the customer of $n = 8$ weeks. The choice of $n = 8$ (i.e., two months) is ruled by business logic that is characteristic of the market in analysis. For each price band a, we estimated the parameters $\boldsymbol{\gamma}(a)$ through standard regression techniques. We used the historical sales data as the response variable predicted with the observed conversion rates for any group of customers (grouped using the number of weeks passed since their last purchase) with respect to any given price band a_t. In this experiment, we compare AR−UCB and the other bandit baselines presented in Section 5.5.1.

Results Figure 5.9 shows that only AR−UCB achieves sublinear regret for all the four products. Exp3 and B−Exp3 achieve sublinear regret for 3 out to 4 products, although their cumulative regret is always larger than

[7] We cannot share the original dataset due to an NDA.



(a) *Product 1.*

(b) *Product 2.*

(c) *Product 3.*

(d) *Product 4.*

Figure 5.9: *AR-UCB, UCB1, Exp3, B-Exp3 and AR2 in the experiment from real-world data (100 runs, mean ± std).*

that of AR-UCB, making the latter the best performing algorithm over the competitors. Lastly, both UCB1 and AR2 suffer linear regret for all the products under analysis.

5.6 Related Works

In this section, we discuss and compare the works that share similarities with the ARBs, focusing on MABs and online learning in non-linear systems.

Multi-Armed Bandits In the more classical Multi-Armed Bandit (MAB) setting, the learning problem does not involve temporal dependencies between successive rewards. The MAB setting has been studied under the assumptions of both *stochastic* and *adversarial* noise models. In the former case, `UCB1` (Lai and Robbins, 1985; Auer et al., 2002a) represents the parent algorithm. Instead, when adversarial noise is involved `Exp3` (Auer et al., 1995, 2002b) is usually employed. This algorithm has been extended by `RExp3` (Besbes et al., 2014) to handle with the *non-stationary* setting. Differently from both the adversarial and non-stochastic setting, we assume that the rewards are not preselected by an adversary or nature but, instead, they change as an effect of the actions played. Indeed, the underlying autoregressive process (affected by a stochastic noise) is such that the current action impacts the future rewards. Therefore, importing the adversarial MAB terminology, the ARBs can be reduced to an adversary setting with an *adaptive* (or non-oblivious) adversary (Dekel et al., 2012). In particular, the $\mathcal{O}(\sqrt{kT})$ regret guarantees of `Exp3` are not achievable in the ARB setting as `Exp3` competes against the best constant policy while the optimal policy for ARBs is not constant (Theorem 5.2.1). As we shall see empirically in Section 5.5, a constant policy suffers a linear regret.

Moreover, our setting presents similarities with MABs with *delayed* feedback (e.g., Pike-Burke et al., 2018). However, in ARB the effect of the actions is propagated (not exactly delayed). Markov (Ortner et al., 2012) and restless (Tekin and Liu, 2012) bandits, instead, consider underlying processes that influence the rewards. However, these processes are not supposed to be controlled by the action history. In Chen et al. (2023), the authors study the problem of learning and control in a setting that considers temporal structure in the feedback, modeled as an AR(1) autoregressive process.

Online Learning in Non-Linear Systems The ARB setting is a specific case of a non-linear dynamical system. Although the literature related to this setting is wide, no work faces all problems that the ARB setting presents, including learning to control with regret guarantees. Mania et al. (2022) focus on learning the parameters of a particular class of non-linear systems. However, the approach is limited to estimation and no control algorithm is proposed. Similarly, Umlauft and Hirche (2017) deal with learning the system parameters with stability guarantees without the chance to control it. Several recent works (Kakade et al., 2020; Lale et al., 2021) focus on the learning and control of non-linear systems with regret guarantees. However, these works make use of an oracle to solve a complex optimization problem to perform optimistic planning (i.e., optimal policy given an optimistic estimate of the system). This problem in a non-linear setting, however, is proven to be NP-hard (Sahni, 1974; Dani et al., 2008). Furthermore, the class of non-linear systems considered in these chapter does not include the ARB setting. Other works (e.g., Albalawi et al., 2021) overcome the request for the oracle by searching in the restricted space of constant policies, leading to the best equilibrium. However, this solution can be suboptimal in several cases, including ARBs (e.g., Section 5.5.2).

5.7 Discussion and Conclusions

In this chapter, we faced the online sequential decision-making problem where an autoregressive temporal structure between the observed rewards is present. First, we formally introduced the ARB setting and defined the notion of optimal policy, demonstrating that, under certain circumstances, a myopic policy is optimal also to optimize the total reward, regardless of the target time horizon, and that the optimal policy is not constant over time and depends on the most recent observed rewards. Then, we proposed an optimistic bandit algorithm, `AR-UCB`, to learn online the parameters of the underlying process for each action. We demonstrated that the presented algorithm enjoys sublinear regret, depending on the AR order n and on an index of the speed at which the system reaches a stable condition. Finally, we provided an experimental campaign to validate the proposed solution demonstrating the effectiveness of `AR-UCB` w.r.t. several bandit baselines on both synthetic and real-world scenarios, and we analyzed the behavior of `AR-UCB` when key parameters are misspecified.

Part II

Advertising

CHAPTER 6

Introduction on Advertising

Whenever we want to sell an item, after having decided the price, a topic we faced in Part I, we have to understand how we can advertise it properly. Indeed, a good pricing strategy without adequate advertising is not effective since no one know us and our products.

Machine Learning can be used to face several aspects of advertising, e.g., budget and bid optimization, ad generation, and audience discovery. In this part, we focus on the problem of budget optimization for advertising campaigns.

In order to advertise a product, we have first of all to create one or more advertising campaigns. Then, we have to select a provider that makes available advertising slots. Once we choose a provider, we have to compete with other advertisers in order to gain the advertising slot. This competition takes the form of an auction. In these auctions, we compete with all the other advertisers to get the slot, and every advertiser makes a bid to get the space. The bids can be referred to the impression, so we pay if we get the space, to click, so we pay if the user clicks on the ad, or a conversion, so we pay the amount we bid only in the case the user makes an action, e.g., buy something. These bids are then compared in terms of bid per impression, thanks to conversion coefficients specific to the case. Auctions can be first-price or second-price, so we can pay the bid we select (i.e., first-price auctions) or a few more than the second-highest bid (i.e., second-price auctions).

Nowadays, the bidding strategies are often consigned to real-time bidding platforms (Zhang et al., 2014; Yuan et al., 2014), and the advertiser focuses more on the definition on the optimal budget. Usually, the management provides an overall budget over a time period (e.g., a quarter), and we are asked to determine how to allocate such a budget over time and among the advertising campaigns we have.

Before getting into the details of budget optimization, we have to introduce the different kinds of campaigns we can be required to manage. In order to classify the campaign type, we can look at which level of the so-called *marketing funnel* (Colicev et al., 2019) such a campaign is optimizing. The marketing funnel, whose simplified representation is provided in Figure 6.1, is a structure useful to characterize the different kinds of campaigns we may face. The structure recalls the one of a funnel since, at every step, it is likely that the number of users is diminishing. On the top, we have awareness (i.e., impression) campaigns designed to let people know that a given product or brand exists. These campaigns are created to reach great amounts of users, and their scope is only to inform and no action is expected from the user, as they are studied to attract attention to their subject. Then, we have click campaigns in order to let the user be able to consider us, and create or increase the buy intent. Their scope is to arouse curiosity and generate a first active interaction from the user. Finally, we have conversion campaigns that let the user able to convert at the proper moment. In this last phase of the funnel, we can suppose that a user is already aware and interested in e.g., buy an item, due to the persuasion made from the campaigns at the higher level of the funnel.

Given this brief overview of the marketing funnel, we now have two directions to follow. First, we can suppose that we want to optimize advertising budgets over a set of campaigns of the same type (e.g., impression campaign, conversion campaign). Second, we can suppose to face different kinds of campaigns, and we want to optimize the whole process from the awareness to the purchase, so find the best *Marketing Mix Model* (MMM). The first problem is already studied several times and the literature covers almost all the possible challenges we may face. So, our attention in this part of the thesis is dedicated to the online optimization of the marketing mix model.

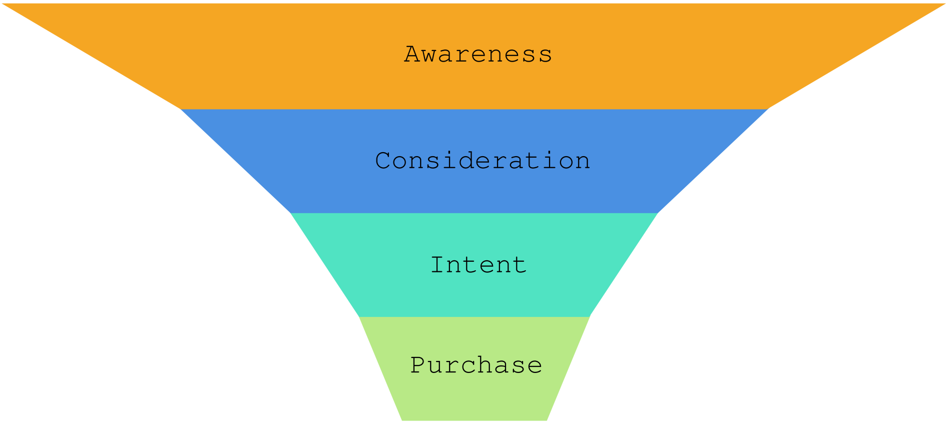


Figure 6.1: *A simplified representation of the marketing funnel.*

6.1 Foundations of Advertising Budget Optimization

In this section, we present the technical notions needed to understand budget optimization in advertising. This section is structured in two parts. First, we discuss how to model and optimize the budget for different campaigns with the same target. Then, we discuss how to optimize the marketing mix model.

6.1.1 Single Target Optimization

In the case of naive single target optimization, we have to model for each campaign the relation between an input metric, usually the budget, and a target metric, e.g., the impressions. The data related to these quantities can be collected every day/week from the advertising platforms. When we try to figure out the relation between these two quantities, we can see a distribution like the one of Figure 6.2. In this figure, we can observe how the model between these quantities can be assumed to be monotonic and concave following the economic principle of the *diminishing return* (Mesak and Means, 1998). This is because, by increasing the budget, we aim to win more auction, and, in stationary market conditions, this will require an increase in our bid, diminishing our marginal return. An example of a possible model following the monotonic non-decreasing and concave assumption is provided in Figure 6.3. Usually, the data about the target are very noisy because the allocation process (i.e., the auction) is influenced by a huge amount of factors. Given that, these assumptions can be used to simplify the model and reduce the number of samples required in order to

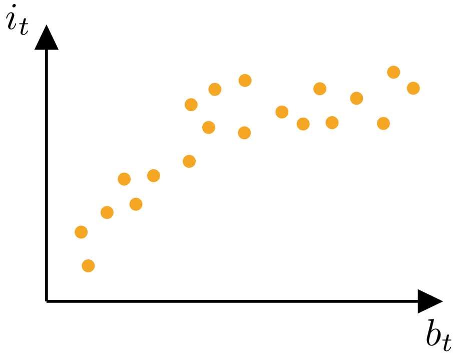


Figure 6.2: *Example of samples (b_t, i_t) drawn from the distribution binding budgets and impressions.*

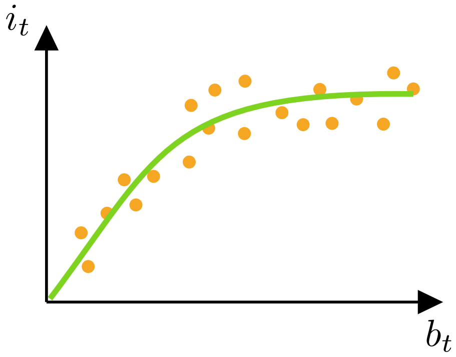


Figure 6.3: *Example of a possible model over the samples (b_t, i_t).*

learn a model with a given accuracy (a.k.a. the *sample complexity*).

Once we have modeled this relation between the budget b_t and the target i_t, we can, given, e.g., a daily budget constraint B_t over all the campaign, use an optimization procedure to find the best combination of budgets.

6.1.2 Marketing Mix Model

When we want to optimize a marketing mix model, we are required to understand how to allocate the budget over all the different types of campaigns. When we design a marketing mix model, our target is usually to generate conversions. However, to increase the conversion probability when we show an advertisement to an user, we have to let it know, e.g., about us, or our product. This implies that the awareness ads increase the probability of conversion when we display a conversion ad. The goal of a good marketing mix model is to find the best *mix* between the various

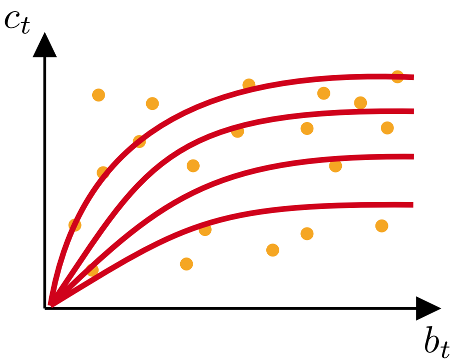


Figure 6.4: *Example of possible models over the samples* (b_t, c_t).

campaigns in order to maximize conversions, given an overall budget B_t.

The goal traduces in creating a model that, given as input the expenses for all the campaigns, generates a conversion prediction. This cannot be done as described above, searching for a map only between the budget of the conversion campaigns and the conversions generated. Indeed, if we were able to find this direct map, the marketing mix models would not be required anymore. An example of that is provided in Figure 6.4, where we can observe how the model between budget and conversion is very challenging to be defined, due to the external influences of the other campaigns that will not be visible if we do not consider the other budgets invested and the related impressions and click generated.

CHAPTER 7

Dynamical Linear Bandits for Marketing Mix Models

In this chapter, we introduce a novel setting, the Dynamical Linear Bandits (DLB), an extension of the linear bandits characterized by a hidden state. When an action is performed, the learner observes a noisy reward whose mean is a linear function of the hidden state and of the action. Then, the hidden state evolves according to linear dynamics, affected by the performed action too. This setting theoretically formalizes the problem of learning online in *Marketing Mix Models*. We start by introducing the setting, discussing the notion of optimal policy, and deriving an expected regret lower bound. Then, we provide an optimistic regret minimization algorithm, Dynamical Linear Upper Confidence Bound (`DynLin-UCB`), that suffers an expected regret of order $\widetilde{\mathcal{O}}\left(\frac{d\sqrt{T}}{(1-\overline{\rho})^{3/2}}\right)$, where $\overline{\rho}$ is a measure of the stability of the system, and d is the dimension of the action vector.

This chapter presents (Mussi et al., 2023a), a joint work with Alberto Maria Metelli and Marcello Restelli, published at the *International Conference on Machine Learning (ICML)*. A preliminary version of this work (Mussi et al., 2022b) appeared at the *Complex Feedback in Online Learning Workshop*.

7.1 Introduction

In a large variety of sequential decision-making problems, a learner must choose an action that, when executed, determines an evolution of the underlying system state that is hidden to the learner. In these partially observable problems, the learner observes a reward (i.e., feedback) representing the combined effect of multiple actions played in the past.

For instance, in online advertising campaigns, the process that leads to a *conversion*, i.e., the *marketing funnel* (Court et al., 2009), is characterized by complex dynamics and comprises several phases, as discussed in Chapter 6. When heterogeneous campaigns/platforms are involved, a profitable budget investment policy has to account for the interplay between campaigns/platforms. In this scenario, a conversion (e.g., a user's purchase of a promoted product) should be attributed not only to the latest ad the user was exposed to, but also to previous ones (Berman, 2018). The *joint* consideration of each funnel phase is a fundamental step towards an optimal investment solution while considering the advertising campaigns/platforms *independently* leads to sub-optimal solutions. Consider, for instance, a simplified version of the funnel with two types of campaigns: *awareness* (i.e., impression) ads and *conversion* ads. The first kind of ad aims at improving brand awareness, while the latter aims at creating the actual conversion. If we evaluate the performances in terms of conversions only, we will discover that impression ads are not instantaneously effective in creating conversions, so we will be tempted to reduce the budget invested in such a campaign. However, this approach is sub-optimal because impression ads increase the chance to convert when a conversion ad is shown after the impression (e.g., Hoban and Bucklin, 2015). In addition, the effect of some ads, especially impression ads delivered via television, may be delayed. It has been demonstrated (Chapelle, 2014) that users remember advertising over time in a vanishing way, leading to consequences that non-dynamical models cannot capture. This kind of interplay comprises more general scenarios than the simple reward delay, including the case where the interaction is governed by a dynamics *hidden* to the observer.

While this scenario can be indubitably modeled as a Partially Observable Markov Decision Process (POMDP, Åström, 1965), the complexity of the framework and its generality are often not required to capture the main features of the problem. Indeed, for specific classes of problems, the Multi-Armed Bandit (MAB, Lattimore and Szepesvári, 2020) literature has explored the possibility of experiencing delayed reward either assuming that the actual reward will be observed, individually, in the future (e.g., Joulani et al., 2013) or with the more realistic assumption that an aggregated feedback is available (e.g., Pike-Burke et al., 2018), with also specific applications to online advertising (Vernade et al., 2017). Although effective in dealing with delay effects and the possibility of a reward spread in the future (Cesa-Bianchi et al., 2018), they do not account for the additional, more complex, dynamical effects, which can be regarded as the evolution of a hidden state.

In this chapter, we take a different perspective. We propose to model the non-observable dynamical effects underlying the phenomena as a Linear Time-Invariant (LTI) system (Hespanha, 2018). In particular, the system is characterized by a hidden internal state $\mathbf{x}_t$ (in our case, the awareness) which evolves via linear dynamics fed by the action $\mathbf{u}_t$ (in our case, the amount invested) and affected by noise. At each round, the learner experiences a reward y_t (in our case, the conversions), which is a noisy observation that linearly combines the state $\mathbf{x}_t$ and the action $\mathbf{u}_t$. Our goal consists in learning an optimal policy so as to maximize the expected cumulative reward. We call this setting *Dynamical Linear Bandits* (DLBs) that, as we shall see, reduces to linear bandits (Abbasi-Yadkori et al., 2011) when no dynamics are involved. Because of the dynamics, the effect of each action persists over time indefinitely but, under stability conditions, it vanishes asymptotically. This allows representing interference and synergy between platforms, thanks to the dynamic nature of the system.

Contributions In Section 7.2, we introduce the Dynamical Linear Bandit (DLB) setting to represent sequential decision-making problems characterized by a hidden state that evolves linearly according to an *unknown* dynamics. We show that, under stability conditions, the optimal policy corresponds to playing the *constant action* that leads the system to the most profitable steady state. Then, we derive an expected regret lower bound of order $\Omega\left(\frac{d\sqrt{T}}{(1-\overline{\rho})^{1/2}}\right)$, being d the dimensionality of the action space and $\overline{\rho} < 1$ the spectral radius of the dynamical matrix of the system evolution law.[1] In Section 7.3, we propose a novel optimistic regret minimization algorithm, *Dynamical Linear Upper Confidence Bound* (`DynLin-UCB`), for the DLB setting. `DynLin-UCB` takes inspiration from `Lin-UCB` but subdivides the optimization horizon T into increasing-length epochs. In each epoch, an action is selected optimistically and kept constant (i.e., persisted) so that the system approximately reaches the steady state. We provide a regret analysis for `DynLin-UCB` showing that, under certain assumptions, it

[1] The smaller $\overline{\rho}$, the faster the system reaches its steady state.

enjoys $\widetilde{\mathcal{O}}\left(\frac{d\sqrt{T}}{(1-\overline{\rho})^{3/2}}\right)$ expected regret. In Section 7.5, we provide a numerical validation, with both synthetic and real-world data, compared with bandit baselines. The proofs of all the results are reported in Appendix C.

7.2 Setting

In this section, we introduce the *Dynamical Linear Bandits* (DLBs), the learner-environment interaction, assumptions, and regret (Section 7.2.1). Then, we derive a closed-form expression for the optimal policy for DLBs (Section 7.2.2). Finally, we derive a lower bound to the regret, highlighting the intrinsic complexities of the DLB setting (Section 7.2.3).

7.2.1 Problem Formulation

In a Dynamical Linear Bandit (DLB), the environment is characterized by a *hidden* state, i.e., a n-dimensional real vector, initialized to $\mathbf{x}_1 \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space. At each round $t \in \mathbb{N}$, the environment is in the hidden state $\mathbf{x}_t \in \mathcal{X}$, the learner chooses an action, i.e., a d-dimensional real vector $\mathbf{u}_t \in \mathcal{U}$, where $\mathcal{U} \subseteq \mathbb{R}^d$ is the action space. Then, the learner receives a noisy reward $y_t = \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle + \langle \boldsymbol{\theta}, \mathbf{u}_t \rangle + \eta_t \in \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}$ is the reward space, $\boldsymbol{\omega} \in \mathbb{R}^n$, $\boldsymbol{\theta} \in \mathbb{R}^d$ are unknown parameters, and η_t is a zero-mean σ^2–subgaussian random noise, conditioned to the past. Then, the environment evolves to the new state according to the unknown linear dynamics $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the dynamic matrix, $\mathbf{B} \in \mathbb{R}^{n \times d}$ is the action-state matrix, and $\boldsymbol{\epsilon}_t$ is a zero-mean σ^2–subgaussian random noise, conditioned to the past, independent of η_t.[2][3]

Remark 7.2.1. *The setting proposed above is a particular case of POMDP (Åström, 1965), in which the state $\mathbf{x}_t$ is non-observable, while the learner accesses the noisy observation y_t that corresponds to the noisy reward too. Furthermore, the setting can be viewed as a MISO (Multiple Input Single Output) discrete-time LTI system (Kalman, 1963). Finally, the DLB reduces to (non-contextual) linear bandit (Abbasi-Yadkori et al., 2011) when the hidden state does not affect the reward, i.e., when $\boldsymbol{\omega} = \mathbf{0}$.*

DLBs in MMM Scenario *The problem of optimal budget allocation in the marketing mix model can be seen as a DLB problem. In this setting,*

[2] A zero-mean random vector $\mathbf{x} \in \mathbb{R}^n$ is σ^2-subgaussian, in the sense of Hsu et al. (2012), if for every vector $\boldsymbol{\zeta} \in \mathbb{R}^n$ it holds that $\mathbb{E}\left[\exp\left(\langle \boldsymbol{\zeta}, \mathbf{x} \rangle\right)\right] \leqslant \exp(\|\boldsymbol{\zeta}\|_2^2 \sigma^2/2)$.

[3] n is the *order* of the LTI system (Kalman, 1963). We make no assumption on the value of n and on its knowledge.

the budget we can set for the different campaigns are our action $\mathbf{u}_t$. The value of awareness, which is not measurable, is the hidden state $\mathbf{x}_t$. The reward y_t is the number of conversion, that we can observe. The saturation effects discussed in Chapter 6 over the single campaigns (Figure 6.3) can be considered in DLBs by limiting the action space $\mathcal{U}$ when we know there is a saturation effect.

Markov Parameters We revise a useful representation, that for every $H \in [\![t]\!]$ allows expressing y_t in terms of the sequence of the most recent $H+1$ actions $(\mathbf{u}_s)_{s \in [\![t-H,t]\!]}$, reward noise η_t, H state noises $(\boldsymbol{\epsilon}_s)_{s \in [\![t-H,t-1]\!]}$, and starting state $\mathbf{x}_{t-H}$ (Ho and Kalman, 1966; Oymak and Ozay, 2019; Tsiamis and Pappas, 2019; Sarkar et al., 2021):

$$y_t = \underbrace{\sum_{s=0}^{H} \langle \mathbf{h}^{\{s\}}, \mathbf{u}_{t-s} \rangle}_{\text{action effect}} + \underbrace{\boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H} \mathbf{x}_{t-H}}_{\text{starting state}} + \underbrace{\eta_t + \sum_{s=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \boldsymbol{\epsilon}_{t-s}}_{\text{noise}}, \tag{7.1}$$

where the sequence of vectors $\mathbf{h}^{\{s\}} \in \mathbb{R}^d$ for every $s \in \mathbb{N}$ are called *Markov parameters* and are defined as: $\mathbf{h}^{\{0\}} = \boldsymbol{\theta}$ and $\mathbf{h}^{\{s\}} = \mathbf{B}^{\mathrm{T}}(\mathbf{A}^{s-1})^{\mathrm{T}}\boldsymbol{\omega}$ if $s \geqslant 1$. Furthermore, we introduce the *cumulative Markov parameters*, defined for every $s, s' \in \mathbb{N}$ with $s \leqslant s'$ as $\mathbf{h}^{[\![s,s']\!]} = \sum_{l=s}^{s'} \mathbf{h}^{\{l\}}$ and the corresponding limit as $s' \to +\infty$, i.e., $\mathbf{h}^{[\![s,+\infty)\!)} = \sum_{l=s}^{+\infty} \mathbf{h}^{\{l\}}$. Finally, we use the abbreviation $\mathbf{h} = \mathbf{h}^{[\![0,+\infty)\!)} = \boldsymbol{\theta} + \mathbf{B}^{\mathrm{T}}(\mathbf{I}_n - \mathbf{A})^{-\mathrm{T}}\boldsymbol{\omega}$.

We will make use of the following standard assumption related to the *stability* of the dynamic matrix $\mathbf{A}$, widely employed in discrete–time LTI literature (Oymak and Ozay, 2019; Lale et al., 2020a,b).

Assumption 7.1 (Stability). *The spectral radius of $\mathbf{A}$ is strictly smaller than 1, i.e., $\rho(\mathbf{A}) < 1$, and the maximum spectral norm to spectral radius ratio of the powers of $\mathbf{A}$ is bounded, i.e., $\Phi(\mathbf{A}) < +\infty$.*[4]

Policies and Performance The learner's behavior is modeled via a deterministic *policy* $\underline{\boldsymbol{\pi}} = (\boldsymbol{\pi}_t)_{t \in \mathbb{N}}$ defined, for every round $t \in \mathbb{N}$, as $\boldsymbol{\pi}_t : \mathcal{H}_{t-1} \to \mathcal{U}$, mapping the history of observations $H_{t-1} = (\mathbf{u}_1, y_1, \ldots, \mathbf{u}_{t-1}, y_{t-1}) \in \mathcal{H}_{t-1}$ to an action $\mathbf{u}_t = \boldsymbol{\pi}_t(H_{t-1}) \in \mathcal{U}$, where $\mathcal{H}_{t-1} = (\mathcal{U} \times \mathcal{Y})^{t-1}$ is the set of histories of length $t-1$. The performance of a policy $\underline{\boldsymbol{\pi}}$ is evaluated in terms of the *(infinite-horizon) expected average reward*:

$$J(\underline{\boldsymbol{\pi}}) := \liminf_{H \to +\infty} \mathbb{E}\left[\frac{1}{H}\sum_{t=1}^{H} y_t\right], \tag{7.2}$$

[4] The latter is a mild assumption: if $\mathbf{A}$ is diagonalizable as $\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{-1}$, then $\Phi(\mathbf{A}) \leqslant \|\mathbf{Q}\|_2\|\mathbf{Q}^{-1}\|_2$ and it is finite. In particular, if $\mathbf{A}$ is symmetric then $\Phi(\mathbf{A}) = 1$.

$$\text{where} \qquad \begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t \\ y_t = \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle + \langle \boldsymbol{\theta}, \mathbf{u}_t \rangle + \eta_t \\ \mathbf{u}_t = \boldsymbol{\pi}_t(H_{t-1}) \end{cases}, \quad \forall t \in \mathbb{N},$$

where the expectation is taken w.r.t. the randomness of the state noise $\boldsymbol{\epsilon}_t$ and reward noise η_t. If a policy $\underline{\boldsymbol{\pi}}$ is *constant*, i.e., $\boldsymbol{\pi}_t(H_{t-1}) = \mathbf{u}$ for every $t \in \mathbb{N}$, we abbreviate $J(\mathbf{u}) = J(\underline{\boldsymbol{\pi}})$. A policy $\underline{\boldsymbol{\pi}}^*$ is an *optimal policy* if it maximizes the expected average reward, i.e., $\underline{\boldsymbol{\pi}}^* \in \arg\max_{\underline{\boldsymbol{\pi}}} J(\underline{\boldsymbol{\pi}})$, and its performance is denoted by $J^* := J(\underline{\boldsymbol{\pi}}^*)$.

We further introduce the following assumption that requires the boundedness of the norms of the relevant quantities.

Assumption 7.2 (Boundedness)**.** *There exist* $\Theta, \Omega, B, U < +\infty$ *such that* $\|\boldsymbol{\theta}\|_2 \leqslant \Theta$, $\|\boldsymbol{\omega}\|_2 \leqslant \Omega$, $\|\mathbf{B}\|_2 \leqslant B$, $\sup_{\mathbf{u}\in\mathcal{U}} \|\mathbf{u}\|_2 \leqslant U$, *and* $\sup_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_2 \leqslant X$, $\sup_{\mathbf{u}\in\mathcal{U}} |J(\mathbf{u})| \leqslant 1$.[5]

Regret The *regret* suffered by playing a policy $\underline{\boldsymbol{\pi}}$, competing against the optimal infinite-horizon policy $\underline{\boldsymbol{\pi}}^*$ over a *learning horizon* $T \in \mathbb{N}$ is given by:

$$R(\underline{\boldsymbol{\pi}}, T) := TJ^* - \sum_{t=1}^{T} y_t, \tag{7.3}$$

where y_t is the sequence of rewards collected by playing $\underline{\boldsymbol{\pi}}$ as in Equation (7.2). The goal of the learner consists in minimizing the *expected regret* $\mathbb{E}\, R(\underline{\boldsymbol{\pi}}, T)$, where the expectation is taken w.r.t. the randomness of the reward.

7.2.2 Optimal Policy

In this section, we derive a closed-form expression for the optimal policy $\underline{\boldsymbol{\pi}}^*$ for the infinite–horizon objective function, as introduced in Equation (7.2).

Theorem 7.2.1 (Optimal Policy)**.** *Under Assumptions 7.1 and 7.2, an optimal policy* $\underline{\boldsymbol{\pi}}^*$ *maximizing the (infinite-horizon) expected average reward* $J(\underline{\boldsymbol{\pi}})$ *(Equation 7.2), for every round* $t \in \mathbb{N}$ *and history* $H_{t-1} \in \mathcal{H}_{t-1}$ *is given by:*

$$\boldsymbol{\pi}_t^*(H_{t-1}) = \mathbf{u}^* \qquad \textit{where} \qquad \mathbf{u}^* \in \arg\max_{\mathbf{u}\in\mathcal{U}} J(\mathbf{u}) = \langle \mathbf{h}, \mathbf{u} \rangle. \tag{7.4}$$

[5] The assumption of the bounded state norm $\|\mathbf{x}\|_2 \leqslant X$ holds whenever the state noise $\boldsymbol{\epsilon}$ is bounded. As shown by Agarwal et al. (2019), this assumption can be relaxed, for unbounded subgaussian noise, by conditioning to the event that none of the noise vectors are ever large at the cost of an additional $\log T$ factor in the regret.

Some remarks are in order. The optimal policy plays the *constant* action $\mathbf{u}^* \in \mathcal{U}$ which brings the system in the "most profitable" steady-state.[6] Indeed, the expression $\langle \mathbf{h}, \mathbf{u} \rangle$ can be rewritten expanding the cumulative Markov parameter as $(\boldsymbol{\theta}^{\mathrm{T}} + \boldsymbol{\omega}^{\mathrm{T}}(\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B})\mathbf{u}^*$ and $\overline{\mathbf{x}}^* = (\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}\mathbf{u}^*$ is the expression of the steady state $\overline{\mathbf{x}}^* = \mathbf{A}\overline{\mathbf{x}}^* + \mathbf{B}\mathbf{u}^*$, when applying action $\mathbf{u}^*$. It is worth noting the role of Assumption 7.1 which guarantees the existence of the inverse $(\mathbf{I}_n - \mathbf{A})^{-1}$. In this sense, our problem shares the constant nature of the optimal policy with the linear bandit setting (Abbasi-Yadkori et al., 2011), although ours is characterized by an evolving state, which introduces a new trade-off in the action selection. From the LTI system perspective, this implies that we can restrict to *open-loop stationary* policies. The reason why DLBs do not benefit from *closed-loop* policies, differently from other classical problems, such as the LQG (Abbasi-Yadkori and Szepesvári, 2011), lies in the linearity of the reward y_t and in the additive noise η_t and $\boldsymbol{\epsilon}_t$, making their presence irrelevant (in expectation) for control purposes. Nonetheless, as we shall see, our problem poses additional challenges compared to linear bandits since, in order to assess the quality of an action $\mathbf{u} \in \mathcal{U}$, instantaneous rewards are not reliable, and we need to let the system evolve to the steady state and, only then, observe the reward.

7.2.3 Regret Lower Bound

In this section, we provide a lower bound to the expected regret that any learning algorithm suffers when addressing the learning problem in a DLB.

Theorem 7.2.2 (Lower Bound)**.** *For any policy $\underline{\boldsymbol{\pi}}$ (even stochastic), there exists a DLB fulfilling Assumptions 7.1 and 7.2, such that for sufficiently large $T \geqslant \mathcal{O}\left(\frac{d^2}{1-\rho(\mathbf{A})}\right)$, policy $\underline{\boldsymbol{\pi}}$ suffers an expected regret lower bounded by:*

$$\mathbb{E}R(\underline{\boldsymbol{\pi}}, T) \geqslant \Omega\left(\frac{d\sqrt{T}}{(1-\rho(\mathbf{A}))^{\frac{1}{2}}}\right).$$

The lower bound highlights the main challenges of the DLB learning problem. First of all, we observe a dependence on $1/(1-\rho(\mathbf{A}))$, being $\rho(\mathbf{A})$ the spectral radius of the matrix $\mathbf{A}$. This is in line with the intuition that, as $\rho(\mathbf{A})$ approaches 1, the problem becomes more challenging. Furthermore, we note that when $\rho(\mathbf{A}) = 0$, i.e., the problem has no dynamical effects, the lower bound matches the one of linear bandits (Lattimore and

[6] In Appendix C.1, we show that the optimal policy is non–stationary for the finite–horizon case.

Szepesvári, 2020). It is worth noting that, for technical reasons, the result of Theorem 7.2.2 is derived under the assumption that, at every round $t \in [\![T]\!]$, the agent observes *both* the state $\mathbf{x}_t$ and the reward y_t (see Appendix C). Clearly, this represents a simpler setting w.r.t. DLBs (in which $\mathbf{x}_t$ is hidden) and, consequently, Theorem 7.2.2 is a viable lower bound for DLBs too.

7.3 Algorithm

In this section, we present an *optimistic* regret minimization algorithm for the Dynamical Linear Bandits setting. *Dynamical Linear Upper Confidence Bound* (`DynLin-UCB`), whose pseudocode is reported in Algorithm 7.1, requires the knowledge of an upper-bound $\overline{\rho} < 1$ on the spectral radius of the dynamic matrix $\mathbf{A}$ (i.e., $\rho(\mathbf{A}) \leqslant \overline{\rho}$) and on the maximum spectral norm to spectral radius ratio $\overline{\Phi} < +\infty$ (i.e., $\Phi(\mathbf{A}) \leqslant \overline{\Phi}$), as well as the bounds on the relevant quantities of Assumption 7.2.[7] `DynLin-UCB` is based on the following simple observation. To assess the quality of action $\mathbf{u} \in \mathcal{U}$, we need to *persist* in applying it so that the system approximately reaches the corresponding steady state and, then, observe the reward y_t, representing a reliable estimate of $J(\mathbf{u}) = \langle \mathbf{h}, \mathbf{u} \rangle$. We shall show that, under Assumption 7.1, the number of rounds needed to approximately reach such a steady state is logarithmic in the learning horizon T and depends on the upper bound of the spectral norm $\overline{\rho}$. After initializing the Gram matrix $\mathbf{V}_0 = \lambda \mathbf{I}_d$ and the vectors $\mathbf{b}_0$ and $\widehat{\mathbf{h}}_0$ both to $\mathbf{0}_d$ (line 1), `DynLin-UCB` subdivides the learning horizon T in $M \leqslant T$ *epochs*. Each epoch $m \in [\![M]\!]$ is composed of $H_m + 1$ rounds, where $H_m = \lfloor \log m / \log(1/\overline{\rho}) \rfloor$ is logarithmic in the epoch index m. At the beginning of each epoch, $m \in [\![M]\!]$, `DynLin-UCB` computes the upper confidence bound (UCB) index (line 4) defined for every $\mathbf{u} \in \mathcal{U}$ as:

$$\mathrm{UCB}_t(\mathbf{u}) := \langle \widehat{\mathbf{h}}_{t-1}, \mathbf{u} \rangle + \beta_{t-1} \|\mathbf{u}\|_{\mathbf{V}_{t-1}^{-1}}, \tag{7.5}$$

where $\widehat{\mathbf{h}}_{t-1} = \mathbf{V}_{t-1}^{-1} \mathbf{b}_{t-1}$ is the Ridge regression estimator of the cumulative Markov parameter $\mathbf{h}$, as in Equation (7.4) and $\beta_{t-1} \geqslant 0$ is an exploration coefficient to be defined later. Similar to `Lin-UCB` (Abbasi-Yadkori et al., 2011), the index $\mathrm{UCB}_t(\mathbf{u})$ is designed to be optimistic, i.e.,

[7] As an alternative, one can consider a more demanding requirement of the knowledge of a bound on the spectral norm $\|\mathbf{A}\|_2$ of $\mathbf{A}$. Similar assumptions regarding the knowledge of analogous quantities are considered in the literature, e.g., *decay of Markov operator norms* (Simchowitz et al., 2020) and *strong stability* (Plevrakis and Hazan, 2020), spectral norm bound (Lale et al., 2020a). As a side note, the knowledge of $\overline{\rho} \geqslant \rho(\mathbf{A})$ (or an equivalent quantity) is proved to be unavoidable by Theorem 7.2.2. Indeed, if no restriction on $\rho(\mathbf{A})$ is enforced (i.e., just $\rho(\mathbf{A}) < 1$), one can always consider the DLB in which $\rho(\mathbf{A}) = 1 - 1/T < 1$ making the regret lower bound degenerate to linear.

Algorithm 7.1: `DynLin-UCB`.

```
   Input: Regularization parameter $\lambda > 0$, exploration coefficients $(\beta_{t-1})_{t \in [\![T]\!]}$,
          spectral radius upper bound $0 \leqslant \overline{\rho} < 1$
1  Initialize $t \leftarrow 1$, $\mathbf{V}_0 = \lambda \mathbf{I}_d$, $\mathbf{b}_0 = \mathbf{0}_d$, $\widehat{\mathbf{h}}_0 = \mathbf{0}_d$,
2  Define $M = \min\{M' \in \mathbb{N} : \sum_{m=1}^{M'} 1 + \lfloor \frac{\log m}{\log(1/\overline{\rho})} \rfloor > T\} - 1$
3  for $m \in [\![M]\!]$ do
4      Compute $\mathbf{u}_t \in \arg\max_{\mathbf{u} \in \mathcal{U}} \mathrm{UCB}_t(\mathbf{u})$
5          where $\mathrm{UCB}_t(\mathbf{u}) := \langle \widehat{\mathbf{h}}_{t-1}, \mathbf{u} \rangle + \beta_{t-1} \|\mathbf{u}\|_{\mathbf{V}_{t-1}^{-1}}$
6      Play arm $\mathbf{u}_t$ and observe $y_t$
7      Define $H_m = \lfloor \frac{\log m}{\log(1/\overline{\rho})} \rfloor$
8      for $j \in [\![H_m]\!]$ do
9          Update $\mathbf{V}_t = \mathbf{V}_{t-1}$, $\mathbf{b}_t = \mathbf{b}_{t-1}$
10         $t \leftarrow t + 1$
11         Play arm $\mathbf{u}_t = \mathbf{u}_{t-1}$ and observe $y_t$
12     end
13     Update $\mathbf{V}_t = \mathbf{V}_{t-1} + \mathbf{u}_t \mathbf{u}_t^{\mathrm{T}}$, $\mathbf{b}_t = \mathbf{b}_{t-1} + \mathbf{u}_t y_t$
14     Compute $\widehat{\mathbf{h}}_t = \mathbf{V}_t^{-1} \mathbf{b}_t$
15     $t \leftarrow t + 1$
16 end
```

$J(\mathbf{u}) \leqslant \mathrm{UCB}_t(\mathbf{u})$ in high-probability for all $\mathbf{u} \in \mathcal{U}$. Then, the optimistic action $\mathbf{u}_t \in \arg\max_{\mathbf{u} \in \mathcal{U}} \mathrm{UCB}_t(\mathbf{u})$ is executed (line 6) and persisted for the next H_m rounds (lines 8-11). The length of the epoch H_m is selected such that, under Assumption 7.1, the system has approximately reached the steady state after $H_m + 1$ rounds. In this way, at the end of epoch m, the reward y_t is an almost-unbiased sample of the steady-state performance $J(\mathbf{u}_t)$. This sample is employed to update the Gram matrix estimate $\mathbf{V}_t$ and the vector $\mathbf{b}_t$ (line 13), while the samples collected in the previous H_m rounds are discarded (line 9). It is worth noting that by setting $H_m = 0$ for all $m \in [\![M]\!]$, `DynLin-UCB` reduces to `Lin-UCB`. The following sections provide the concentration of the estimator $\widehat{\mathbf{h}}_{t-1}$ of $\mathbf{h}$ (Section 7.3.1) and the regret analysis of `DynLin-UCB` (Section 7.3.2).

7.3.1 Self-Normalized Concentration Inequality for the Cumulative Markov Parameter

In this section, we provide a self-normalized concentration result for the estimate $\widehat{\mathbf{h}}_t$ of the cumulative Markov parameter $\mathbf{h}$. For every epoch $m \in [\![M]\!]$, we denote with t_m the last round of epoch m: $t_0 = 0$ and $t_m = t_{m-1} + 1 + H_m$. At the end of each epoch m, we solve the Ridge regression problem, defined for every round $t \in [\![T]\!]$ as:

$$\widehat{\mathbf{h}}_t = \operatorname*{arg\,min}_{\widetilde{\mathbf{h}} \in \mathbb{R}^d} \sum_{l \in [\![M]\!] : t_l \leqslant t_m} (y_{t_l} - \langle \widetilde{\mathbf{h}}, \mathbf{u}_{t_l} \rangle)^2 + \lambda \left\| \widetilde{\mathbf{h}} \right\|_2^2 = \mathbf{V}_t^{-1} \mathbf{b}_t.$$

We now present the following self-normalized maximal concentration inequality and, then, we compare it with the existing results in the literature.

Theorem 7.3.1 (Self-Normalized Concentration). *Let $(\widehat{\mathbf{h}}_t)_{t \in \mathbb{N}}$ be the sequence of solutions of the Ridge regression problems of Algorithm 7.1. Then, under Assumption 7.1 and 7.2, for every $\lambda \geqslant 0$ and $\delta \in (0,1)$, with probability at least $1-\delta$, simultaneously for all rounds t $\in \mathbb{N}$, it holds that:*

$$\left\| \widehat{\mathbf{h}}_t - \mathbf{h} \right\|_{\mathbf{V}_t} \leqslant \frac{c_1}{\sqrt{\lambda}} \log(e(t+1)) + c_2 \sqrt{\lambda} + \sqrt{2\widetilde{\sigma}^2 \left(\log\left(\frac{1}{\delta}\right) + \frac{1}{2} \log\left(\frac{\det(\mathbf{V}_t)}{\lambda^d} \right) \right)},$$

where:

$$c_1 = U \Omega \Phi(\mathbf{A}) \left(\frac{UB}{1-\rho(\mathbf{A})} + X \right),$$
$$c_2 = \Theta + \frac{\Omega B \Phi(\mathbf{A})}{1-\rho(\mathbf{A})},$$
$$\widetilde{\sigma}^2 = \sigma^2 \left(1 + \frac{\Omega^2 \Phi(\mathbf{A})^2}{1-\rho(\mathbf{A})^2} \right).$$

First, we note that when $\Omega = 0$ ($\boldsymbol{\omega} = \mathbf{0}_n$), i.e., the state does not affect the reward, the bound perfectly reduces to the self-normalized concentration used in linear bandits (Abbasi-Yadkori et al., 2011, Theorem 1). In particular, we recognize the second term due to the regularization parameter $\lambda > 0$ and the third one, which involves the subgaussianity parameter $\widetilde{\sigma}^2$, related to the joint contribution of the state and reward noises. Furthermore, the first term is an additional bias that derives from the epochs of length $H_m + 1$. The choice of the value H_m represents one of the main technical novelties that, on the one hand, leads to a bias that conveniently grows logarithmically with t and, on the other hand, can be computed without the knowledge of T.

It is worth looking at our result from the perspective of learning the LTI system parameters. We can compare our Theorem 7.3.1 with the concentration presented in (Lale et al., 2020a, Appendix C), which represents, to the best of our knowledge, the only result for the closed-loop identification of LTI systems with non-observable states. First, note that, although we focus on a MISO system (y_t is a scalar, being our reward), extending our estimator to multiple-outputs (MIMO) is straightforward. Second, the approach of (Lale et al., 2020a) employs the *predictive form* of the LTI system to cope with the correlation introduced by closed-loop control. This choice allows for convenient analysis of the estimated Markov parameters of the predictive form. However, recovering the parameters of the original system requires an application of the Ho-Kalman method (Ho and Kalman, 1966) which, unfortunately, does not preserve the concentration properties in general, but only for *persistently exciting* actions. Our method, instead, forces to play an open-loop policy within a single epoch (each with logarithmic duration), while the overall behavior is closed-loop, as the next action depends on the previous-epoch estimates. In this way, we are able to provide a concentration guarantee on the parameters of the original system without assuming additional properties on the action signal.

7.3.2 Regret Analysis

In this section, we provide the analysis of the regret of `DynLin-UCB`, when we select the exploration coefficient β_t based on the knowledge of the upper bounds $\overline{\rho} < 1$, $\overline{\Phi} < +\infty$, and those specified in Assumption 7.2, defined for every round $t \in [\![T]\!]$ as:

$$\beta_t := \frac{\overline{c}_1}{\sqrt{\lambda}} \log(e(t+1)) + \overline{c}_2\sqrt{\lambda} + \sqrt{2\overline{\sigma}^2 \left(\log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1 + \frac{tU^2}{d\lambda}\right)\right)}, \tag{7.6}$$

where $\overline{c}_1 = U\Omega\overline{\Phi}\left(\frac{UB}{1-\overline{\rho}} + X\right)$, $\overline{c}_2 = \Theta + \frac{\Omega B\overline{\Phi}}{1-\overline{\rho}}$, and $\overline{\sigma}^2 = \sigma^2\left(1 + \frac{\Omega^2\overline{\Phi}^2}{1-\overline{\rho}^2}\right)$. The following result provides the bound on the expected cumulative regret of `DynLin-UCB`.

Theorem 7.3.2 (Upper Bound)**.** *Under Assumptions 7.1 and 7.2, selecting β_t as in Equation* (7.6) *and $\delta = 1/T$,* `DynLin-UCB` *suffers an expected regret bounded as (highlighting the dependencies on T, $\overline{\rho}$, d, and σ only):*

$$\mathbb{E}[R(\underline{\boldsymbol{\pi}}^{\texttt{DynLin-UCB}}, T)] \leqslant \mathcal{O}\left(\frac{d\sigma\sqrt{T}(\log T)^{\frac{3}{2}}}{1-\overline{\rho}} + \frac{\sqrt{dT}(\log T)^2}{(1-\overline{\rho})^{\frac{3}{2}}} + \frac{1}{(1-\rho(\mathbf{A}))^2} \right).$$

Proof Sketch. The analysis of `DynLin-UCB` poses additional challenges compared to that of `Lin-UCB` Abbasi-Yadkori et al. (2011) because of the dynamic effects of the hidden state. The idea behind the proof is to first derive a bound on a different notion of regret, i.e., the *offline regret*: $R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) = TJ^* - \sum_{t=1}^{T} J(\mathbf{u}_t)$, that compares J^* with the steady-state performance $J(\mathbf{u}_t)$ of the action $\mathbf{u}_t = \boldsymbol{\pi}_t(H_{t-1})$ (Theorem C.0.2). This analysis of $R^{\text{off}}(\underline{\boldsymbol{\pi}}, T)$ can be comfortably carried out, by adopting a proof strategy similar to that of `Lin-UCB`. However, when applying action $\mathbf{u}_t$, the DLB does not immediately reach the performance $J(\mathbf{u}_t)$ as the expected reward $\mathbb{E}[y_t]$ experiences a transitional phase before converging to the steady state. Under stability (Assumption 7.1), it is possible to show that the expected offline regret and the expected regret differ by a constant: $|\mathbb{E}\, R(\underline{\boldsymbol{\pi}}, T) - \mathbb{E}\, R^{\text{off}}(\underline{\boldsymbol{\pi}}, T)| \leqslant \mathcal{O}(1/(1-\rho(\mathbf{A}))^2)$ (Lemma C.0.1). □

Some observations are in order. We first note a dependence on the term $1/(1-\overline{\rho})$, which, in turn, depends on the upper bound $\overline{\rho}$ of the spectral gap $\rho(\mathbf{A})$. If the system does not display a dynamics, i.e., we can set $\overline{\rho} = 0$, we obtain a regret bound that, apart from logarithmic terms, coincides with that of `Lin-UCB`, i.e., $\widetilde{\mathcal{O}}(d\sigma\sqrt{T})$. Instead, for slow-converging systems, i.e., $\overline{\rho} \approx 1$, the regret bound enlarges, as expected. Clearly, a value of $\overline{\rho}$ too large compared to the optimization horizon T (e.g., $\overline{\rho} = 1 - 1/T^{1/3}$) makes the regret bound degenerate to linear. This is a case in which the underlying system is so slow that the whole horizon T is insufficient to approximately reach the steady state. Third, the regret bound is the sum of three components: the first one depends on the subgaussian proxy σ and is due to the noisy estimation of the relevant quantities; the second one is a bias due to the epoch-based structure of `DynLin-UCB`; finally, the third one is constant (does not depend on T) accounts for the time needed to reach the steady state.

Remark 7.3.1 (Regret upper bound (Theorem 7.3.2) and lower bound (Theorem 7.2.2) Comparison)**.** *Apart from logarithmic terms, we notice a tight dependence on* d *and on* T*. Instead, concerning the spectral properties of* $\mathbf{A}$*, in the upper bound, we experience a dependence on* $1/(1-\overline{\rho})$ *raised to a higher power (either* 1 *for the term multiplied by* d *and* $3/2$ *for the term multiplied by* $\sqrt{d}$*) w.r.t. the exponent appearing in the lower bound (i.e.,* $1/2$*). It is currently an open question whether the lower bound is not tight (which is obtained for a simpler setting in which the state is observable* $\mathbf{x}_t$*) or whether more efficient algorithms for DLBs can be designed. Furthermore, Theorem 7.3.2 highlights the impact of the upper bound* $\overline{\rho}$ *compared with the true* $\rho(\mathbf{A})$*.*

7.4 Related Works

In this section, we survey and compare the literature from the perspective of the online learning algorithms with similar objectives. In Section 7.4.1, we compare and map our setting with the ones of bandits with delayed, aggregated, and composite feedback. In Section 7.4.2, we consider online control for Linear Time-Invariant (LTI) systems. In Section 7.4.3, we make a comparison with adversarial bandits. Finally, in Sections 7.4.4 and 7.4.5, we discuss POMDPs and others settings sharing similarities with DLBs.

7.4.1 Bandits with Delayed/Aggregated/Composite Feedback

The Multi-Armed Bandit setting has been widely employed as a principled approach to address sequential decision-making problems (Lattimore and Szepesvári, 2020). The possibility of experiencing delayed rewards has been introduced by Joulani et al. (2013) and widely exploited in advertising applications (Chapelle, 2014; Vernade et al., 2017). A large number of approaches have extended this setting either considering stochastic delays (Vernade et al., 2020), unknown delays (Li et al., 2019; Lancewicki et al., 2021), arm-dependent delays (Manegueu et al., 2020), non-stochastic delays (Ito et al., 2020; Thune et al., 2019; Jin et al., 2022). Some methods relaxed the assumption that the individual reward is revealed after the delay expires, admitting the possibility of receiving anonymous feedback, which can be aggregated (Pike-Burke et al., 2018; Zhang et al., 2022) or composite (Cesa-Bianchi et al., 2018; Garg and Akash, 2019; Wang et al., 2021b). Most of these approaches are able to achieve $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, plus additional terms depending on the extent of the delay. In our DLBs, the reward is generated over time as a combined effect of past and present actions through a *hidden state*, while these approaches generate the reward instantaneously and reveal it (individually or in aggregate) to the learner in the future and no underlying state dynamics is present.

Delayed/Aggregated Feedback with DLBs In what follows, we show how we can model *delayed* and *composite* feedback with DLBs. For the delayed feedback, we focus on the case in which either the delay is fixed to the value $\tau \geqslant 1$, i.e., the reward of the pull performed at round t is experienced at round $t + \tau$. For the composite feedback, we assume that the reward of the pull performed at round t is spread over the next $\tau \geqslant 1$ rounds with fixed weights $(w_1, \ldots, w_\tau)$. Denoting with R_t the full reward (not observed) due to the pull performed at round t, the agent at round t

observes the weighted sum of the rewards reported below:[8]

$$\sum_{l=1}^{\tau} w_l R_{t-l}. \tag{7.7}$$

These two cases can be modeled as DLBs with a suitable encoding of the arms and choice of matrices. In particular, assuming to have K arms, we take the arm set $\mathcal{U}$ to be the canonical basis of $\mathbb{R}^K$, and we denote with $\boldsymbol{\mu}$ the vector of expected rewards. We define $\boldsymbol{\theta} = \mathbf{0}$ and:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \in \mathbb{R}^{\tau \times \tau}, \qquad \mathbf{B} = \begin{pmatrix} \boldsymbol{\mu}_K^T \\ \mathbf{0}_K^T \\ \mathbf{0}_K^T \\ \vdots \\ \mathbf{0}_K^T \end{pmatrix} \in \mathbb{R}^{\tau \times K},$$

$$\boldsymbol{\omega}_{\text{delay}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{\tau}, \qquad \boldsymbol{\omega}_{\text{composite}} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_\tau \end{pmatrix} \in \mathbb{R}^{\tau}.$$

However, DLBs cannot model random or adversarial delays. Nevertheless, DLBs can capture scenarios of composite feedback in which the reward is spread over an infinite number of rounds. Keeping the K-armed case introduced above, we can consider the simplest example of a reward that spreads as an autoregressive process AR(1) with parameter $\gamma \in (0, 1)$, that cannot be represented using the standard composite feedback. In such a case, we simply need a system with order $n = 1$ with matrices (actually scalars):

$$\mathbf{A} = \gamma, \qquad \mathbf{B} = \mathbf{u}^T, \qquad \boldsymbol{\omega} = 1.$$

Clearly, one can consider AR(m) processes (Bacchiocchi et al., 2024) by employing systems of order $n = m > 1$.

[8] It is worth noting that the fixed-delay case is a particular case of composite feedback, where $w_1 = \cdots = w_{\tau-1} = 0$ and $w_\tau = 1$.

7.4.2 Online Control of Linear Time-Invariant Systems

The particular structure imposed by linear dynamics makes our approach comparable to LTI (Hespanha, 2018) online control for partially observable systems (e.g., Lale et al., 2020b; Simchowitz et al., 2020; Plevrakis and Hazan, 2020). While the dynamical model is similar, in online control of LTI systems, the perspective is quite different. Most of the works either consider the Linear Quadratic Regulator (Mania et al., 2019; Lale et al., 2020b) or (strongly) convex objective functions (Mania et al., 2019; Simchowitz et al., 2020; Lale et al., 2020a), achieving, in most of the cases $\widetilde{\mathcal{O}}(\sqrt{T})$ regret for strongly convex functions and $\widetilde{\mathcal{O}}(T^{2/3})$ for convex functions. Recently, $\widetilde{\mathcal{O}}(\sqrt{T})$ regret rate has been obtained for convex function too, by means of geometric exploration methods (Plevrakis and Hazan, 2020). Compared to `DynLin-UCB`, the algorithm of Plevrakis and Hazan (2020) considers general convex costs but assumes the observability of the state and limits to the class of disturbance response controllers (Li and Bosch, 1993) that do not include the constant policy. Moreover, the regret bound of Plevrakis and Hazan (2020) differs from Theorem 7.3.2, as it shows a cubic dependence on the system order[9] and an implicit non-trivial dependence on the dynamic matrix $\mathbf{A}$. Instead, our Theorem 7.3.2 is remarkably independent of the system order n. Furthermore, Lale et al. (2020a) reach $\mathcal{O}(\log(T))$ regret in the case of strongly convex cost functions competing against the best *persistently exciting* controller (i.e., a controller implicitly maintaining a non-null exploration). Some approaches are designed to deal with adversarial noise (Simchowitz et al., 2020). All of these solutions, however, look for the best closed-loop controller within a specific class, e.g., disturbance response control (Li and Bosch, 1993). These controllers, however, do not allow us to easily incorporate constraints on the action space, which could be of crucial importance in practice, e.g., in advertising domains. `DynLin-UCB` works with an arbitrary action space and, thanks to the linearity of the reward, does not require complex closed-loop controllers.

7.4.3 Adversarial Bandits

It is worth elaborating on the adaptation of adversarial MAB algorithms to this setting. First, since the reward distribution in DLBs depends at every round t on the sequence of actions played by the agent prior to t, we can reduce the DLB setting to an adversarial bandit with an *adaptive*

[9] This holds for *known* cost functions. Instead, for *unknown* costs, the exponent becomes 24 (Plevrakis and Hazan, 2020).

(or non-oblivious) adversary. Second, such an adversary must have *infinite memory* in principle. Third, our regret definition of Section 7.2 is a *policy regret* (Dekel et al., 2012) that compares the algorithm performance against playing the optimal policy in hindsight from the beginning, as opposed to the *external regret* often employed for non-adaptive adversaries. It is well known that for infinite-memory adaptive adversaries, no algorithm can achieve sublinear policy regret. Nevertheless, for DLB setting, we know that the effect of the past is always vanishing (given Assumption 7.1 enforcing $\rho(\mathbf{A}) < 1$), so we can approximate our setting as a *finite-memory* setting, by considering memory length $k \propto \lceil \frac{\log M}{\log 1/\overline{\rho}} \rceil$, where M is the one defined in Algorithm 7.1 (line 2), with an additional regret term only logarithmic in the optimization horizon T. Then, given this approximation, we can make use of an adversarial bandit algorithm (designed for non-adaptive adversaries) in the framework proposed by Dekel et al. (2012) to make it effective for the finite-memory adaptive adversary setting. In the case of an optimal algorithm, such as Exp3 Auer et al. (2002b), suffering an external regret of order $\widetilde{\mathcal{O}}(\sqrt{MT})$, being M the number of arms, the version to address this finite-memory adaptive adversary setting suffers a regret bounded by $\widetilde{\mathcal{O}}((k+1)M^{1/3}T^{2/3})$, as shown in Theorem 2 of Dekel et al. (2012).

7.4.4 Partially Observable Markov Decision Processes

As already noted, looking at DLBs in their generality, we realize that our model is a particular subclass of the Partially Observable Markov Decision Processes (POMDP, Åström, 1965). However, in the POMDP literature, no particular structure of the hidden state dynamics is assumed. The specific linear dynamics are rarely considered, as well as the possibility of a reward that is a linear combination of the hidden state and the action. Nevertheless, several works accounted for the presence of constraints (Isom et al., 2008; Undurti and How, 2010; Kim et al., 2011) without exploiting the linearity and without regret guarantees.

7.4.5 Other Approaches

Non-stationary bandits (Gur et al., 2014) can be regarded as bandits with a hidden state that evolves through a (possibly non-linear) dynamics. The main difference compared with our DLBs is that the hidden state evolves in an *uncontrollable* way, i.e., it does not depend on the sequence of actions performed so far. Russac et al. (2019) extend the linear bandit setting by considering a non-stationary evolution of the parameter $\boldsymbol{\theta}_t^*$. The notion of *dynamic* bandit is further studied by Chen et al. (2023), where an autoregressive process is considered for the evolution of the reward through time and by Nobari (2019) that propose a practical approach to cope with this setting.

7.5 Numerical Simulations

In this section, we provide numerical validations of `DynLin-UCB` in both a synthetic scenario and a domain obtained from real-world data related to the optimization of a *Marketing Mix Model*. The goal of these simulations is to highlight the behavior of `DynLin-UCB` in comparison with bandit baselines, describing advantages and disadvantages. The first experiment is a synthetic setting in which we can evaluate the performances of all the solutions and the sensitivity of `DynLin-UCB` w.r.t. the $\overline{\rho}$ parameter (Section 7.5.1). Then, we show a comparison in a DLB scenario retrieved from real-world data (Section 7.5.2). Finally, we briefly discuss about the running time of the `DynLin-UCB` (Section 7.5.3). The code of the experiments can be found at `https://github.com/marcomussi/DLB`.

Baselines We consider as main baseline `Lin-UCB` (Abbasi-Yadkori et al., 2011), designed for linear bandits. We include `Exp3` (Auer et al., 1995) usually employed in (non-adaptive) adversarial settings, and its extension to n-length memory (adaptive) adversaries `B-Exp3` by Dekel et al. (2012).[10] Additionally, we perform a comparison with algorithms for regret minimization in non-stationary environments: `D-Lin-UCB` (Russac et al., 2019), an extension of `Lin-UCB` for non-stationary settings, and `AR2` (Chen et al., 2023), a bandit algorithm for processes presenting temporal structure. Finally, in the case of real-world data, we compare our solution with a human-expert policy (`Expert`). This policy is directly generalized from the original dataset by learning via regression the average budget allocation over all platforms from the available data.

For the baselines which do not support vectorial actions, we perform a discretization of the action space $\mathcal{U}$ that surely contains optimal action. Concerning the hyperparameters of the baselines, whenever possible, they are selected as in the respective original papers. The experiments are presented with a regularization parameter $\lambda \in \{1, \log T\}$ for the algorithms which require it (i.e., `DynLin-UCB`, `Lin-UCB`, and `D-Lin-UCB`).[11] The bounds used for the exploration in `Lin-UCB` and `D-Lin-UCB` are ad-

[10] In Section 7.4.3 we elaborate on the use of adversarial bandit algorithms for DLBs.

[11] For `DynLin-UCB`, $\log T$ is a nearly optimal choice for λ as it can be seen by looking at the first two addenda of the exploration factor in Equation (7.6).

justed in order to be able to fairly compete in this setting, and are considered as follows:

$$\beta_t^{\texttt{Lin-UCB}} := \overline{c}_2\sqrt{\lambda} + \sqrt{2\overline{\sigma}^2\left(\log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1 + \frac{tU^2}{d\lambda}\right)\right)},$$

$$\beta_t^{\texttt{D-Lin-UCB}} := \overline{c}_2\sqrt{\lambda} + \sqrt{2\overline{\sigma}^2\left(\log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1 + \frac{tU^2}{d\lambda}\left(\frac{1-\gamma^{2t}}{1-\gamma^2}\right)\right)\right)},$$

where $\overline{c}_2$ and $\overline{\sigma}^2$ are as prescribed in Section 7.3.2, and the hyperparameter γ (i.e., the forgetting factor) of `D-Lin-UCB` is tuned. For `AR2`, the hyperparameter α, describing the correlation over time is considered equal to $\rho(\mathbf{A})$. In the case of `Exp3`, the rewards are rescaled in order to make them range in $[0, 1]$ with high probability, as follows:

$$\overline{r}_t = \frac{r_t + 2\xi}{4\xi}, \qquad \text{where} \qquad \xi = \left(\Theta + \frac{\Omega B}{1-\rho(\mathbf{A})}\right)U.$$

Furthermore, in the case of `B-Exp3` , the batch dimension k is considered as:

$$n = \left\lceil \frac{\log M}{\log 1/\overline{\rho}} \right\rceil,$$

where M is the one defined in Algorithm 7.1 (line 2). This batch size n ensures that, at each time t, the contribution of actions $\mathbf{u}_s$ is negligible, with $s \in [\![t-n-1]\!]$. The rewards collected in the same batch are averaged and transformed as in `Exp3`.

7.5.1 Synthetic Data

Setting We consider a DLB defined by the following matrices:

$$\begin{aligned}
\mathbf{A} &= \mathrm{diag}((0.2, 0, 0.1)),\\
\mathbf{B} &= \mathrm{diag}((0.25, 0, 0.1)),\\
\boldsymbol{\theta} &= (0, 0.5, 0.1)^{\mathsf{T}},\\
\boldsymbol{\omega} &= (1, 0, 0.1)^{\mathsf{T}},
\end{aligned}$$

and a Gaussian noise with $\sigma = 0.01$ (diagonal covariance matrix for the state noise).[12] This way, the spectral gap of the dynamical matrix is $\rho(\mathbf{A}) =$

[12] It is worth noting that the decision of using diagonal matrices is just for explanation purposes and w.l.o.g. (at least in the class of diagonalizable dynamic matrices). Indeed, we are just interested in the cumulative Markov parameter $\mathbf{h}$ and we could have obtained the same results with an equivalent (non-diagonal) representation, by applying an inevitable transformation $\mathbf{T}$ as $\mathbf{A}' = \mathbf{TAT}^{-1}$, $\boldsymbol{\omega}' = \mathbf{T}^{-\mathsf{T}}\boldsymbol{\omega}$, and $\mathbf{B}' = \mathbf{TB}$.

0.2 and $\Phi(\mathbf{A}) = 1$. Moreover, the cumulative Markov parameter is given by $\mathbf{h} = (0.56, 0.5, 0.11)^{\mathrm{T}}$. We consider the action space $\mathcal{U} = \{(u_1, u_2, u_3)^{\mathrm{T}} \in [0,1]^3$ with $u_1 + u_2 + u_3 \leqslant 1.5\}$ that simulates a total budget of 1.5 to be allocated to the three platforms. Thus, a "myopic" agent would simply look at how the action immediately propagates to the reward through $\boldsymbol{\theta}$, and will invest the budget in the second component of the action, which is weighted by 0.5. Instead, a "far-sighted" agent, aware of the system evolution, will look at the cumulative Markov parameter $\mathbf{h}$, realizing that the most convenient action is investing in the first component, weighted by 0.56. Therefore, the optimal action is $\mathbf{u}^* = (1, 0.5, 0)^{\mathrm{T}}$ leading to $J^* = 0.81$.

Comparison with the bandit baselines Figure 7.1 shows the performance in terms of cumulative regret of `DynLin-UCB`, `Lin-UCB`, `D-Lin-UCB`, `AR2`, `Exp3`, and `B-Exp3` . The experiments are conducted over a time horizon of 1 million rounds. For `DynLin-UCB`, we employed, for the sake of this experiment, the true value of the spectral gap, i.e., $\overline{\rho} = \rho(\mathbf{A}) = 0.2$. First of all, we observe that both `Exp3` and `B-Exp3` suffers a significantly large cumulative regret. Similar behavior is displayed by `AR2`. Moreover, all the versions of `Lin-UCB` and `D-Lin-UCB` suffer linear regret. The best performance of `D-Lin-UCB` is obtained when the forgetting factor γ is close to 1 (the weights take the form $w_t = \gamma^{-t}$), and the behavior is comparable with the one of `Lin-UCB`. Even for a quite fast system ($\rho(\mathbf{A}) = 0.2$), ignoring the system dynamics, and the presence of the hidden state, has made both `Lin-UCB` and `D-Lin-UCB` commit (in their best version, with $\lambda = \log T$) to the sub-optimal (myopic) action $\mathbf{u}^\circ = (0.5, 1, 0)^{\mathrm{T}}$ with performance $J^\circ = 0.78 < J^*$, with also a relevant variance. On the other hand, `DynLin-UCB` is able to maintain a smaller and stable (variance is negligible) sublinear regret in both its versions, with a notable advantage when using $\lambda = \log T$.

Sensitivity to the Choice of $\overline{\rho}$ The upper bound $\overline{\rho}$ of the spectral radius $\rho(\mathbf{A}) = 0.2$ represents a crucial parameter of `DynLin-UCB`. While an overestimation $\overline{\rho} \gg \rho(\mathbf{A})$ does not compromise the regret rate but tends to slow down the convergence process, a severe underestimation $\overline{\rho} \ll \rho(\mathbf{A})$ might prevent learning at all. In Figure 7.2, we test `DynLin-UCB` against a misspecification of $\overline{\rho}$, when $\lambda = \log T$. We can see that by considering $\overline{\rho} = 2\rho(\mathbf{A})$, `DynLin-UCB` experiences a larger regret but still sublinear and smaller w.r.t. `Lin-UCB` with $\lambda = \log T$. Even by reducing $\overline{\rho} \in \{0.1, 0.05\}$, `DynLin-UCB` is able to keep the regret sublinear, showing remarkable robustness to misspecification. Clearly, setting $\overline{\rho} = 0$ makes the regret

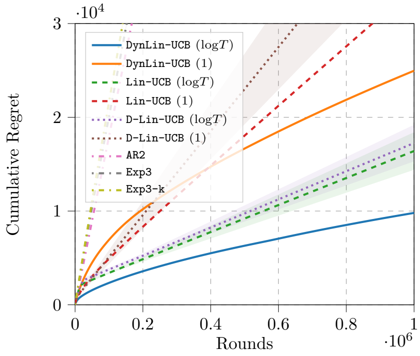


Figure 7.1: *Cumulative regret as a function of the rounds comparing* `DynLin-UCB` *and the other bandit baselines (50 runs, mean ± std).*

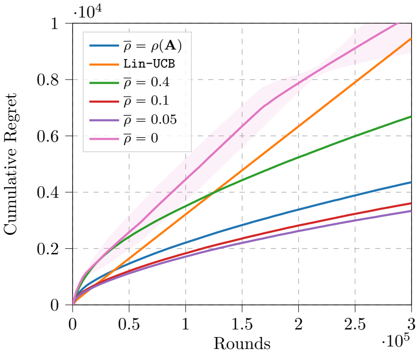


Figure 7.2: *Cumulative regret as a function of the rounds comparing* `Lin-UCB`*, and* `DynLin-UCB` *with* $\lambda = \log T$*, varying the upper bound on the spectral radius* $\overline{\rho}$ *(50 runs, mean ± std).*

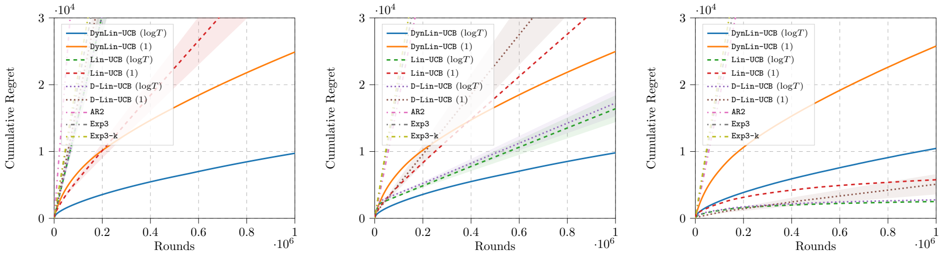
almost degenerate to linear.

Empirical study on the noise σ We want to analyze the behavior of our solution and the other baselines at different magnitudes of noise in both the state transition model and the output. The noise in this simulation is a zero-mean Gaussian random noise with $\sigma \in \{0.001, 0.01, 0.1\}$. Figure 7.3 shows the results of the experiment for the different values of σ.[13] It is clearly visible how `DynLin-UCB` performs in almost the same way no matter the noise to which the system is subject, always leading to sub-linear regret. On the other hand, the cumulative regret of both `Lin-UCB` and `D-Lin-UCB` is different in every simulation we perform. Indeed, with a low level of noise (Figure 7.3a) reaches linear regret and does not converge, while for large values of noise, it converges very quickly (Figure 7.3c). This is due to the nature of the confidence bound of linear bandits, which is not able to take into account such a complex scenario and leads to no guarantees in this setting. `Exp3`, `B-Exp3` , and `AR2` are not able to reach the optimum in this scenario, independently from the noise magnitude σ, and provide large values of (linear) regret.

7.5.2 Real-world Data - Marketing Mix Model

We present an experimental evaluation based on real-world data coming from three web advertising platforms (`Facebook`, `Google`, and `Bing`),

[13] Figure 7.3b is the same as Figure 7.1 and is reported for the sake of simplicity in the comparison.



(a) $\sigma = 0.001$. **(b)** $\sigma = 0.01$. **(c)** $\sigma = 0.1$.

Figure 7.3: *Performance of* `DynLin-UCB`*,* `Lin-UCB`*,* `D-Lin-UCB`*,* `AR2`*,* `Exp3` *and* `B-Exp3` *at different values of σ (50 runs, mean ± std).*

related to several campaigns for an invested budget of 5 Million EUR over 2 years. The data are representative of a complex advertising scenario, in which the advertisers are manually optimizing the *Marketing Mix Model*. Starting from such data, considering the budgets as actions, and the conversions as target, we learn the best DLB model by means of a specifically designed variant of the Ho-Kalman algorithm (Ho and Kalman, 1966) described in Appendix E. We used the matrices estimated with the Ho-Kalman method to build up a simulator. The resulting system has $\rho(\mathbf{A}) = 0.67$, and is characterized as follows:

$$\mathbf{A} = \begin{pmatrix} 0.38 & 0.33 & 0.6 \\ 0.07 & 0.76 & -0.54 \\ 0.18 & 0.34 & 0.05 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 0.14 & 0.34 & -0.05 \\ -0.17 & 0.03 & -0.01 \\ 0.04 & -0.09 & 0.17 \end{pmatrix},$$

$$\boldsymbol{\omega} = \begin{pmatrix} -0.61 \\ -0.04 \\ -0.13 \end{pmatrix}, \qquad \boldsymbol{\theta} = \begin{pmatrix} 0.13 \\ 0.41 \\ 0.02 \end{pmatrix}.$$

We evaluate `DynLin-UCB` against the baselines for $T = 10^6$ steps over 50 runs.

Results Figure 7.4 shows the results in terms of cumulative regret. It is worth noting that no algorithm, except for `DynLin-UCB`, is able to converge to the optimal choice. Indeed, they immediately commit to a sub-optimal solution. `DynLin-UCB`, instead, shows a convergence trend towards the optimal policy over time for both $\lambda = 1$ and $\lambda = \log T$, even if the best-performing version is even in this case the one which employs $\lambda = \log T$. The `Expert`, which has a preference towards maximizing the instantaneous effect of the actions only and does not take into account correlations between platforms, displays a sub-optimal performance.

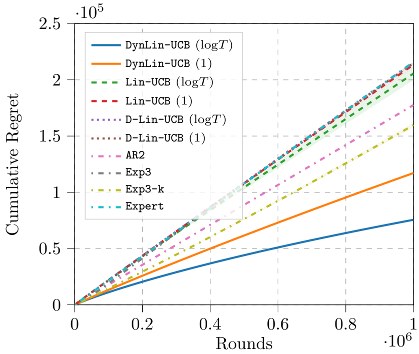


Figure 7.4: *Cumulative regret for* `DynLin-UCB`*, the other bandit baselines and the* `Expert` *in the system generalized from real-world data (50 runs, mean ± std).*

7.5.3 Computational Time

The code used for the results provided in this section has been run on an Intel(R) I5 8259U @ 2.30GHz CPU with 8 GB of LPDDR3 system memory. The operating system was macOS 12.2.1, and the experiments have been run on *Python* 3.9.7. A single run of `DynLin-UCB` takes 110 seconds to run. It is worth noting that the *time complexity* of `DynLin-UCB` is upper-bounded by the one of `Lin-UCB`.

7.6 Discussion and Conclusions

In this chapter, we introduced the Dynamical Linear Bandits (DLBs), a novel model to represent sequential decision-making problems in which the system is characterized by a non-observable hidden state that evolves according to linear dynamics and by an observable noisy reward that linearly combines the hidden state and the action played. This model accounts for scenarios that cannot be easily represented by existing bandit models that consider delayed and aggregated feedback, like the one of the *Marketing Mix Model*. We derived a regret lower bound that highlights the main complexities of the DLB problem. Then, we proposed a novel optimistic regret minimization approach, `DynLin-UCB`, that, under stability assumption, is able to achieve sub-linear regret. The numerical simulation in both synthetic and real-world domains succeeded in showing that, in a setting where the baselines mostly suffer linear regret, our algorithm consistently enjoys sublinear regret. Furthermore, `DynLin-UCB` proved to be robust to misspecification of its most relevant hyper-parameter $\overline{\rho}$. To the best of our knowledge, in this chapter, we present the first work addressing this family of problems, characterized by hidden linear dynamics, with a simple, yet effective, bandit-like approach.

Part III

Joint Pricing and Advertising

CHAPTER 8

Factored Reward Bandits for Joint Pricing and Advertising

In this chapter, we introduce the Factored-Reward Bandits (FRBs), a novel setting able to effectively capture and exploit the structure of this class of scenarios, where the reward is computed as the product of the action intermediate observations. We characterize the statistical complexity of the learning problem in the FRBs, by deriving worst-case and asymptotic instance-dependent regret lower bounds. Then, we devise and analyze two regret minimization algorithms. The former, `F-UCB`, is an anytime optimistic approach matching the worst-case lower bound (up to logarithmic factors) but fails to perform optimally from the instance-dependent perspective. The latter, `F-Track`, is a bound-tracking approach, that enjoys optimal asymptotic instance-dependent regret guarantees.

This chapter presents (Mussi et al., 2024), a joint work with Simone Drago, Alberto Maria Metelli and Marcello Restelli, published at the *International Conference on Machine Learning (ICML)*. A preliminary version of this work (Drago et al., 2024) appeared at the *Adaptive and Learning Agents Workshop*.

8.1 Introduction

In several real-world sequential decision-making problems, the learner is required to select, at every interaction, different actions, i.e., an *action vector*, acting on different portions of the system, each producing an *intermediate observation*. In such scenarios, the reward is often a combination of these observations. Consider, for instance, the case in which we want to sell a product on an e-commerce website. Our goal is to maximize the overall revenue derived from the sales of a given item. In this business process, we have to choose (i) the *price* at which to sell the product and (ii) how much *budget* to invest in advertising. On the one hand, the price we set determines the propensity of the users to buy a given item, i.e., the *conversion rate*, representing for each price, the fraction of the customers that will buy the item (Broder and Rusmevichientong, 2012; Den Boer, 2015). On the other hand, the advertising budget we invest influences the number of potential customers that will be exposed to such an item, i.e., the number of *impressions* we are able to generate with the advertisement campaign (Feldman et al., 2007). Thus, every time we select a *price-budget* pair (i.e., *action vector*), we observe a noisy realization of the conversion rate, which depends on the price, and a noisy realization of the expected number of impressions, which depends on the budget we invest in advertising (i.e., *intermediate observations*). Thus, our objective is to maximize the revenue (i.e., *reward*) that is computed as the *product* between the price, the conversion rate, and the impressions (which will give us our income) subtracting the invested advertising budget.

This scenario can be, in principle, addressed as a standard Multi-Armed Bandit (MAB, Lattimore and Szepesvári, 2020) by looking at the reward (i.e., revenue) only and considering price-budget couples as actions. However, with such an approach, intermediate observations (i.e., the *conversion rate* – consequence of the price we set – and the *impressions* we generate – a consequence of the adv budget we invest) that could provide useful information would be ignored with a possible detrimental effect on the learning process. Indeed, if we look just at the reward and disregard this *factored* structure, the learning problem will: (i) present an unnecessarily large action space, including all the possible combinations of action components (e.g., price and budget pairs), and (ii) suffer a possibly amplified effect of the noise in the reward due to the product of the noisy intermediate observations (e.g., *impressions* times *conversion rate*).

A notion of *factored bandits* has been studied in (Zimmert and Seldin, 2018) in which the expected reward is a *general* function of the action components. No intermediate observations are considered and the noise is applied to the final reward only. Thus, this setting ultimately fails to model the real-world scenarios we are interested in, where the intermediate observations play a crucial role and are combined with a *specific* function (i.e., the product). As we shall see later in the chapter, this specificity, motivated by the considered real-world scenarios, will allow us to obtain tighter and more detailed performance guarantees.

Contributions In this chapter, we propose the novel setting of the *Factored-Reward Bandits* (FRBs) to model sequential decision-making problems in which the agent is required to play an action vector $\mathbf{a} = (a_1, \dots, a_d)^{\mathrm{T}}$ consisting of d action components. Each action component a_i provides a noisy intermediate observation x_i whose product forms the reward $r = x_1 x_2 \cdots x_d$. We study this setting from computational and statistical perspectives and propose two regret minimization algorithms endowed with theoretical guarantees. The contributions are summarized as follows:

- In Section 8.2, we introduce the FRB setting, describe the *feedback and noise models*, and the learning problem.

- In Section 8.3, we study the *statistical complexity* of the learning problem in the FRB setting by deriving regret *lower bounds*. First, in Theorem 8.3.1, we present the *worst-case* regret lower bound of order $\Omega(\sigma d\sqrt{kT})$, being σ the subgaussian proxy, d the number of action components, k the number of possible choices for each action component, and T is the learning horizon.[1] This result highlights how the complexity of the problem scales linearly with d and its derivation makes use of technical tools from the multitask bandits literature. In Theorem 8.3.2, we show that dependence on σ^d (exponential in d) is unavoidable when intermediate observations are not present, motivating their crucial role. Second, we present the *instance-dependent* asymptotic regret lower bound which is first formulated as a linear program of $\mathcal{O}(k^d)$ variables (Theorem 8.3.3) and, subsequently, elaborated in a more explicit form (Theorem 8.3.4), whose derivation makes use of the *rearrangement inequalities* (Hardy et al., 1952) and that enjoys a computational complexity of $\mathcal{O}(dk \log k)$. Qualitatively, this result shows how the different action components choices need to *coordinate* to match the lower bound.

- In Section 8.4, we provide a novel intuitive *optimistic anytime regret*

[1] In the following, we provide more general results in which each action component i can have a different number k_i of choices.

minimization algorithm, `Factored Upper Confidence Bound` (`F-UCB`), in which optimism is applied to every action component *independently*. Then, we characterize its *worst-case* regret which has order $\widetilde{\mathcal{O}}(\sigma d\sqrt{kT})$, matching the lower bound up to logarithmic factors (Theorem 8.4.1). Then, we empirically study its *instance-dependent* regret, revealing that it does not match the lower bound (Theorem 8.4.3). This confirms how *coordination* between action components is necessary.

- In Section 8.5, we design and analyze a novel algorithm, *`Factored Track`* (`F-Track`). `F-Track` is based on *tracking* the bound (Lattimore and Szepesvari, 2017), and succeeds in matching the instance-dependent lower bound in the asymptotic regime (Theorem 8.5.1). Its analysis reveals, once more, the need for coordinating the action components to achieve the optimal performance.

Numerical simulations are provided in Section 8.6. Section 8.7 discusses the relevant literature for the FRB setting. The proofs of all the statements are reported in Appendix D.

8.2 Setting

In this section, we introduce the *Factored-Reward Bandits* (FRBs), the learner-environment interaction, the assumptions, and we present the learning problem.

Problem Formulation Let $T \in \mathbb{N}$ be the learning horizon. In a Factored-Reward Bandits, at every round $t \in [\![T]\!]$, the learner chooses an *action vector* $\mathbf{a}(t) = (a_1(t), \dots, a_d(t))^{\mathrm{T}}$ in the action space $\mathcal{A} := [\![k_1]\!] \times \cdots \times [\![k_d]\!]$, where for every $i \in [\![d]\!]$ we have that $k_i \in \mathbb{N}_{\geqslant 2}$ is the number of options of the i^{th} *action component* $a_i(t)$ of the vector, and $d \in \mathbb{N}_{\geqslant 1}$ is the action vector dimension (i.e., the number of components that the learner must select at every round t). As an effect of the action, the learner observes a vector of d *intermediate observations* $\mathbf{x}(t) = (x_1(t), \dots, x_d(t))^{\mathrm{T}}$ and receives as reward the product of the intermediate observations $r(t) = \prod_{i \in [\![d]\!]} x_i(t)$. The i^{th} component $x_i(t)$ of the intermediate observation vector $\mathbf{x}(t)$ is the effect of the i^{th} action component $a_i(t)$ in the action vector $\mathbf{a}(t)$. Specifically, every component $i \in [\![d]\!]$ of the intermediate observation vector $\mathbf{x}(t)$ is independent of the others and sampled from a distribution $x_i(t) \sim \nu_{i,a_i(t)}$, so that, $\mathbf{x}(t) \sim \boldsymbol{\nu}_{\mathbf{a}(t)} := \otimes_{i \in [\![d]\!]} \nu_{i,a_i(t)}$. Thus, we will denote an FRB as $\underline{\boldsymbol{\nu}} := \otimes_{i \in [\![d]\!]} \otimes_{a_i \in [\![k_i]\!]} \nu_{i,a_i}$. Furthermore, we can write $x_i(t) = \mu_{i,a_i(t)} + \epsilon_i(t)$, where $\mu_{i,a_i(t)}$ is the *expected intermediate observation* of the i^{th} action component $a_i(t)$, and $\epsilon_i(t)$ is σ^2-subgaussian random noise, independent conditioned to the past and the other noise realizations $\epsilon_j(t)$ for $j \in [\![d]\!]\backslash\{i\}$. As customary, we assume bounded expected values for the intermediate observations, i.e., $\mu_{i,a_i} \in [0,1]$ for every $i \in [\![d]\!]$ and $a_i \in [\![k_i]\!]$, and all intermediate observation components $x_i(t)$ characterized by the same known subgaussian proxy σ.[2]

Learning Problem An optimal action vector is $\mathbf{a}^* = (a_1^*, \ldots, a_d^*)^{\mathrm{T}} \in \arg\max_{\mathbf{a}=(a_1,\ldots,a_d)^{\mathrm{T}}\in\mathcal{A}} \prod_{i\in[\![d]\!]} \mu_{i,a_i}$ and, since all expected intermediate observations are non-negative, we can factorize the optimization problem observing that $a_i^* \in \arg\max_{a_i\in[\![k_i]\!]} \mu_{i,a_i}$ for every $i \in [\![d]\!]$. We denote with $\mu_i^* = \mu_{i,a_i^*}$ the expected intermediate observation of the optimal i^{th} action component. We define the suboptimality gap related to the i^{th} action component as $\Delta_{i,a_i} := \mu_i^* - \mu_{i,a_i}$ for $a_i \in [\![k_i]\!]$, and the suboptimality gap related to the action vector $\mathbf{a} = (a_1, \ldots, a_d)^{\mathrm{T}} \in \mathcal{A}$ as $\Delta_{\mathbf{a}} := \prod_{i\in[\![d]\!]} \mu_i^* - \prod_{i\in[\![d]\!]} \mu_{i,a_i}$.

Let $\underline{\boldsymbol{\nu}}$ be an FRB, $\mathfrak{A}$ be a learning algorithm, and $T \in \mathbb{N}$ be the learning horizon, we define its *cumulative regret* as:

$$R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}}) := T \prod_{i\in[\![d]\!]} \mu_i^* - \sum_{t\in[\![T]\!]} \prod_{i\in[\![d]\!]} \mu_{i,a_i(t)} = \sum_{t\in[\![T]\!]} \Delta_{\mathbf{a}(t)}. \tag{8.1}$$

The goal of the learner consists in minimizing the *expected cumulative regret* $\mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})]$, where the expectation is taken w.r.t. the randomness of the observations and the possible randomness of the algorithm $\mathfrak{A}$.

Joint Pricing and Adverising as a FRB We now discuss how the joint pricing and advertising problem presented in Section 8.1 can be formalized as a FRB.

Example 8.2.1 (Joint Pricing and Advertising)**.** *Consider the case of joint pricing and advertising described in Section 8.1. In this scenario, at every round $t \in [\![T]\!]$, we must select a vector of dimension $d = 2$. Suppose that the first action component is the advertising* budget*, and the second action component is the selling* price*. We have k_1 advertising budgets over which we want to choose and k_2 prices at which we can sell our item. At every round t, we select the budget $a_1(t)$ and the price $a_2(t)$. Then, we observe a realization of the impressions we generate due to the budget $a_1(t)$ we invested: $x_1(t) = \mu_{1,a_1(t)} + \epsilon_1(t)$, and a realization of the conversion rate due to the price $a_2(t)$ we set: $x_2(t) = \mu_{2,a_2(t)} + \epsilon_2(t)$. The reward*

[2] The extension with different known subgaussian proxies σ_i for every component $i \in [\![d]\!]$ is straightforward.

is equal to $r(t) = a_2(t)x_1(t)x_2(t) - a_1(t)$*, corresponding to the return for each sales (the price, considering the turnover as target), multiplied by the fraction of users willing to buy and by the number of customers exposed to the price (i.e., the impressions), minus the budget invested in advertising. Note that the operations of multiplying by the selling price and subtracting the advertising budget do not increase the statistical complexity of the learning problem, as after we select an action, such quantities are deterministic. However, to deal with this more elaborated formulation, we have to take care of it in the choice of the optimal action* $\mathbf{a}^*$*:*

$$\mathbf{a}^* \in \operatorname*{arg\,max}_{\mathbf{a}=(a_1,a_2)^T\in\mathcal{A}} a_2 \prod_{i\in[\![2]\!]} \mu_{i,a_i} - a_1.$$

8.3 Regret Lower Bound

In this section, we provide lower bounds to the expected regret that any learning algorithm suffers when addressing the learning problem in a FRB, both in the minimax (Section 8.3.1) perspective and in the instance-dependent (Section 8.3.2) one.

8.3.1 Worst-Case Lower Bound

We present the worst-case lower bound that every algorithm suffers in the FRB setting and discuss the role of the structure of the FRB.

Theorem 8.3.1 (Worst-Case Lower Bound)**.** *For every algorithm* $\mathfrak{A}$*, there exists an FRB* $\underline{\boldsymbol{\nu}}$ *such that for:*

$$T \geqslant 2\big(1 - 2^{-\frac{1}{d-1}}\big)^{-2}\sigma^2 \max_{i\in[\![d]\!]} k_i, \tag{8.2}$$

$\mathfrak{A}$ *suffers an expected cumulative regret of at least:*

$$\mathbb{E}\left[R_T(\mathfrak{A},\underline{\boldsymbol{\nu}})\right] \geqslant \frac{\sigma}{4\sqrt{2}} \sum_{i\in[\![d]\!]} \sqrt{k_i T}.$$

In particular, if $k_i =: k$ *for every* $i \in [\![d]\!]$*, we have:*

$$\mathbb{E}\left[R_T(\mathfrak{A},\underline{\boldsymbol{\nu}})\right] \geqslant \Omega(\sigma d\sqrt{kT}).$$

Proof Sketch. The challenge is the structure of the regret in a FRB. We lower-bound the regret $R_T(\mathfrak{A},\underline{\boldsymbol{\nu}})$ as a sum of the regrets $R_T^{(i)}(\mathfrak{A},\underline{\boldsymbol{\nu}})$ that an algorithm $\mathfrak{A}$ would have suffered by playing d *parallel MABs*. Choosing $\mu_i^* = 1$:

$$\begin{aligned} R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}}) &= \sum_{t \in [\![T]\!]} \Big(1 - \prod_{i \in [\![d]\!]} \big(1 - \Delta_{i,a_i(t)}\big) \Big) \\ &\geqslant \frac{1}{2} \sum_{i \in [\![d]\!]} \sum_{t \in [\![T]\!]} \Delta_{i,a_i(t)} =: \frac{1}{2} \sum_{i \in [\![d]\!]} R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}}). \end{aligned}$$

This derivation leverages an ad-hoc technical Lemma D.1.2, which holds for sufficiently small suboptimality gaps, i.e., $\Delta_{i,a_i(t)} \leqslant 1 - 2^{-\frac{1}{d-1}}$. This condition gives rise to the constraint on the minimum time horizon (Equation 8.2), since the suboptimality gaps will be chosen $\propto T^{-1/2}$. Indeed, intuitively, if the suboptimality gaps Δ_{i,a_i} are too large (depending on d) we will have $1 - \prod_{i \in [\![d]\!]} (1 - \Delta_{i,a_i}) \ll \sum_{i \in [\![d]\!]} \Delta_{i,a_i}$ making the instances more distinguishable and, consequently, reducing the regret. The result is obtained by showing that regret component satisfies $R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}}) \geqslant \Omega(\sigma\sqrt{k_i T})$ redesigning for the subgaussian case the solution designed for Bernoulli rewards from the *multitask bandit* literature (Wang et al., 2021c, Theorem 10). □

To understand the beneficial effect of: (i) the factored structure and (ii) the intermediate observations, it is worth comparing the result of Theorem 8.3.1 with the regret lower bounds of common settings. If we remove (i), we are in the presence of a MAB with $\mathcal{A} = [\![k_1]\!] \times \cdots \times [\![k_d]\!]$ as action space.[3] It is worth noting that, even in this case, the reward $r(t) = \prod_{i \in [\![d]\!]} x_i(t)$ is the *product* of d subgaussian random variables which is not, in general, subgaussian (see Lemma D.2.1). Nevertheless, $r(t)$ is guaranteed to preserve a finite variance of order at least $\underline{\sigma}^2 = \sigma^{2d}$ (see Lemma D.2.3). Thus, we can look at the setting as a *heavy-tailed* MAB with finite variance (Bubeck et al., 2013) with $\prod_{i \in [\![d]\!]} k_i$ actions, leading to a regret of order $\Omega(\underline{\sigma}\sqrt{\prod_{i \in [\![d]\!]} k_i T})$, which becomes $\Omega(\sigma^d \sqrt{k^d T})$ when $k_i = k$ for every $i \in [\![d]\!]$.

It is natural to wonder if (i) is enough to break the exponential dependence in d (on both σ and k). This setting is similar, but not exactly overlapping, to that of Zimmert and Seldin (2018), in which a general "factored" structure is considered without intermediate observations and assuming that the subgaussian noise is applied to the reward directly. Nevertheless, (Zimmert and Seldin, 2018) provide neither worst-case lower bound nor worst-

[3] Note that makes no sense to consider (ii) without (i).

case regret analysis of the proposed algorithm. The following result shows that (i) only is enough to remove the exponential dependence in d on k but not on σ, which remains unavoidable without (ii).

Theorem 8.3.2 (Worst-Case Lower Bound without Intermediate Observations)**.** *For every algorithm $\mathfrak{A}^\dagger$ that ignores the intermediate observations $\mathbf{x}(t)$ and observes the reward $r(t)$ only, there exists an FRB $\underline{\boldsymbol{\nu}}$ such that for:*

$$T \geqslant 4(\min_{i \in \llbracket d \rrbracket} k_i - 1)/d,$$

$\mathfrak{A}^\dagger$ suffers an expected cumulative regret of at least:

$$\mathbb{E}\left[R_T(\mathfrak{A}^\dagger, \underline{\boldsymbol{\nu}})\right] \geqslant \frac{\sigma^d}{8}\sqrt{\frac{(\min_{i \in \llbracket d \rrbracket} k_i - 1)T}{d}}.$$

In particular, if $k_i =: k$ for every $i \in \llbracket d \rrbracket$, we have:

$$\mathbb{E}\left[R_T(\mathfrak{A}^\dagger, \underline{\boldsymbol{\nu}})\right] \geqslant \Omega(\sigma^d \sqrt{kT/d}).$$

Thus, Theorem 8.3.2 shows that the exponential dependence of d on σ is maintained even with the factored structure. This is particularly significant when $\sigma > 1$, a regime in which the function $\sigma^d/\sqrt{d}$ is exponentially increasing in d. This motivates the interest in studying this setting combining factored structure (i) and intermediate observations (ii).

Remark 8.3.1 (About the independence of the intermediate observations)**.** *The formulation of the FRB in Section 8.2 assumes that the components $x_i(t)$ of the observation vector $\mathbf{x}(t)$ are* independent. *This is necessary to treat the problem with appropriate advantages over standard MABs on the combinatorial action space $\mathcal{A}$. Indeed, if we rule out the independence assumption, we can always define a FRB in which $\mathbf{x}(t) = (y(t), 1, \dots, 1)^T$, where $y(t) \sim \nu_{1,\mathbf{a}(t)}$. This corresponds to a standard σ^2-subgaussian MAB with $\mathcal{A}$ as action space and arm distributions $\nu_{1,\mathbf{a}}$. Nevertheless, it is possible to relax the* independence *assumption, by requiring* non-correlation *among the intermediate observations.*

8.3.2 Instance-Dependent Lower Bound

We present the instance-dependent lower bound that every algorithm suffers on a specific instance $\underline{\boldsymbol{\nu}}$ of the FRB setting.

Theorem 8.3.3 (Instance-Dependent Lower Bound). *For every consistent*[4] *algorithm $\mathfrak{A}$ and FRB $\underline{\boldsymbol{\nu}}$ with unique optimal arm $\mathbf{a}^* \in \mathcal{A}$ it holds that:*

$$\liminf_{T \to +\infty} \frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T} \geqslant \underline{C}(\underline{\boldsymbol{\nu}}), \tag{8.3}$$

where $\underline{C}(\underline{\boldsymbol{\nu}})$ is defined as the solution to the following optimization problem:

$$\min_{(L_{\mathbf{a}})_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}}} \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}} L_{\mathbf{a}} \Delta_{\mathbf{a}} \tag{8.4}$$

$$\text{s.t.} \quad L_{i,j} = \sum_{\substack{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\} \\ a_i = j}} L_{\mathbf{a}}, \ \forall i \in [\![d]\!], \ j \in [\![k_i]\!] \setminus \{a_i^*\} \tag{8.5}$$

$$L_{i,j} \geqslant \frac{2\sigma^2}{\Delta_{i,j}^2}, \ \forall i \in [\![d]\!], j \in [\![k_i]\!] \setminus \{a_i^*\} \tag{8.6}$$

$$L_{\mathbf{a}} \geqslant 0, \quad \forall \mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}. \tag{8.7}$$

Proof Sketch. Here we provide an informal derivation that captures the intuition, although the formal proof requires some additional technical effort (see Appendix D.1.1). Thanks to the factored structure, we can show, as for stochastic bandits, that for every $j \in [\![k_i]\!] \setminus \{a_i^*\}$ and $i \in [\![d]\!]$ the expected number of pulls $\mathbb{E}[N_{i,j}(T)]$ is lower bounded by (Constraint 8.6):

$$L_{i,j} := \frac{\mathbb{E}[N_{i,j}(T)]}{\log T} \geqslant \frac{2\sigma^2}{\Delta_{i,j}^2} \qquad \text{for} \quad T \to +\infty$$

We now want to find the arrangements of the number of pulls of action vectors $N_{\mathbf{a}}(T)$, for every $\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}$, to minimize the cumulative regret. Recalling that $N_{i,j}(T) = \sum_{\mathbf{a} \in \mathcal{A} \,:\, a_i = j} N_{\mathbf{a}}(T)$, we define $L_{i,j} = \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\} \,:\, a_i = j} L_{\mathbf{a}}$ (Constraint 8.5). Finally, by recalling that $\frac{\mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})]}{\log T} = \sum_{\mathbf{a} \in \mathcal{A}} L_{\mathbf{a}} \Delta_{\mathbf{a}}$ we get the objective function in Equation (8.4) to be minimized. Notice that to make the proof fully formal we need to properly manage the asymptotic behavior of the sequences $\mathbb{E}[N_{i,j}(T)]$ and $\mathbb{E}[N_{\mathbf{a}}(T)]$ when $T \to +\infty$. □

The optimization problem in Theorem 8.3.3 is a Linear Program (LP) with $\prod_{i \in [\![d]\!]} k_i + \sum_{i \in [\![d]\!]} k_i - d - 1$ variables and $\prod_{i \in [\![d]\!]} k_i + 2 \sum_{i \in [\![d]\!]} k_i - 2d - 1$ constraints. Constraint (8.5) establishes the relation between the number of pulls of the action vectors $L_{\mathbf{a}}$ and the number of pulls of the action components $L_{i,j}$. This captures the "information sharing" of the setting in which

[4] An algorithm $\mathfrak{A}$ is *consistent* if for every FRB $\underline{\boldsymbol{\nu}}$ and $p > 0$, it holds that $\limsup_{T \to +\infty} \mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})]/T^p = 0$.

we obtain a sample for the action component (i, j) whenever we pull an action vector $\mathbf{a}$ such that $a_i = j$. Being a minimization problem, Constraint (8.6) will be satisfied with equality allowing the removal of variables $L_{i,j}$ and the relative constraints. Thus, the LP can be solved in polynomial time w.r.t. $\prod_{i \in [\![d]\!]} k_i$ (Vaidya, 1989).

Explicit Solution of the LP Program We now illustrate how to solve the LP program with a smaller time complexity of order $\mathcal{O}(\sum_{i \in [\![d]\!]} k_i \log k_i)$. We first provide the intuition and, then, provide the formal argument.

The minimum proportion with which the action component (i, j) is to be pulled (Constraint 8.6) can be accomplished by pulling different sequences of action vectors $\mathbf{a}$ such that $a_i = j$. *How to "arrange" the pulls of the action vectors to satisfy Constraint* (8.6) *and minimize the regret?* To start capturing the intuition, consider the simplest setting with $d = 2$, $k_1 = k_2 = 2$, $a_1^* = a_2^* = 1$, $\mu_{1,1} = \mu_{2,1} = 1$ and $\mu_{1,2} = \mu_{2,2} = y \in (0, 1)$. To satisfy Constraint (8.6), we have to guarantee $L_{1,2} = L_{2,2} = 2\sigma^2(1 - y)^{-2}$ (in the solution the constraint is satisfied with equality) and we have at our disposal 4 action vectors $\mathcal{A} = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$. We can satisfy the constraint in two ways:[5]

(i) playing action $(2, 2)$ (i.e., with both suboptimal components) for a proportion of $2\sigma^2(1 - y)^{-2}$ times, suffering $1 - y^2$ instantaneous regret;

(ii) playing actions $(1, 2)$ and $(2, 1)$ (i.e., with one suboptimal component) for a proportion of $2\sigma^2(1 - y)^{-2}$ *each*, suffering $1 - y$ instantaneous regret;

It is simple to convince that (i) is the choice that minimizes the cumulative regret. Indeed, for $y \in (0, 1)$, we have:

$$\underbrace{2\sigma^2(1 - y)^{-2}(1 - y^2)}_{\text{case } (i)} \leqslant \underbrace{4\sigma^2(1 - y)^{-2}(1 - y)}_{\text{case } (ii)} . \tag{8.8}$$

This intuitive reasoning can be extended to the general case. To this end, let us define the *sorting functions* $\pi_i : [\![k_i]\!] \to [\![k_i]\!]$ for every $i \in [\![d]\!]$ as any bijective function such that:

$$\mu_{i,\pi_i(1)} \leqslant \cdots \leqslant \mu_{i,\pi_i(k_i-1)} \leqslant \mu_{i,\pi_i(k_i)} = \mu_i^* .$$

We claim that in the optimal arrangement the action components need to *coordinate* as illustrated in Figure 8.1. For every dimension $i \in [\![d]\!]$ (row), we sort the action components in non-decreasing order of $\mu_{i,j}$ according to the sorting function π_i. To every $j \in [\![k_i - 1]\!]$, an interval of length $L_{i,j}$ is

[5] Any mix between (i) and (ii) is clearly suboptimal.

associated corresponding to the proportion of pull. Now, we combine the different rows to obtain the "active action vector" (represented by different colors) made by the corresponding action components. Each active action vector will be pulled for a proportion (the colored vertical slices) depending on the $L_{i,j}$ values of the corresponding components. Notice that we can have at most $\sum_{i \in [\![d]\!]} k_i - 1$ active action vectors and the total proportion of the pulls (the width of the full table in Figure 8.1) is given by $M := \max_{i \in [\![d]\!]} \sum_{j \in [\![k_i - 1]\!]} L_{i,j}$. To formally characterize the solution, we introduce, for every $i \in [\![d]\!]$ and $l \in [\![k_i - 1]\!]$, the variables $M_{i,l} := \sum_{l' \in [\![l]\!]} L_{i,\pi_i(l')}$ and $M_{i,k_i} = +\infty$ as the cumulative proportion of pulls of the action components more suboptimal than $(i, \pi_i(l))$, i.e., fixing a row i, the position of the black vertical lines in Figure 8.1 sorted from left to right. Let us define the sorting function $\boldsymbol{\pi} : [\![K]\!] \rightarrow \bigcup_{i \in [\![d]\!]} (\{i\} \times [\![k_i]\!])$, where $K = \sum_{i \in [\![d]\!]} k_i$, as any bijection such that:

$$M_{\boldsymbol{\pi}(1)} \leqslant \cdots \leqslant M_{\boldsymbol{\pi}(K-d)},$$

with the convention $M_{\boldsymbol{\pi}(0)} = 0$, i.e., the position in which we move from one vertical slice to the next one in Figure 8.1 sorted from left to right. For every $\ell \in [\![K]\!]$, we define the active action vector as $\boldsymbol{\alpha}_\ell = (j_{1,\ell}, \ldots, j_{d,\ell})^\top \in \mathcal{A}$ where:

$$j_{i,\ell} := \pi_i^{-1} \left(\arg\max_{l \in [\![k_i]\!]} \{ M_{i,l} \geqslant M_{\boldsymbol{\pi}(\ell)} \} \right).$$

This allows us to prove the following result.

Theorem 8.3.4 (Instance-Dependent Lower Bound (Explicit)). *Let $\underline{C}(\underline{\boldsymbol{\nu}})$ be the solution of the optimization problem of Theorem 8.3.3. It holds that:*

$$\underline{C}(\underline{\boldsymbol{\nu}}) = \sum_{\ell=1}^{K-d} \left(M_{\boldsymbol{\pi}(\ell)} - M_{\boldsymbol{\pi}(\ell-1)} \right) \Delta_{\boldsymbol{\alpha}_\ell},$$

that can be computed in $\mathcal{O}(\sum_{i \in [\![d]\!]} k_i \log k_i)$.

Proof Sketch. We generalize Equation (8.8) with the *rearrangement inequality* for integrals (Luttinger and Friedberg, 1976), the continuous version of the more known rearrangement inequality for sequences (Hardy et al., 1952). □

8.4 A Worst-Case Optimal Algorithm

In this section, we present an *optimistic any-time* regret minimization algorithm for the FRB setting. `Factored Upper Confidence Bound`

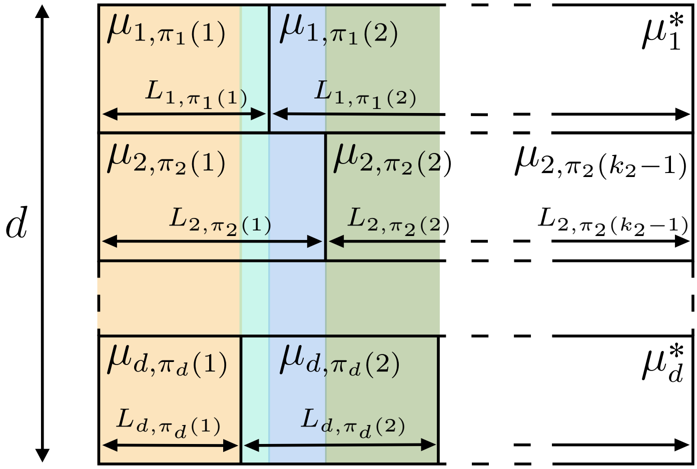


Figure 8.1: *Efficient solution to the LP presented in Theorem 8.3.3.*

Algorithm 8.1: F-UCB.

```
  Input : Exploration Parameter $\alpha$, Subgaussian proxy $\sigma$,
          Action component size $k_i$, $\forall i \in [\![d]\!]$
1 Initialize $N_{i,a_i}(0) \leftarrow 0$, $\widehat{\mu}_{i,a_i}(0) \leftarrow 0$ $\forall a_i \in [\![k_i]\!]$, $i \in [\![d]\!]$
2 for $t \in [\![T]\!]$ do
3     Select $\mathbf{a}(t) \in \operatorname*{arg\,max}_{\mathbf{a}=(a_1,\dots a_d)^\intercal \in \mathcal{A}} \prod_{i \in [\![d]\!]} \mathrm{UCB}_{i,a_i}(t)$ where
        $\mathrm{UCB}_{i,a_i}(t) = \widehat{\mu}_{i,a_i}(t-1) + \sigma\sqrt{\frac{\alpha \log t}{N_{i,a_i}(t-1)}}$
4     Play $\mathbf{a}(t)$ and observe $\mathbf{x}(t) = (x_1(t), \dots, x_d(t))^\intercal$
5     Update $\widehat{\mu}_{i,a_i(t)}(t)$ and $N_{i,a_i(t)}(t)$ for every $i \in [\![d]\!]$
6 end
```

(F-UCB), whose pseudo-code is reported in Algorithm 8.1, is based on the idea of running a UCB-like exploration (Auer et al., 2002a) *independently* for every dimension $i \in [\![d]\!]$ and estimate the expected observation μ_{i,a_i} for every action component $a_i \in [\![k_i]\!]$.

The algorithm requires as input the number of action components k_i for every $i \in [\![d]\!]$, the exploration parameter $\alpha > 2$, and the subgaussian proxy σ. After initializing the variables to keep track of the number of pulls $N_{i,a_i}(t)$ and the sample mean $\widehat{\mu}_{i,a_i}(t)$ for all action components (line 1), the algorithm starts the learner-environment interaction. At every round $t \in [\![T]\!]$, F-UCB computes the optimistic action, i.e., the action $\mathbf{a}(t)$ maximizing the optimistic index:

$$\mathbf{a}(t) \in \operatorname*{arg\,max}_{\mathbf{a}=(a_1,\dots,a_d)^\intercal \in \mathcal{A}} \prod_{i \in [\![d]\!]} \mathrm{UCB}_{i,a_i}(t),$$

where $\widehat{\mu}_{i,a_i}(t)$ is the empirical mean of the observations for the i^{th} component of the observation vector determined by the action component a_i, and $N_{i,a_i}(t)$ is the number of times the corresponding component of the action vector has been played (line 3). Then, the algorithm plays it and observes the d-dimensional observation vector $\mathbf{x}(t) = (x_1(t), \ldots, x_d(t))^{\mathrm{T}}$ (line 4). The observation vector is used to incrementally update the sample means of *all* action components involved and the related counters (lines 5). Finally, the algorithm reduces to UCB1 when $d = 1$.

F-UCB enjoys a *time complexity* of $\mathcal{O}(T \sum_{i \in [\![d]\!]} k_i)$ and a *space complexity* of $\mathcal{O}(\sum_{i \in [\![d]\!]} k_i)$. Indeed, at every round $t \in [\![T]\!]$, we need to recompute the index $\text{UCB}_{i,a_i}(t)$ for all $\sum_{i \in [\![d]\!]} k_i$ action components (at least the bonus changes at every round). Note that the computation of the optimistic action is not combinatorial since the optimization can be performed *independently* for every dimension $i \in [\![d]\!]$.

8.4.1 Worst-Case Regret Analysis

In this section, we provide the worst-case regret analysis of F-UCB as summarized in the following result.

Theorem 8.4.1 (Worst-Case Upper Bound for F-UCB)**.** *For any FRB $\underline{\boldsymbol{\nu}}$,* F-UCB *with $\alpha > 2$ suffers an expected regret bounded as:*

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant 4\sigma \sum_{i \in [\![d]\!]} \sqrt{\alpha k_i T \log T} + g(\alpha) \sum_{i \in [\![d]\!]} k_i,$$

where $g(\alpha) = \widetilde{\mathcal{O}}\left((\alpha - 2)^{-2}\right)$.[6]
In particular, if $k_i =: k$, for every $i \in [\![d]\!]$, we have:

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant \widetilde{\mathcal{O}}(\sigma d \sqrt{kT}).$$

Proof Sketch. Under a suitable "good event", we have that $\mu_{i,a_i} \leqslant \text{UCB}_{i,a_i}(t)$ for every $i \in [\![d]\!]$, $a_i \in [\![k_i]\!]$, and $t \in [\![T]\!]$. Thus, the instantaneous regret is bounded as:

$$\begin{aligned}\prod_{i \in [\![d]\!]} \mu_i^* - \prod_{i \in [\![d]\!]} \mu_{i,a_i(t)} &= \sum_{l \in [\![d]\!]} \prod_{i \in [\![l-1]\!]} \underbrace{\mu_i^*}_{\in[0,1]} \underbrace{\left(\mu_l^* - \mu_{l,a_l(t)}\right)}_{\leqslant \text{UCB}_{i,a_i(t)}(t) - \mu_{l,a_l(t)}} \prod_{i \in [\![l+1,d]\!]} \underbrace{\mu_{i,a_i(t)}}_{\in[0,1]} \\ &\leqslant \sum_{l \in [\![d]\!]} \left(\text{UCB}_{l,a_l(t)}(t) - \mu_{l,a_l}\right),\end{aligned}$$

[6] The complete expression is reported in the proof.

where the first line is obtained by summing and subtracting all mixed terms $\prod_{i \in [\![l]\!]} \mu_i^* \prod_{i \in [\![l+1,d]\!]} \mu_{i,a_i(t)}$ and the second by optimism $\mu_l^* \leqslant \mathrm{UCB}_{l,a_l^*}(t) \leqslant \mathrm{UCB}_{l,a_l(t)}(t)$. □

Comparing the upper bound of Theorem 8.4.1 with the lower bound in Theorem 8.3.1, we realize that the dependence on the learning horizon T is tight up to logarithmic factors (just like UCB1) and the dependence on the number of action components k_i, the number of dimensions d, and the subgaussian proxy σ are tight up to constant factors.

It is worth comparing our results with the ones that could be obtained by applying literature algorithms to our FRB setting. As already mentioned in Section 8.3, although each intermediate observation $x_i(t)$ is σ^2-subgaussian, their product $r(t)$, i.e., the reward, is not in general. This prevents, for instance, the application of UCB1 which assumes subgaussian (or bounded) reward. Precisely, for $d = 2$, the reward $r(t) = x_1(t)x_2(t)$ is a *subexponential* random variable, a scenario that can be still approached with the standard sample mean estimator but leveraging the Bernstein's concentration bound (Boucheron et al., 2013). However, for $d \geqslant 3$, as shown in Lemma D.2.1, the reward $r(t)$ does not admit a moment-generating function and, consequently, displays a *heavy-tailed* behavior (Bubeck et al., 2013). Nevertheless, the reward $r(t)$ random variable maintains a finite variance bounded by $\overline{\sigma}^2 = (1+\sigma^2)^d - 1$ (see Lemma D.2.2). This enables the application of algorithms designed for heavy-tailed bandits, such as Robust-UCB (Bubeck et al., 2013), able to handle generic distributions with finite variance, by resorting to estimators other than the sample mean. It is easy to verify that by considering the *Median of Means* estimator (Bubeck et al., 2013), we obtain a regret upper bound in the order of $\widetilde{\mathcal{O}}\left(\overline{\sigma}\sqrt{\prod_{i \in [\![d]\!]} k_i T}\right)$. This result is in line with the discussion in Section 8.3 and, clearly, not optimal. Indeed, the dependence on the product $\prod_{i \in [\![d]\!]} k_i \gg \sum_{i \in [\![d]\!]} k_i$ is because Robust-UCB does not exploit the factored property of the FRB setting. Furthermore, the dependence on $\overline{\sigma} = \sqrt{(1+\sigma^2)^d - 1} \geqslant \sigma$ is justified by the fact that the intermediate observations are ignored. Finally, the analysis of Factored Bandit TEA (Zimmert and Seldin, 2018) cannot be adapted to our setting since, as already mentioned, the subgaussian noise is applied to the final reward only.

8.4.2 Instance-Dependent Upper Bound

In this section, we provide the analysis of the instance-dependent regret upper bound for the `F-UCB` algorithm. The following theorem summarizes the result.

Theorem 8.4.2 (Instance-Dependent Upper Bound for `F-UCB`)**.** *For a given FRB* $\underline{\boldsymbol{\nu}}$*,* `F-UCB` *with* $\alpha > 2$ *suffers an expected regret bounded as:*

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant \overline{C}(\texttt{F-UCB}, \underline{\boldsymbol{\nu}}),$$

where $\overline{C}(\textit{F-UCB}, \underline{\boldsymbol{\nu}})$ *is defined as the solution to the following optimization problem (where* $g(\alpha) = \widetilde{\mathcal{O}}((\alpha-2)^{-2})$*):*

$$\max_{(N_{\mathbf{a}})_{\mathbf{a}\in\mathcal{A}}} \sum_{\mathbf{a}\in\mathcal{A}\setminus\{\mathbf{a}^*\}} N_{\mathbf{a}}\Delta_{\mathbf{a}} \tag{8.9}$$

$$\text{s.t.} \quad N_{i,j} = \sum_{\substack{\mathbf{a}\in\mathcal{A}\setminus\{\mathbf{a}^*\} \\ a_i=j}} N_{\mathbf{a}}, \qquad \forall i \in [\![d]\!],\ j \in [\![k_i]\!]\setminus\{a_i^*\} \tag{8.10}$$

$$N_{i,j} \leqslant \frac{4\alpha\sigma^2 \log T}{\Delta_{i,j}^2} + g(\alpha), \qquad \forall i \in [\![d]\!],\ j \in [\![k_i]\!]\setminus\{a_i^*\} \tag{8.11}$$

$$\sum_{\mathbf{a}\in\mathcal{A}} N_{\mathbf{a}} = T \tag{8.12}$$

$$N_{\mathbf{a}} \geqslant 0, \qquad \forall \mathbf{a} \in \mathcal{A} \tag{8.13}$$

The derivation of the LP in Theorem 8.4.2 follows a similar rationale as that of the instance-dependent lower bound of Theorem 8.3.3. Since `F-UCB` runs an optimistic UCB strategy *independent* for every action component, we can derive an upper bound on the expected number of pulls for every $i \in [\![d]\!]$ and $j \in [\![k_i]\!]\setminus\{a_i^*\}$ (denoted with $N_{i,j}$ in the LP):

$$\mathbb{E}[N_{i,j}(T)] \leqslant \frac{4\alpha\sigma^2 \log T}{\Delta_{i,j}^2} + g(\alpha),$$

generating Constraint (8.11), that, since the problem involves a maximization, will be satisfied with equality. To relate the expected number of pulls $\mathbb{E}[N_{\mathbf{a}}(T)]$ of the action vectors $\mathbf{a} \in \mathcal{A}\setminus\{\mathbf{a}^*\}$ (denoted with $N_{\mathbf{a}}$ in the LP) with the ones of the action components $\mathbb{E}[N_{i,j}(T)]$, we use the same argument of Theorem 8.3.3, producing Constraint (8.10). Similarly to the LP in Theorem 8.3.3, the problem is made of $\prod_{i\in[\![d]\!]} k_i + \sum_{i\in[\![d]\!]} k_i - d$ variables and $1 + \prod_{i\in[\![d]\!]} k_i + 2\sum_{i\in[\![d]\!]} k_i - 2d$ constraints. We now provide an explicit solution to a *relaxation* of the LP of Theorem 8.4.2.

Corollary 8.4.3 (Explicit Instance-Dependent Upper Bound for `F-UCB`). *For a given FRB $\underline{\boldsymbol{\nu}}$,* `F-UCB` *with $\alpha > 2$ suffers an expected regret bounded by:*

$$\begin{aligned}\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] &\leqslant \overline{C}(\texttt{F-UCB}, \underline{\boldsymbol{\nu}}) \\ &\leqslant 4\alpha\sigma^2 \log T \sum_{i \in [\![d]\!]} \mu^*_{-i} \sum_{j \in [\![k_i]\!] \setminus \{a^*_i\}} \Delta^{-1}_{i,j} + g(\alpha) \sum_{i \in [\![d]\!]} k_i,\end{aligned}$$

where $\mu^*_{-i} = \prod_{l \in [\![d]\!] \setminus \{i\}} \mu^*_l \leqslant 1$ for every $i \in [\![d]\!]$.

Proof Sketch. The result is based on providing a *relaxation* of the objective function of the optimization problem in Theorem 8.4.2, which is based on the following bound on the suboptimality gaps of the action vector $\mathbf{a} = (a_1, \dots, a_d)^{\mathrm{T}}$ in terms of the suboptimality gaps of the action components:

$$\Delta_{\mathbf{a}} \leqslant \sum_{i \in [\![d]\!]} \Delta_{i,a_i} \mu^*_{-i}.$$

This allows to upper bound the objective function as:

$$\sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}*\}} N_{\mathbf{a}} \Delta_{\mathbf{a}} \leqslant \sum_{i \in [\![d]\!]} \mu^*_{-i} \sum_{j \in [\![k_i]\!] \setminus \{a^*_i\}} N_{i,j} \Delta_{i,j}.$$

By Constraint (8.11) to upper bound N_{i,a_i}, we get the result. Alternatively, we can drop the constraint $\sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}*\}} N_{\mathbf{a}} = T$ and use a *rearrangement inequality* (Hardy et al., 1952) to upper bound the objective function. □

It is worth comparing this instance-dependent regret upper bound of `F-UCB` with the one achievable with an algorithm for heavy-tailed bandits, such as `Robust-UCB` (Bubeck et al., 2013). Our result of Corollary 8.4.3 is of order (neglecting the dependence on α and on constants):

$$\mathcal{O}\left(\sigma^2 \sum_{i \in [\![d]\!]} \mu^*_{-i} \sum_{j \in [\![k_i]\!] \setminus \{a^*_i\}} \frac{\log T}{\Delta_{i,j}}\right). \tag{8.14}$$

Instead, `Robust-UCB`, for instance with the *Median of Means* estimator, is characterized by the following instance-dependent regret of order (neglecting constants):

$$\mathcal{O}\left(\overline{\sigma}^2 \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}*\}} \frac{\log T}{\Delta_{\mathbf{a}}}\right). \tag{8.15}$$

where $\overline{\sigma}^2 = (1 + \sigma^2)^d - 1 \geqslant \sigma^2$. It is simple to observe that Equation (8.15) is larger than Equation (8.14). Indeed, consider the subset of action vectors in which exactly one component is not optimal, i.e., $\mathcal{A}^\circ = \bigcup_{i \in [\![d]\!]} \mathcal{A}_i^\circ$ where $\mathcal{A}_i^\circ := \{\mathbf{a} \in \mathcal{A} : a_i \neq a_i^*, a_j = a_j^*, j \in [\![d]\!] \backslash \{i\}\}$. We observe that for every $\mathbf{a} \in \mathcal{A}_i^\circ$, the action vector suboptimality gap is related with equality to that of the suboptimal component:

$$\Delta_{\mathbf{a}} = \prod_{l \in [\![d]\!]} \mu_l^* - \mu_{i,a_i} \prod_{l \in [\![d]\!] \backslash \{i\}} \mu_l^* = \mu_{-i}^* \Delta_{i,a_i}.$$

This allows the conclusion of the following as desired:

$$\sum_{\mathbf{a} \in \mathcal{A} \backslash \{\mathbf{a}^*\}} \frac{\log T}{\Delta_{\mathbf{a}}} \geqslant \sum_{\mathbf{a} \in \mathcal{A}^\circ} \frac{\log T}{\Delta_{\mathbf{a}}} \geqslant \sum_{i \in [\![d]\!]} \mu_{-i}^* \sum_{j \in [\![k_i]\!] \backslash \{a_i^*\}} \frac{\log T}{\Delta_{i,j}}.$$

Finally, let us compare Corollary 8.4.3 with the instance-dependent regret upper bound of the `Factored Bandit TEA` algorithm (Zimmert and Seldin, 2018), although the noise model is different. Theorem 2 of (Zimmert and Seldin, 2018) provides a bound of order (neglecting constants):

$$\mathcal{O}\left(\kappa \sum_{i \in [\![d]\!]} \sum_{j \in [\![k_i]\!] \backslash \{a_i^*\}} \frac{\log(T \log T) + \log \frac{\log(T \log T)}{\Delta_{i,j}^2}}{\Delta_{i,j}} \right),$$

where κ is such that $\Delta_{\mathbf{a}} \leqslant \kappa \sum_{i \in [\![d]\!]} \Delta_{i,a_i}$. Thus, we can set $\kappa = \max_{i \in [\![d]\!]} \mu_{-i}^*$. This result is slightly worse than ours because of the presence of the larger κ and the additional $\log \log T$ and $\log(1/\Delta_{i,j}^2)$ terms.

Remark 8.4.1 (About Instance-Dependent Optimality of `F-UCB`). *We argue about the instance-dependent optimality of `F-UCB`. To this end, we focus on a specific FRB instance with generic $d > 1$ and $k_1 = \cdots = k_d = 2$. We consider Gaussian intermediate observations with expected values $\mu_{i,1} = 1$ and $\mu_{i,2} = 1 - \Delta$ where $\Delta \in (0,1)$ for every $i \in [\![d]\!]$. By applying Theorems 8.3.3 and 8.4.2, we deduce that for $T \to +\infty$, we have the lower bound (left) and the `F-UCB` upper bound (right) on the number of pulls of each suboptimal action component $i \in [\![d]\!]$ bounded as:*

$$\frac{\mathbb{E}[N_{i,2}(T)]}{\log T} \geqslant \frac{2\sigma^2}{\Delta^2} \quad \textit{and} \quad \frac{\mathbb{E}[N_{i,2}(T)]}{\log T} \leqslant \frac{4\alpha\sigma^2}{\Delta^2}.$$

Thanks to Theorem 8.3.4 and Corollary 8.4.3, we can compute $\underline{C}(\underline{\boldsymbol{\nu}})$ and upper bound $\overline{C}(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})$:

$$\underline{C}(\underline{\boldsymbol{\nu}}) = \frac{2\sigma^2(1-(1-\Delta)^d)}{\Delta^2} \quad \textit{and} \quad \frac{\overline{C}(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})}{\log T} \leqslant \frac{4d\alpha\sigma^2}{\Delta}.$$

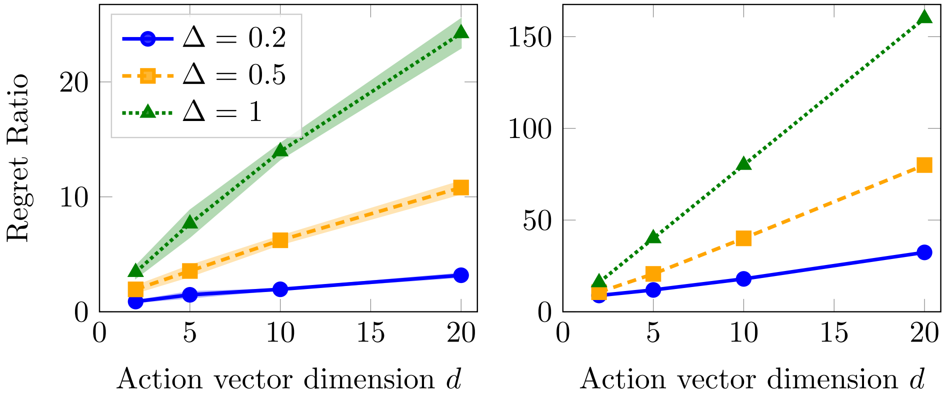


Figure 8.2: *Ratio between the* actual *regret of* F−UCB *and the instance-dependent lower bound (left) and ratio between the regret upper bound and the instance-dependent lower bound (Equation 8.16) (right), for different values of* d *(*5 *runs, mean* $\pm$ 2*std).*

It is immediate to realize the following extreme behaviors:

$$\frac{\overline{C}(\texttt{F-UCB},\underline{\boldsymbol{\nu}})}{\underline{C}(\underline{\boldsymbol{\nu}})\log T} \leqslant \frac{2d\alpha\Delta}{1-(1-\Delta)^d} \to \begin{cases} 2\alpha & \Delta\to 0 \\ 2\alpha d & \Delta\to 1 \end{cases}. \tag{8.16}$$

This suggests that for sufficiently large $\Delta \approx 1$*,* F−UCB *can perform significantly worse than the lower bound, introducing an additional dependence on* d*. Instead, for sufficiently small* $\Delta \approx 0$*,* F−UCB *can match the lower bound up to constant factors.*[7] *Clearly, we conducted this analysis employing an* upper bound *to the expected regret of* F−UCB*, which might, in principle, be affected by some analysis artifacts, making it not tight. In Figure 8.2, we compare the ratio between the* actual *regret obtained by running* F−UCB *(5 runs) on the proposed FRB example and the instance-dependent lower bound (left) with the ratio between the upper bound and the instance-dependent lower bound computed in Equation* (8.16) *(right). We clearly observe that, although the* y*-scales are different, the behavior confirms a linear dependence of the actual regret of* F−UCB *on the number of dimensions of the action vector* d*.*

8.5 Optimal Asymptotic Instance-Dependent Algorithm

In this section, we provide an algorithm that matches the derived instance-dependent lower bound (Theorem 8.3.3) in the asymptotic regime. The

[7] Indeed, when the suboptimality gaps are close to 0, the instantaneous regret $\prod_{i\in[\![d]\!]}\mu_i^* - \prod_{i\in[\![d]\!]}\mu_{i,a_i(t)}$ approaches the sum of the regrets on each action component $\sum_{i\in[\![d]\!]}(\mu_i^* - \mu_{i,a_i(t)})$.

algorithm, named `Factored Track` (`F-Track`), whose pseudocode is reported in Algorithm 8.2, is based on the idea of *tracking the lower bound* (Lattimore and Szepesvari, 2017).

The rationale behind the algorithm is that if we want to match the instance-dependent lower bound, we need to properly *coordinate* the choice of the action vectors $\mathbf{a} \in \mathcal{A}$, given that we have a lower bound on the minimum number of pulls for the action components (i, j) (Theorem 8.3.3). To impose such a structure we must plan in advance our sequence of action vector choices. We devise an algorithm composed of three phases: *warm-up*, *success*, and *recovery*. In the warm-up phase, the algorithm pulls some action vectors in such a way that each action component is pulled at least N_0 times, i.e., $N_{i,j} \geqslant N_0$ (line 3). This can be achieved by round-robing the action components values j of each component i, leading to a number of pulls in the warm-up phase equal to $T_{\text{warm-up}} = N_0 \max_{i \in [\![d]\!]} k_i$. We use these samples to estimate the expected values $\widehat{\mu}_{i,j}(T_{\text{warm-up}})$ and define the confidence interval threshold ϵ_T. Then, we use these values as if they were the true ones $\mu_{i,j}$ to compute the suboptimality gaps $\widehat{\Delta}_{i,j} := \max_{j' \in [\![k_i]\!]} \widehat{\mu}_{i,j'}(T_{\text{warm-up}}) - \widehat{\mu}_{i,j}(T_{\text{warm-up}})$ (line 7) and, using them, the number of pulls (line 8):

$$\widehat{N}_{i,j} = \frac{2\sigma^2 f_T(1/T)}{\widehat{\Delta}^2_{i,j}}, \quad \forall j \in [\![k_i]\!],\ i \in [\![d]\!]$$

where for every $\delta \in (0, 1)$:

$$f_T(\delta) := \left(1 + \frac{1}{\log T}\right)\left(c \log\log T + \log\left(\frac{1}{\delta}\right)\right),$$

where c is a universal constant and, with them, we compute the number of pulls for every action vector $\widehat{N}_{\mathbf{a}}$ by solving the optimization problem in Theorem 8.3.3 (line 9). It is worth noting that $f_T(1/T) \approx \log T$ and this form is needed for technical reasons to guarantee that the confidence bounds hold. In the success phase, until we run out of the rounds $t \leqslant T$, we track the lower bound by pulling in a round-robin fashion all arms whose number of pulls $N_{\mathbf{a}}(t) < \widehat{N}_{\mathbf{a}}$ (line 11). If we realize that the estimated expected reward $\widehat{\mu}_{i,j}(t-1)$ are too far from the ones estimated at the end of the warm-up phase $\widehat{\mu}_{i,a_i}(T_{\text{warm-up}})$ based on the threshold ϵ_T, we move to the recovery phase (line 10). In this phase, we play `F-UCB` until the end of the rounds discarding all the data collected so far (line 13).

The following result shows that `F-Track` asymptotically matches the lower bound for a proper choice of N_0 and ϵ_T.

Algorithm 8.2: F-Track.

```
Input : Warm-up sample size $N_0$, Threshold $\epsilon_T$, Action component size $k_i$, $\forall i \in [\![d]\!]$,
$t \leftarrow 1$
while $\min_{i \in [\![d]\!]} \min_{j \in [\![k_i]\!]} N_{i,j}(t) < N_0$ do
    Pull action vector $\mathbf{a}(t)$ with $a_i(t) = (t-1) \mod k_i + 1$ for all $i \in [\![d]\!]$,
    $t \leftarrow t + 1$
end
$T_{\text{warm-up}} \leftarrow t - 1$
Estimate the suboptimality gaps $\forall i \in [\![d]\!],\ j \in [\![k_i]\!]$ :
    $\widehat{\Delta}_{i,j} := \max_{j' \in [\![k_i]\!]} \widehat{\mu}_{i,j'}(T_{\text{warm-up}}) - \widehat{\mu}_{i,j}(T_{\text{warm-up}})$
Compute the number of pulls $\widehat{N}_{i,j} = 2\sigma^2 f_T(1/T)\widehat{\Delta}_{i,j}^{-2}$ for every action component
  $i \in [\![d]\!]$ and $j \in [\![k_i]\!]$
Compute the number of pulls $\widehat{N}_{\mathbf{a}}$ for every action vector $\mathbf{a} \in \mathcal{A}$ by solving the LP in
  Theorem 8.3.3
while $t \leqslant T$ and $\max_{i \in [\![d]\!], j \in [\![k_i]\!]} |\widehat{\mu}_{i,j}(T_{\text{warm-up}}) - \widehat{\mu}_{i,j}(t-1)| \leqslant 2\epsilon_T$ do
    Pull action vector $\mathbf{a}(t) \in \arg\min\{N_{\mathbf{a}}(t) : \mathbf{a} \in \mathcal{A}$ and $N_{\mathbf{a}}(t) \leqslant \widehat{N}_{\mathbf{a}}\}$, $t \leftarrow t + 1$
end
Discard all data and play F-UCB until $t = T$
```

Theorem 8.5.1 (Instance-Dependent Upper Bound for F-Track)**.** *For any FRB $\underline{\boldsymbol{\nu}}$,* F-Track *run with:*

$$N_0 = \left\lceil \sqrt{\log T} \right\rceil \quad \textit{and} \quad \epsilon_T = \sqrt{\frac{2\sigma^2 f_T(1/\log T)}{N_0}},$$

suffers an expected regret of:

$$\limsup_{T \to +\infty} \frac{\mathbb{E}\left[R_T(\texttt{F-Track}, \underline{\boldsymbol{\nu}})\right]}{\log T} = \underline{C}(\underline{\boldsymbol{\nu}}).$$

8.6 Numerical Simulations

In this section, we provide numerical simulations to validate the proposed solutions. First, in Section 8.6.1, we validate F-UCB against bandit baselines in several scenarios. Then, in Section 8.6.2, we compare the two algorithms we propose (i.e., F-UCB and F-Track) in different scenarios to highlight their peculiarities. Finally, in Section 8.6.3, we evaluate the proposed algorithms' behavior in the case in which the noise affecting intermediate observations is partially correlated. The code used to run the experiments in this section can be found at https://github.com/marcomussi/FRB.

8.6.1 Comparison of F-UCB against Bandit Baselines

In this part, we show the effectiveness of F-UCB against bandit baselines.

Baselines The first baseline we consider is UCB1 (Auer et al., 2002a), which is designed for stochastic bandits. We consider the anytime version of the algorithm, proposed by Bubeck (2010). Due to its characteristics, we expect it to perform in a comparable manner to F-UCB for $d = 1$, with its performance degrading as the dimensionality grows. As an additional baseline, we consider a *robust* version of UCB algorithm designed for heavy-tail (HT) distributions (Bubeck et al., 2013) considering the *Median of Means* estimator (RUCB-MoM). Due to the capability of this algorithm to handle non-subgaussian noise, we expect it to converge for any problem dimensionality, although at a slower rate. Finally, we consider the TEA algorithm, proposed by Zimmert and Seldin (2018). Since this algorithm provides theoretical guarantees for handling only subgaussian noise applied to the reward, we expect it to have a performance that degrades when $d > 1$. For all the baselines, we consider the values of the hyperparameters as prescribed in the respective original papers.

Setting For the sake of simplicity in the presentation of the results, we consider the scenario in which all the problem dimensions present the same number of actions (i.e., $k_1 = \dots = k_d =: k$). Moreover, we consider the setting in which the intermediate observations are drawn from Gaussian distributions with mean $\mu_{i,a_i(t)}$ for every action component $a_i(t)$ in position i of the action vector $\mathbf{a}$, formally $x_i(t) \sim \mathcal{N}(\mu_{i,a_i(t)}, \sigma^2), \ \forall i \in [\![d]\!]$. We consider values of $k \in [\![3, 5]\!]$, and values of $d \in [\![4]\!]$. We draw the expected values $\mu_{i,j}$ for $i \in [\![d]\!]$ and $j \in [\![k]\!]$ from a uniform distribution in the range $[0.7, 1]$. We fix a value of $\sigma = 0.1$. It is worth noting that the results in the following paragraph are not comparable among the different k and d, mostly for what concerns the comparison between different values of d. We evaluate the performances in terms of cumulative regret with $T = 10^4$, averaged over 50 trials.

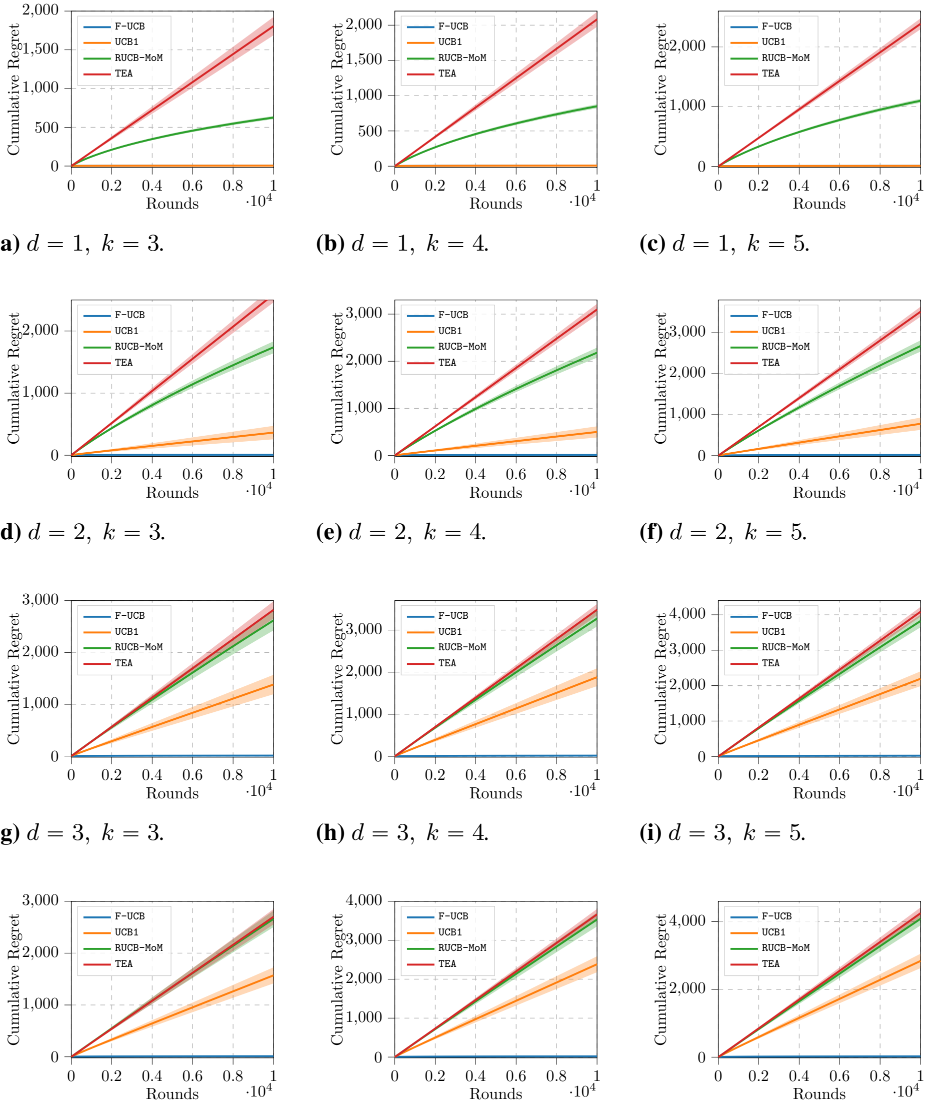
Results In Figure 8.3, we present the cumulative regret for the F-UCB algorithm and the other bandit baselines. The value of k increases with the columns, and the value of d increases with the rows of the figure. The following comments are valid for all the considered values of k, as no unexpected or relevant behaviors are present when we increase the number of actions for each action component. We observe that for $d = 1$, F-UCB achieves a cumulative regret that matches that of UCB1. This is expected, as F-UCB collapses to UCB1 for $d = 1$. RUCB-MoM achieves a sublinear regret, although higher than the previous algorithms, whereas TEA suffers a cumulative regret that is linear in the considered time horizon. The behavior changes for $d = 2$. F-UCB achieves a low cumulative regret. The cumulative regret of UCB1, instead, constantly increases over the time horizon. RUCB-MoM continues to achieve a sublinear regret, however it is higher, due to the increased cardinality of the equivalent action space and the incremented effect of the noise. The behavior of TEA remains the same as for $d = 1$. For $d \geqslant 3$, we observe a stabilization of the behavior. F-UCB manages to achieve a cumulative regret that scales well as d and k increase. UCB1 now suffers a linear regret, RUCB-MoM a sublinear regret worse with the increase of d, and TEA behaves as in the previous cases.

8.6.2 Comparison of F-UCB and F-Track

In this part, we provide numerical simulations intended to compare F-UCB and F-Track in different scenarios. As discussed in Remark 8.4.1 and shown Figure 8.2, the performances of F-UCB decrease when the number d of dimensions increases and when the suboptimality gaps are large. The goal of this part is to (i) verify once again this fact and (ii) observe if F-Track is able to mitigate such a phenomenon.

Setting We consider the scenario in which the number of arms is constant across all dimensions, i.e., $k_i = k, \forall i \in [\![d]\!]$. Given our goal to verify the algorithms' behavior over the action vector dimensionality d and the suboptimality gaps dimension, we fixed the other parameters. We consider a scenario in which we have $k = 2$ and observations affected Gaussian i.i.d. noise with $\sigma = 0.5$. We evaluate the two algorithms for $d \in \{2, 5, 10, 20, 30\}$. For what concerns the expected values, for all the dimensions, we enforce the first arm to be the best one, with expected value $\mu_{i,1} = \mu_i^* = 1, \forall i \in [\![d]\!]$. The suboptimal arms have all the same expected values $\mu_{i,2} = 1 - \Delta_{i,2}, \forall i \in [\![d]\!]$. Such a value $\Delta_{i,2}$ has been tested in the set $\Delta_{i,2} \in \{0.5, 0.7, 0.9\}$. We evaluate the performances in terms of regret, averaged over 10 runs with target time horizons $T \in [10^4, 10^5]$. We remark that F-UCB is an anytime algorithm and can be run once to obtain the entire curve of the cumulative regret. Instead, F-Track requires the knowledge of the horizon to compute the correct values of N_0 and ϵ_T. As such, we repeated the experiment for F-Track several times, each with a different time horizon up to the maximum T.

Results In Figure 8.4, we present the cumulative regret for F-UCB and F-Track in the above-mentioned setting. First, we observe that for small values of d (i.e., $d \in \{2, 5\}$), F-UCB outperforms F-Track for all the



(a) $d = 1,\ k = 3.$

(b) $d = 1,\ k = 4.$

(c) $d = 1,\ k = 5.$

(d) $d = 2,\ k = 3.$

(e) $d = 2,\ k = 4.$

(f) $d = 2,\ k = 5.$

(g) $d = 3,\ k = 3.$

(h) $d = 3,\ k = 4.$

(i) $d = 3,\ k = 5.$

(j) $d = 4,\ k = 3.$

(k) $d = 4,\ k = 4.$

(l) $d = 4,\ k = 5.$

Figure 8.3: *Performance of* F-UCB*,* UCB1*,* RUCB-MoM *and* TEA *considering* $k \in [\![3, 5]\!]$ *and* $d \in [\![4]\!]$ *(50 runs, mean* ± *std).*

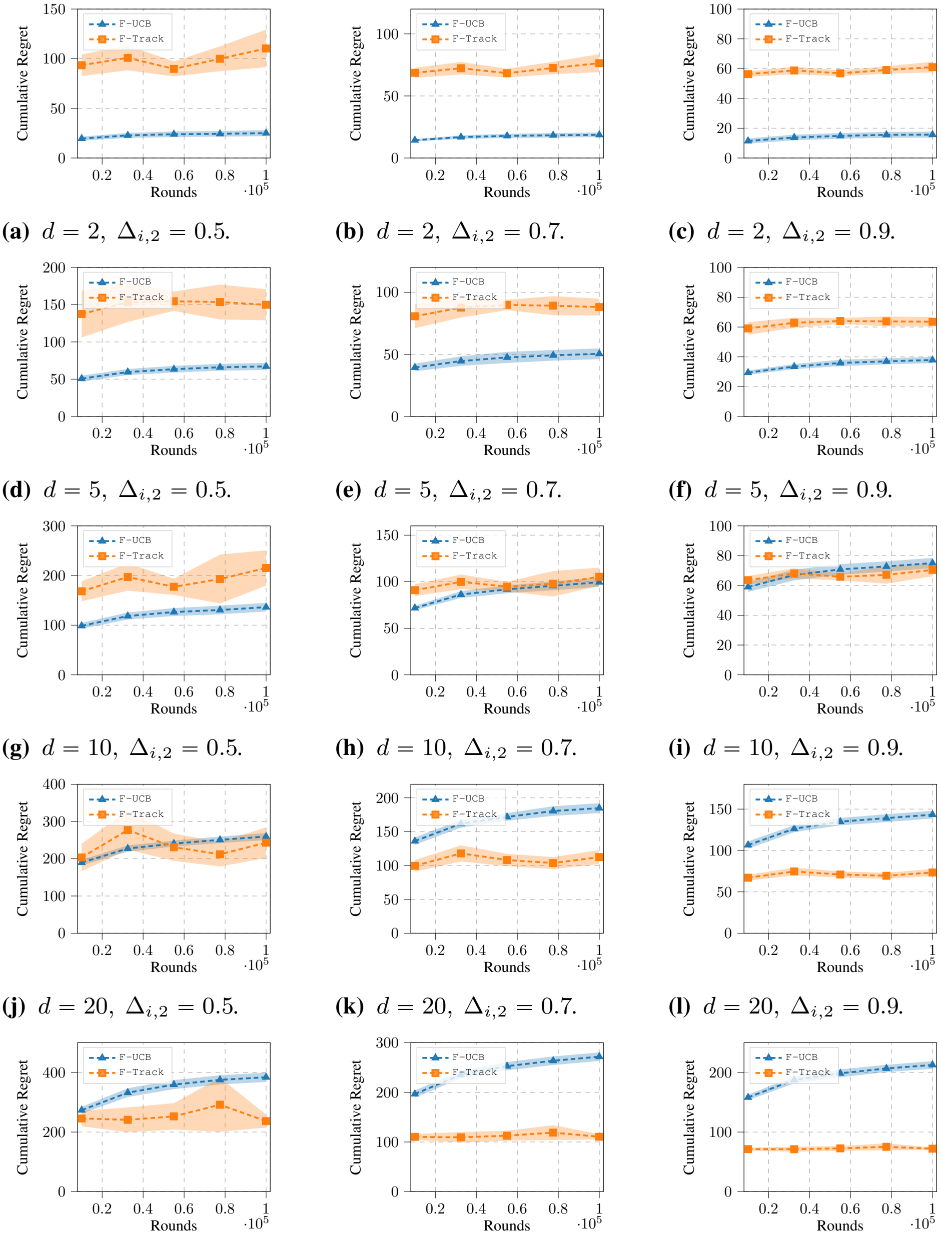
values of $\Delta_{i,2}$. This behavior is less evident when we move to $d = 10$, where the performances become comparable, with an advantage for F-UCB for smaller values of $\Delta_{i,2}$, while for larger value of the suboptimality gap, F-Track is better. The results turn in favor of F-Track when d becomes larger (i.e., $d \in \{20, 30\}$), and such an advantage further increases when $\Delta_{i,2}$ is large.

8.6.3 Robustness to Correlated Noise

In this part, we provide numerical simulations intended to compare F-UCB and F-Track when there is a correlation between the noises affecting the different dimensions. As discussed in Remark 8.3.1, in our setting, we require that the observations must be non-correlated. Otherwise, the problem cannot be factored properly given that, in general, if there is a correlation between the noises, we have that:

$$\mathbb{E}\left[\prod_{i\in[\![d]\!]} x_i(t)\right] \neq \prod_{i\in[\![d]\!]} \mathbb{E}\left[x_i(t)\right]. \tag{8.17}$$

Setting We consider the scenario in which the number of arms is constant across all dimensions, i.e., $k_i = k, \forall i \in [\![d]\!]$. We consider $k = 2$ and $d = 10$. For what concerns the expected values, for all the dimensions, we enforce the first arm to be the best one, with expected value $\mu_{i,1} = \mu_i^* = 1, \forall i \in [\![d]\!]$. The suboptimal arms have all the same expected values $\mu_{i,2} = 0.5, \forall i \in [\![d]\!]$. In order to evaluate the behavior of the algorithms in the presence of correlation in the noise of intermediate observations, we introduce a term $\alpha \in [0, 1]$ to control the interdependence of the intermediate observations. The additive noise applied to the observations $x_i(t)$ is defined as $\alpha\,\eta(t) + (1-\alpha)\epsilon_i(t)$, where $\eta(t), \epsilon_i(t) \sim \mathcal{N}(0, \sigma^2)$. The noise term $\eta(t)$ is applied to all the dimensions, whereas the $\epsilon_i(t)$ terms are individual and applied to the single dimensions $i \in [\![d]\!]$. Given this formulation, if $\alpha = 0$ the intermediate observations are independent, while if $\alpha = 1$, the intermediate observations are fully correlated. For values of $\alpha \in (0, 1)$, the noise term in the intermediate observations will comprise a correlated term and an independent term. We consider the case in which the Gaussian noise with $\sigma = 0.5$ (for both the independent and correlated components) affects *only action components* $a_i = 2$ (i.e., those with expected value $\mu_{i,2} = 0.5$) for $i \in [\![d]\!]$. We consider values of $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. We evaluate the performances in terms of cumulative regret averaged over 10 runs with target time horizons $T \in [10^4, 10^5]$.



(a) $d = 2,\ \Delta_{i,2} = 0.5$.

(b) $d = 2,\ \Delta_{i,2} = 0.7$.

(c) $d = 2,\ \Delta_{i,2} = 0.9$.

(d) $d = 5,\ \Delta_{i,2} = 0.5$.

(e) $d = 5,\ \Delta_{i,2} = 0.7$.

(f) $d = 5,\ \Delta_{i,2} = 0.9$.

(g) $d = 10,\ \Delta_{i,2} = 0.5$.

(h) $d = 10,\ \Delta_{i,2} = 0.7$.

(i) $d = 10,\ \Delta_{i,2} = 0.9$.

(j) $d = 20,\ \Delta_{i,2} = 0.5$.

(k) $d = 20,\ \Delta_{i,2} = 0.7$.

(l) $d = 20,\ \Delta_{i,2} = 0.9$.

(m) $d = 30,\ \Delta_{i,2} = 0.5$.

(n) $d = 30,\ \Delta_{i,2} = 0.7$.

(o) $d = 30,\ \Delta_{i,2} = 0.9$.

Figure 8.4: *Cumulative regret of* `F-UCB` *and* `F-Track` *considering* $k = 2$, $\sigma = 0.5$, $d \in \{2, 5, 10, 20, 30\}$, *and* $\Delta_{i,2} \in \{0.5, 0.7, 0.9\}$, $\forall i \in [\![d]\!]$ *(10 runs, mean* ± *2std).*

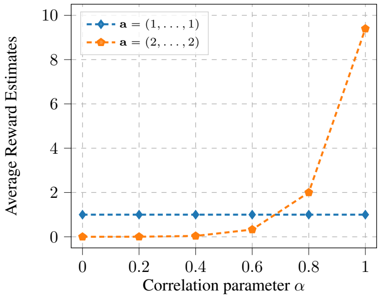


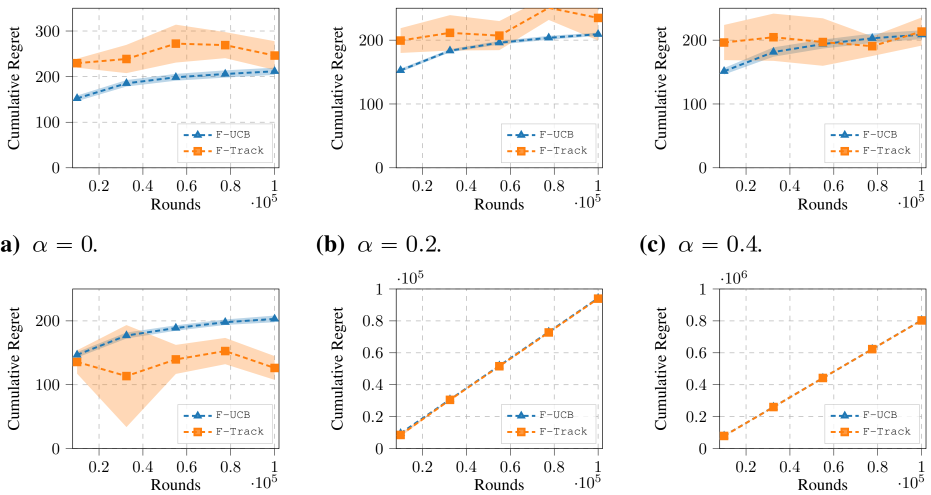
Figure 8.5: *Monte Carlo estimates of the expected values for the tested values of the correlation parameter* $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ *(*10^6 *Monte Carlo simulations).*

Results Before commenting on the results, we observe that the presence of correlated noise over action components $a_i = 2$ has the effect of changing the optimal vector action depending on the value of α. In Figure 8.5, we plot the value of the expected reward of the action vectors $(1, \dots, 1)$ and $(2, \dots, 2)$ estimated using 10^6 Monte Carlo simulations for the values of α under analysis. We consider just the two action vectors $(1, \dots, 1)$ and $(2, \dots, 2)$, given that all the other combinations of action components will give intermediate results (and are suboptimal). We first observe that, given that all the observations of the action vector $(1, \dots, 1)$ are not influenced by any noise, its expected reward is stable over α. On the other hand, for action vector $(2, \dots, 2)$, affected by noise, we see how as the correlation increases, the expected reward increases itself and overtakes the one of action vector $(1, \dots, 1)$.

Moving to the simulations, Figure 8.6 shows a comparison of the performances of `F-UCB` and `F-Track` when we vary correlation parameter α. First, we observe how the two algorithms present a consistent behavior over the different values of α. They are able to achieve satisfactory performances (i.e., sublinear regret) up to $\alpha = 0.6$. Then, the regret degenerates to linear. This is consistent with what we observed in Figure 8.5, as these algorithms look at the expected values of the single action components, but in this case, the noise correlation altered the optimal arm, which is no longer the one with the highest product of the expected observations.

8.7 Related Works

In this section, we discuss the related works from the *action structure* perspective and the works that present a *notion of factored structure*. Then, we compare the most significant related algorithms with our work from the



(a) $\alpha = 0$.

(b) $\alpha = 0.2$.

(c) $\alpha = 0.4$.

(d) $\alpha = 0.6$.

(e) $\alpha = 0.8$.

(f) $\alpha = 1$.

Figure 8.6: *Cumulative regret of* `F-UCB` *and* `F-Track` *considering* $k = 2$, $\sigma = 0.5$, $d = 5$, $\Delta_{i,2} = 0.5, \forall i \in [\![d]\!]$, *and correlation parameter* $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ *(*10 *runs, mean* $\pm$ 2*std).*

theoretical perspective.

Action Structure Originally, multi-armed bandit frameworks focused on independent arms with no inherent structure (Lai and Robbins, 1985). However, in recent decades, various bandit models with several kinds of structure have emerged, such as linear (Dani et al., 2008; Abbasi-Yadkori et al., 2011), Lipschitz (Agrawal, 1995; Magureanu et al., 2014) and unimodal (Yu and Mannor, 2011) bandits. These contributions aim to incorporate diverse forms of structure into the arms being considered. Combes et al. (2017) introduced a generalization of structured bandits, accommodating a wide range of structural concepts among arms. Their work offers a statistically efficient (at least in the general case) algorithm for handling generic structures, at the expense of solving a semi-infinite linear program at each time step. The necessity of choosing several actions at a time in a structured manner has been widely studied in the field of combinatorial bandits (Cesa-Bianchi and Lugosi, 2012; Kveton et al., 2015; Combes et al., 2015).

Notions of Factored Bandits Among the several kinds of structure, Zimmert and Seldin (2018) is the most similar to the work we propose from the point of view of the action structure, although the two works differ from the feedback perspective. Both works employ an action structure in which an action component a_i is selected for each problem dimension $i \in [\![d]\!]$. The action components are combined with a general function that obeys a *uniform identifiability* assumption under which the performance of each action vector can only improve when any action component is switched with the optimal one. However, in the work of Zimmert and Seldin (2018) the feedback comprises a single observation of the subgaussian reward $r(\mathbf{a}_t)$ applied to the aggregated expected reward, whereas, in our work, the feedback comprises one noisy observation for every action component. This peculiarity of our work implies that the reward obtained as the product over all the dimensions is not subgaussian anymore (Lemma D.2.1). (Zimmert and Seldin, 2018) generalizes (Katariya et al., 2017) to the case of more than two dimensions.

8.7.1 Comparison of the Theoretical Results

In Table 8.1, we summarize our setting with the one of Heavy-Tails Bandits (Bubeck et al., 2013) and the Factored Bandits (Zimmert and Seldin, 2018). We also analyze and compare both our solutions with `Robust-UCB` (Bubeck et al., 2013) and `TEA` (Zimmert and Seldin, 2018) from the instance-dependent point of view. Then, in Table 8.2 we compare worst-case lower and upper bounds from the worst-case perspective.

8.8 Discussion and Conclusions

In this chapter, we introduced the Factored-Reward Bandits, a novel setting to represent decision-making problems in which the learner is required to perform a set of actions, whose effects can be observed, and the reward is the product of those effects. We characterized the inherent complexity through worst-case and instance-dependent lower bounds, and we discussed the performances of current solutions. To address the regret minimization problem, we proposed two algorithms using the intermediate observations to reduce the complexity of learning in this setting. The first `F-UCB` is an optimistic solution that we proved minimax optimal (up to logarithmic factors). Such a solution deals with action components independently of the others and we have illustrated how, without coordination, we cannot reach instance-dependent optimality. To overcome this issue, we propose `F-Track`, an algorithm able to perform planning on the action components, and we proved its asymptotically instance-dependent optimality. As future lines of research, we plan to investigate the possibility of developing an algorithm able to guarantee both non-asymptotic instance-

Table 8.1: *Comparison with the instance-dependent guarantees of (Bubeck et al., 2013) and (Zimmert and Seldin, 2018).* †*This result holds for $T \to \infty$.* ‡*The authors consider $\sigma = 1$.*

	Setting Characteristics		Lower Bound	Upper Bound	Match			
	Factored Structure	Intermediate Feedback			σ	d	k	T
Robust-UCB (Bubeck et al., 2013)	✗	✗	$\Omega\left(\underline{\sigma}^2 \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}} \frac{\log T}{\Delta_{\mathbf{a}}}\right)$	$\mathcal{O}\left(\overline{\sigma}^2 \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}} \frac{\log T}{\Delta_{\mathbf{a}}}\right)$	✗	✓	✓	✓
TEA (Zimmert and Seldin, 2018)	✓	✗	$\Omega\left(\sum_{i \in [\![d]\!]} \sum_{j \in [\![k_i]\!] \setminus \{a_i^*\}} \frac{\log T}{\Delta_{i,j}}\right)^{\dagger}$	$\mathcal{O}\left(\sum_{i \in [\![d]\!]} \sum_{j \in [\![k_i]\!] \setminus \{a_i^*\}} \frac{\log(T \log T) + \log \frac{\log(T \log T)}{\Delta_{i,j}^2}}{\Delta_{i,j}}\right)$	✓ ‡	✓	✓	✓
This Work F-UCB	✓	✓	Theorem 8.3.3†	Theorem 8.4.2	✓	✗	✓	✓
This Work F-Track				Theorem 8.5.1†	✓	✓	✓	✓

Table 8.2: *Comparison with the worst-case guarantees of (Bubeck et al., 2013) (Zimmert and Seldin 2018 do not provide worst-case bounds).*

	Setting Characteristics		Lower Bound	Upper Bound	Match			
	Factored Structure	Intermediate Feedback			σ	d	k	T
Robust-UCB (Bubeck et al., 2013)	✗	✗	$\Omega\left(\underline{\sigma}\sqrt{k^d T}\right)$	$\mathcal{O}\left(\overline{\sigma}\sqrt{k^d T}\right)$	✗	✓	✓	✓
This Work (F-UCB)	✓	✓	Theorem 8.3.1	Theorem 8.4.1	✓	✓	✓	✓

dependent optimality and to consider functions for aggregating intermediate observations different from the product.

CHAPTER 9

Discussion and Conclusions

In this dissertation, we presented online learning methods for dynamic pricing and advertising budget optimization from both the theoretical and the applicative perspectives. In Part I, after having introduced the basic notions of dynamic pricing, we presented two novel methods for dynamic pricing. We first faced the problem of performing dynamic pricing for an e-commerce website from a practical perspective by designing an algorithm able to price different kinds of products with different dynamics. Then, with the second proposal, we went through more theoretical aspects of pricing by proposing a model that allows us to efficiently model temporal dependencies in pricing through the adoption of parametric autoregressive processes. In Part II, we faced the problem of advertising items properly. After having introduced the basic notions of budget optimization in advertising, we focused our attention on Marketing Mix Models. We discussed the peculiarities of such models, and we propose an algorithm for online optimizing such models, under linearity assumptions, in order to perform exploration-exploitation with theoretical guarantees. In Part III, we discussed the problem related to the suboptimality we may face when we perform pricing and advertising separately, and we proposed a method to perform such a task together in a coherent way using a single agent.

In this chapter, we revise and highlight the contributions of this dissertation, and we discuss possible extensions.

9.1 Dynamic Pricing

The first part of the thesis discussed bandit methods for dynamic pricing.

We first presented (Chapter 4) a MAB solution to optimize the pricing strategy of an e-commerce with several kinds of products with different market dynamics and in the presence of scarce data. We conducted an empirical evaluation, first on synthetic and then on real-world data. In the synthetic scenario, we demonstrated the practical effectiveness of enforcing monotonic demand curves in the presence of scarce data. The empirical evaluation in the real world gives interesting insights on the long-tail market. Our algorithm provided a significant increment in the revenue of the long-tail products, while the effect on popular products is positive but more limited due to (i) the higher competition in their market and (ii) the chance of e-commerce expert to properly define pricing strategy due to the limited number of products to manage.

Then, we changed our focus on the temporal dependences in pricing strategies (Chapter 5). We focused on autoregressive processes, as we observed they are an interesting trade-off between the need to impose a temporal structure in our decision-making and the need for avoiding sample inefficient reinforcement learning solutions. We presented a setting called *Autoregressive Bandits* in which the goal is to minimize the regret while we learn an action-dependent temporal structure. We proposed an optimistic algorithm, and we characterize its performance in terms of expected cumulative regret. We tested our solution in synthetic scenarios, and we evaluated our algorithm in a realistic pricing scenario generated using real-world data.

Possible extensions of this part include the study of the positive and negative interactions between the products and modeling them in a tractable way.

9.2 Advertising

In the second part, we changed our focus to consider advertising problems. This is the other important topic that requires our attention when we want to sell a product. Among the several problems that artificial intelligence can help to solve, our focus is on budget optimization. In particular, we found that an area not yet studied properly is one of the budget optimization for marketing mix models.

Given that, we propose (Chapter 7) a theoretical framework, called *Dynamical Linear Bandits* in which the reward is influenced by (i) the action we perform and (ii) a hidden state that evolves due to our past actions. This framework is suited to model the problem of the MMM, assuming that the behavior can be approximated as linear with constraints. We studied the setting and its intrinsic complexity by providing a regret lower bound. We proposed an optimistic regret minimization algorithm to learn in this setting, and we demonstrated its theoretical soundness. After having tested the algorithm in a synthetic setting, we created a simulator using real-world data in an MMM scenario, and we verified its behavior in this context.

An interesting direction in this field includes the relaxation of the linearity assumptions, avoiding at the same time considering the problem as a non-linear system, which is known to be intractable.

9.3 Joint Pricing and Advertising

The last part of this dissertation is dedicated to understanding what happens when we want to join the two worlds discussed above: dynamic pricing and advertising budget optimization. We define (Chapter 8) the first general framework for treating bandits with intermediate observations that generalize the scenario of the joint optimization of pricing and advertising. We studied the complexity of this setting by presenting both the instance-dependent and the worst-case lower bounds. We developed two algorithms for solving the regret minimization task in this setting, and we discussed their theoretical guarantees.

The problem of jointly optimizing these two aspects is considered by taking into account a simple advertising scenario and a basic pricing problem. Future research directions may focus on the integration of complex dynamics like the one presented in Parts I and II in this joint optimization algorithm.

Bibliography

Samet Oymak and Necmiye Ozay. Non-asymptotic identification of LTI systems from a single trajectory. In *American Control Conference*, pages 5655–5661, 2019.

Marco Mussi, Gianmarco Genalti, Francesco Trovò, Alessandro Nuara, Nicola Gatti, and Marcello Restelli. Pricing the long tail by explainable product aggregation and monotonic bandits. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3623–3633, 2022a.

Gianmarco Genalti, Marco Mussi, Alessandro Nuara, and Nicola Gatti. Dynamic pricing with online data aggregation and learning. In *Fifteenth European Workshop on Reinforcement Learning*, 2022.

Francesco Bacchiocchi, Gianmarco Genalti, Davide Maran, Marco Mussi, Marcello Restelli, Nicola Gatti, and Alberto Maria Metelli. Autoregressive bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Francesco Bacchiocchi, Gianmarco Genalti, Davide Maran, Marco Mussi, Marcello Restelli, Nicola Gatti, and Alberto Maria Metelli. Online learning in autoregressive dynamics. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

Marco Mussi, Alberto Maria Metelli, and Marcello Restelli. Dynamical linear bandits. In *International Conference on Machine Learning*

(ICML), volume 202 of *Proceedings of Machine Learning Research*, pages 25563–25587. PMLR, 2023a.

Marco Mussi, Alberto Maria Metelli, and Marcello Restelli. Dynamical linear bandits for long-lasting vanishing rewards. *Complex Feedback in Online Learning Workshop at International Conference on Machine Learning*, 2022b.

Marco Mussi, Simone Drago, Marcello Restelli, and Alberto Maria Metelli. Factored-reward bandits with intermediate observations. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Simone Drago, Marco Mussi, Marcello Restelli, and Alberto Maria Metelli. Intermediate observations in factored-reward bandits. *Adaptive and Learning Agents Workshop at International Conference on Autonomous Agents and Multi-Agent Systems*, 2024.

Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2021.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1015–1022. Omnipress, 2010.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2312–2320, 2011.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002a.

Sébastien Bubeck. *Bandits games and clustering foundations*. PhD thesis, Université des Sciences et Technologie de Lille, 2010.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the Conference on Learning Theory (COLT)*, 2009.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785–2836, 2010.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 844–853. PMLR, 2017.

Xu Cai and Jonathan Scarlett. On lower bounds for standard and robust gaussian process bandit optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 1216–1226. PMLR, 2021.

Michael Chui, James Manyika, Mehdi Miremadi, Nicolaus Henke, Rita Chung, Pieter Nel, and Sankalp Malhotra. Notes from the ai frontier: Insights from hundreds of use cases. *McKinsey Global Institute*, 2018.

Arnoud V Den Boer. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18, 2015.

Gurpreet Singh, Harjot Kaur, and Amritpal Singh. Dropshipping in e-commerce: A perspective. In *Proceedings of the International Conference on E-business, Management and Economics (ICEME)*, pages 7–14. ACM, 2018.

Chris Anderson. *The long tail: Why the future of business is selling less of more*. Hachette Books, 2006.

Erik Brynjolfsson, Yu Hu, and Duncan Simester. Goodbye pareto principle, hello long tail: The effect of search costs on the concentration of product sales. *Management Science*, 57(8):1373–1386, 2011.

George Kingsley Zipf. *Human behavior and the principle of least effort*. Addison-Wesley Press, 1949.

Stephen Kokoska and Daniel Zwillinger. *CRC standard probability and statistics tables and formulae*. Crc Press, 2000.

Yadati Narahari, CVL Raju, K Ravikumar, and Sourabh Shah. Dynamic pricing models for electronic business. *Sadhana*, 30(2):231–256, 2005.

Dimitris Bertsimas and Georgia Perakis. Dynamic pricing: A learning approach. *Mathematical and computational models for congestion charging*, pages 45–79, 2006.

Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202, 1974.

Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Annual IEEE Symposium on Foundations of Computer Science*, pages 594–605. IEEE, 2003.

Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Multi-armed bandit for pricing. In *European Workshop on Reinforcement Learning*, pages 1–9, 2015.

Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98:196–235, 2018.

Marco Mussi, Gianmarco Genalti, Alessandro Nuara, Francesco Trovó, Marcello Restelli, and Nicola Gatti. Dynamic pricing with volume discounts in online settings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15560–15568, 2023b.

Kanishka Misra, Eric M Schwartz, and Jacob Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252, 2019.

Omar Besbes and Assaf Zeevi. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739, 2015.

Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.

Josef Broder and Paat Rusmevichientong. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980, 2012.

Eric Cope. Bayesian strategies for dynamic pricing in e-commerce. *Naval Research Logistics (NRL)*, 54(3):265–281, 2007.

Josef Bauer and Dietmar Jannach. Optimal pricing in e-commerce based on sparse and noisy data. *Decision Support Systems*, 106:53–63, 2018.

Victor F Araman and René Caldentey. Dynamic pricing for nonperishable products with demand learning. *Operations research*, 57(5):1169–1188, 2009.

Yining Wang, Boxiao Chen, and David Simchi-Levi. Multimodal dynamic pricing. *Management Science*, 2021a.

Mila Nambiar, David Simchi-Levi, and He Wang. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11):4980–5000, 2019.

Amit Gandhi, Zhentong Lu, and Xiaoxia Shi. Estimating demand for differentiated products with zeroes in market share data. *Available at SSRN*, -(-):1–57, 2020.

Hammaad Adam, Pu He, and Fanyin Zheng. Machine learning for demand estimation in long tail markets. *Columbia Business School Research Paper Forthcoming*, -(-):1–43, 2020.

Sentao Miao, Xi Chen, Xiuli Chao, Jiaxi Liu, and Yidong Zhang. Context-based dynamic pricing with online clustering. *arXiv preprint arXiv:1902.06199*, pages 1–53, 2019.

Peng Ye, Julian Qian, Jieying Chen, Chen-hung Wu, Yitong Zhou, Spencer De Mars, Frank Yang, and Li Zhang. Customized regression model for airbnb dynamic pricing. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 932–940. ACM, 2018.

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1(Jun):211–244, 2001.

William R Dougan. Giffen goods and the law of demand. *Journal of Political Economy*, 90(4):809–815, 1982.

Simon Kemp. Perceiving luxury and necessity. *Journal of Economic Psychology*, 19(5):591–606, 1998.

Sergej Bernstein. Démonstration du théoréme de weierstrass fondée sur le calcul des probabilités. *Communications of the Kharkov Mathematical Society*, -:1–2, 1912.

George G. Lorentz. *Bernstein Polynomials*. University of Toronto Press, 1953a.

George G. Lorentz. Degree of approximation. In *Bernstein Polynomials*, chapter 1.6, pages 19–23. University of Toronto Press, 1953b.

Curtis S McKay and Sujit K Ghosh. A variable selection approach to monotonic regression with bernstein polynomials. *Journal of Applied Statistics*, 38(5):961–976, 2011.

Ander Wilson, Jessica Tryner, Christian L'Orange, and John Volckens. Bayesian nonparametric monotone regression. *Environmetrics*, 31(8): e2642, 2020.

Peter J. Olver and Chehrzad Shakiban. Practical linear algebra. In *Applied Linear Algebra*, chapter 1.7, pages 52–53. Springer, 2019.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

W John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of Information Science*, 18(1):45–55, 1992.

Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *Journal of Research and Development*, 1(4):309–317, 1957.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

Fionn Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, 1983.

Chris Ding and Xiaofeng He. Cluster merging and splitting in hierarchical clustering algorithms. In *International Conference on Data Mining (ICDM)*, pages 139–146. IEEE Computer Society, 2002.

James Douglas Hamilton. *Time series analysis*. Princeton university press, 2020.

Jonas W Mueller, Vasilis Syrgkanis, and Matt Taddy. Low-rank bandit methods for high-dimensional dynamic pricing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 1–11. NeurIPS Proceedings, 2019.

John T Bowen and Shiang-Lih Chen. The relationship between customer loyalty and customer satisfaction. *International Journal of Contemporary Hospitality Management*, 2001.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *JMLR Proceedings*, pages 208–214. JMLR.org, 2011.

Ofer Dekel, Ambuj Tewari, and Raman Arora. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Qinyi Chen, Negin Golrezaei, and Djallel Bouneffouf. Non-stationary bandits with auto-regressive temporal dependency. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvari, and Steffen Grunewalder. Bandits with delayed, aggregated anonymous feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4105–4113. PMLR, 2018.

Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless markov bandits. In *International Conference on Algorithmic Learning Theory*, volume 7568 of *Lecture Notes in Computer Science*, pages 214–228. Springer, 2012.

Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

Horia Mania, Michael I Jordan, and Benjamin Recht. Active learning for nonlinear system identification with guarantees. *Journal of Machine Learning Research*, 23:32–1, 2022.

Jonas Umlauft and Sandra Hirche. Learning stable stochastic nonlinear dynamical systems. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3502–3510. PMLR, 2017.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15312–15325, 2020.

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Model learning predictive control in nonlinear dynamical systems. In *IEEE Conference on Decision and Control (CDC)*, pages 757–762. IEEE, 2021.

Sartaj Sahni. Computationally related problems. *SIAM Journal on Computing*, 3(4):262–279, 1974.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 355–366, 2008.

Fahad Albalawi, Zihang Dong, and David Angeli. Regret-based robust economic model predictive control for nonlinear dissipative systems. In *European Control Conference (ECC)*, pages 1105–1111. IEEE, 2021.

Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1077–1086. ACM, 2014.

Yong Yuan, Feiyue Wang, Juanjuan Li, and Rui Qin. A survey on real time bidding advertising. In *Proceedings of International Conference on Service Operationsand Logistics, and Informatics*, pages 418–423. IEEE, 2014.

Anatoli Colicev, Ashish Kumar, and Peter O'Connor. Modeling the relationship between firm and user generated content and the stages of the marketing funnel. *International Journal of Research in Marketing*, 36 (1):100–116, 2019.

Hani I Mesak and Thomas L Means. Modelling advertising budgeting and allocation decisions using modified multinomial logit market share models. *Journal of the Operational Research Society*, 49:1260–1269, 1998.

David Court, Dave Elzinga, Susan Mulder, and Ole Jørgen Vetvik. The consumer decision journey. *McKinsey Quarterly*, 3:96–107, 2009.

Ron Berman. Beyond the last touch: Attribution in online advertising. *Marketing Science*, 37(5):771–792, 2018.

Paul R. Hoban and Randolph E. Bucklin. Effects of internet display advertising in the purchase funnel: Model-based insights from a randomized field experiment. *Journal of Marketing Research*, 52(3):375–393, 2015.

Olivier Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1097–1105. Association for Computing Machinery, 2014.

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Pooria Joulani, András György, and Csaba Szepesvári. Online learning under delayed feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1453–1461, 2013.

Ciara Pike-Burke, Shipra Agrawal, Csaba Szepesvári, and Steffen Grünewälder. Bandits with delayed, aggregated anonymous feedback.

In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4102–4110, 2018.

Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Nonstochastic bandits with composite anonymous feedback. In *Proceedings of the Conference On Learning Theory (COLT)*, pages 750–773, 2018.

JoÃ£o P. Hespanha. *Linear Systems Theory: Second Edition*. Princeton University Press, 2018.

Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17:1–6, 2012.

Rudolf Emil Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics*, 1 (2):152–192, 1963.

B. L. Ho and Rudolf Emil Kalman. Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548, 1966.

Anastasios Tsiamis and George J. Pappas. Finite sample analysis of stochastic system identification. In *IEEE Conference on Decision and Control*, pages 3648–3654, 2019.

Tuhin Sarkar, Alexander Rakhlin, and Munther A. Dahleh. Finite time LTI system identification. *Journal of Machine Learning Research*, 22:26:1–26:61, 2021.

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Logarithmic regret bound in partially observable linear dynamical systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Regret minimization in partially observable linear quadratic control. *CoRR*, abs/2002.00082, 2020b.

Naman Agarwal, Elad Hazan, and Karan Singh. Logarithmic regret for online control. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10175–10184, 2019.

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 1–26, 2011.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In *Proceedings of the Conference on Learning Theory (COLT)*, volume 125, pages 3320–3436. PMLR, 2020.

Orestis Plevrakis and Elad Hazan. Geometric exploration for online control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brückner. Linear bandits with stochastic delayed feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 9712–9721, 2020.

Bingcong Li, Tianyi Chen, and Georgios B. Giannakis. Bandit online learning with unknown delays. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 993–1002, 2019.

Tal Lancewicki, Shahar Segal, Tomer Koren, and Yishay Mansour. Stochastic multi-armed bandits with unrestricted delay distributions. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5969–5978, 2021.

Anne Gael Manegueu, Claire Vernade, Alexandra Carpentier, and Michal Valko. Stochastic bandits with arm-dependent delays. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3348–3356, 2020.

Shinji Ito, Daisuke Hatano, Hanna Sumita, Kei Takemura, Takuro Fukunaga, Naonori Kakimura, and Ken-ichi Kawarabayashi. Delay and cooperation in nonstochastic linear bandits. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Tobias Sommer Thune, Nicolò Cesa-Bianchi, and Yevgeny Seldin. Nonstochastic multiarmed bandits with unrestricted delays. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6538–6547, 2019.

Tiancheng Jin, Tal Lancewicki, Haipeng Luo, Yishay Mansour, and Aviv Rosenberg. Near-optimal regret for adversarial mdp with delayed bandit feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:33469–33481, 2022.

Mengyan Zhang, Russell Tsuchida, and Cheng Soon Ong. Gaussian process bandits with aggregated feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9074–9081, 2022.

Siddhant Garg and Aditya Kumar Akash. Stochastic bandits with delayed composite anonymous feedback. *CoRR*, abs/1910.01161, 2019.

Siwei Wang, Haoyun Wang, and Longbo Huang. Adaptive algorithms for multi-armed bandit with composite and anonymous feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10210–10217, 2021b.

Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10154–10164, 2019.

H. X. Li and P. P. J. Van Den Bosch. A robust disturbance-based control and its application. *International Journal of Control*, 58(3):537–554, 1993.

Joshua D. Isom, Sean P. Meyn, and Richard D. Braatz. Piecewise linear dynamic programming for constrained pomdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 291–296. AAAI Press, 2008.

Aditya Undurti and Jonathan P. How. An online algorithm for constrained pomdps. In *IEEE International Conference on Robotics and Automation*, pages 3966–3973. IEEE, 2010.

Dongho Kim, Jaesong Lee, Kee-Eung Kim, and Pascal Poupart. Point-based value iteration for constrained pomdps. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1968–1974, 2011.

Yonatan Gur, Assaf Zeevi, and Omar Besbes. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 199–207, 2014.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12017–12026, 2019.

Sadegh Nobari. DBA: dynamic multi-armed bandit algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9869–9870, 2019.

Jon Feldman, Shanmugavelayutham Muthukrishnan, Martin Pal, and Cliff Stein. Budget optimization in search-based advertising auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 40–49, 2007.

Julian Zimmert and Yevgeny Seldin. Factored bandits. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2840–2849, 2018.

Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. *Inequalities*. Cambridge University Press, 1952.

Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 728–737. PMLR, 2017.

Zhi Wang, Chicheng Zhang, Manish Kumar Singh, Laurel D. Riek, and Kamalika Chaudhuri. Multitask bandit learning through heterogeneous feedback aggregation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130, pages 1531–1539. PMLR, 2021c.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

Pravin M Vaidya. Speeding-up linear programming using fast matrix multiplication. In *Annual symposium on foundations of computer science*, pages 332–337. IEEE Computer Society, 1989.

J.M. Luttinger and R. Friedberg. A new rearrangement inequality for multiple integrals. *Archive for Rational Mechanics and Analysis*, 61:45–64, 1976.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

Stefan Magureanu, Richard Combes, and Alexandre Proutière. Lipschitz bandits: Regret lower bound and optimal algorithms. In *Proceedings of the Conference on Learning Theory (COLT)*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 975–999. JMLR.org, 2014.

Jia Yuan Yu and Shie Mannor. Unimodal bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 41–48. Omnipress, 2011.

Richard Combes, Stefan Magureanu, and Alexandre Proutière. Minimal exploration in structured stochastic bandits. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1763–1771, 2017.

Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 535–543. PMLR, 2015.

Richard Combes, Mohammad Sadegh Talebi, Alexandre Proutière, and Marc Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2116–2124, 2015.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvári, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 392–401. PMLR, 2017.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283. USENIX Association, 2016.

Iosif Pinelis. Product of three or more independent sub-gaussian varibles. MathOverflow, 2021.

APPENDIX $\mathcal{A}$

Algorithmical Details for Chapter 4

In this chapter, we provide implementation details of the algorithm presented in Chapter 4.

The implementation of the *DynaLT* algorithm for the experiments presented in Section 4.7 has been done using Python 3. More specifically, the Bayesian Regression Model is implemented using TensorFlow Probability library (Abadi et al., 2016). In what follows, we provide the implementation details to allow the replicability of the experiments, i.e., the seasonality estimation (Appendix A.1), the distance estimation procedure (Appendix A.2), and the synthetic environment creation (Appendix A.3). Finally, we discuss the algorithm running time (Appendix A.4).

A.1 Seasonality

Figure 4.2 shows that, even if the seasonality effect is relevant, it is stable across years since the standard deviation bounds provided as semitransparent areas are small. Therefore, we model it as a multiplicative factor $s_{j\tau}$ for each product j at time τ such that we can compute seasonality adjusted volumes as $\bar{v}_{j\tau} := v_{j\tau} \cdot s_{j\tau}$.

The seasonality term $s_{j\tau}$ is estimated in a data-driven way using data coming from a set of previous years $\mathcal{Y}$. We denote with v_{jwy} the volume for product j corresponding to a week of the year $w \in \{1, \dots, 52\}$ and a year $y \in \mathcal{Y}$. At first, let us compute for a product j the proportion of the volumes sold in a specific week w for a year y, formally:

$$\hat{v}_{jwy} = \frac{v_{jwy}}{\sum_{i \in \mathcal{W}} v_{jiy}}. \tag{A.1}$$

The seasonality factor $s_j(w)$ for a specific week w is computed as follows:

$$s_j(w) = \frac{1}{\sum_{y \in \mathcal{Y}} \hat{v}_{jwy}/Y + H}, \tag{A.2}$$

where H is a shrinkage factor, and $Y := |\mathcal{Y}|$ is the cardinality of the set $\mathcal{Y}$. Finally, the correction factor $s_{j\tau}$ is equal to the $s_j(w)$ for the week w of the year corresponding to time τ.

The same procedure is applicable for meta-product by using the aggregated volumes of the product therein, i.e., for meta-product $\mathcal{K}$:

$$v_{\mathcal{K}wy} = \sum_{k \in \mathcal{K}} v_{kwy}.$$

In the experiments, the shrinkage factor is selected equal to $H = 0.005$ based on empirical evidence.

A.2 Similarity Estimation

In our application, each product $j \in \mathcal{J}$ has a textual description ρ_j, which contains information regarding the product, like its brand, color, and material.[1] The corpus of strings $\Pi = \{\rho_j\}_{j \in \mathcal{J}}$[2] is represented by the descriptions of all the available products.

With $|\rho_j|$ we denote the dimension of the string computed as the number of words it contains. TF-IDF encoding balances the importance tf_{ij} of a word i in a string ρ_j and the importance idf_i of the word i across the whole textual data set. Formally:

$$tf_{ij} = \frac{v_{ij}}{|\rho_j|},$$

$$idf_i = \log_{10} \frac{|\Pi|}{|\{\rho \in \Pi \text{ s.t. } i \in \rho\}|},$$

[1] For the sake of presentation we focus on the textual description, but one might also concatenate additional textual information, like the product category or its name.

[2] Note that $|\Pi| = |\mathcal{J}|$.

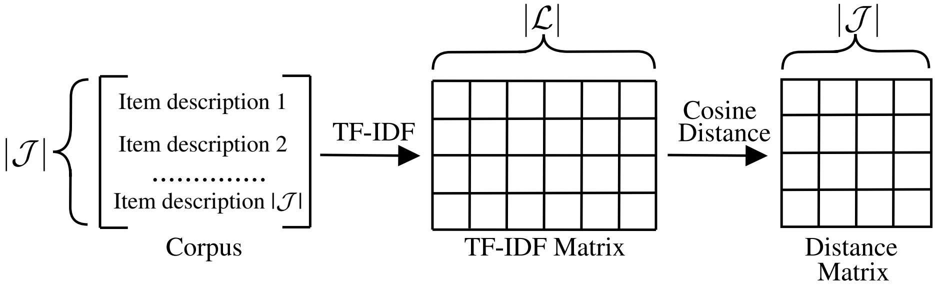


Figure A.1: *Overall scheme of the TF-IDF algorithm.*

where v_{ij} is the number of occurrences of the word i in description ρ_j of product j, $|\Pi|$ is the number of string present in the corpus, and $|\{\rho \in \Pi \text{ s.t. } i \in \rho\}|$ is the number of textual descriptions ρ in which the word i is present among the one of the entire catalog Π. The TF-IDF score for the word i in the description ρ_j of product j is computed as follows:

$$tfidf_{ij} = tf_{ij} \cdot idf_i.$$

The result is a vector $\eta_j \in [0,1]^{|\mathcal{L}|}$, where $\mathcal{L}$ is the set of distinct words (obtained after a stop-word removal procedure) in all the texts. For each product j, $[\eta_j]_i = tfidf_{ij}$ is the TF-IDF score of word $i \in \mathcal{L}$ for the text defined by ρ_j. The distance d_{jl} between two products j and l is computed using a transformation of the cosine similarity, formally:

$$d_{jl} = 1 - \frac{\eta_j \cdot \eta_l}{||\eta_j||_2 \cdot ||\eta_l||_2}.$$

where $\cdot$ represents the scalar product between vectors and $||\eta||_2$ represents the 2-norm of η. Figure A.1 represents the whole process that goes from textual data to the computation of a pairwise distance matrix between the products.[3]

A.3 Simulation Details

A.3.1 Noisy Environment Simulation

The volumes for the single product pricing experiments in Section 4.7.1 are generated from the volume function:[4]

$$v_1(x) = 2e^{-(x+1.2)^{\frac{5}{2}}} + \epsilon,$$

[3] Other ways of vectorization such as embedded-based ones are also viable options in the case the textual descriptions are succinct.

[4] Notice that the chosen demand function satisfies the motononicity assumption.

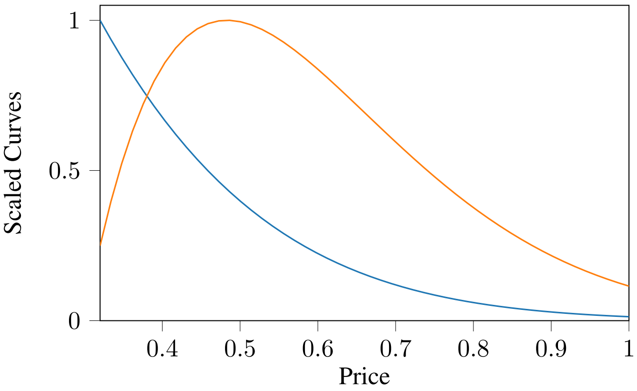


Figure A.2: *Demand curve used in the noisy experiment and corresponding reward function obtained maximizing profit.*

where prices $x \in [0.32, 1]$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian zero-mean noise with variance σ^2. The product had a unitary cost $c = 0.3$. A graphic representation of the volumes curve corresponding to this product is provided in Figure A.2.
In this, we modified noise's standard deviation σ and introduced some outliers in the data generation process. More specifically, the outliers generation is obtained through the probability $o \in (0, 1)$ that a sample drawn from a demand curve has noise ϵ' such that its standard deviation is 10 times the one of ϵ.

A.3.2 Non-stationary Environment Simulation

In the second experiment, three different volume functions have been used during the different phases of the non-stationary process. The volume functions were:

$$\begin{aligned} v_1(x) &= \frac{3}{10}(1 - x), \\ v_2(x) &= 2e^{-(x+1.2)^{\frac{5}{2}}}, \\ v_3(x) &= 7e^{-(x+1.2)^3}. \end{aligned}$$

Their corresponding volumes curves are provided in Figure A.3. The first abrupt change substituted the volume function $v_1(x)$ with $v_2(x)$, the second substituted $v_2(x)$ with $v_3(x)$, and the third one $v_3(x)$ with $v_1(x)$. In this set of experiments the noise's standard deviation is $\sigma = 0.001$, and the outliers' percentage is $o = 0\%$.

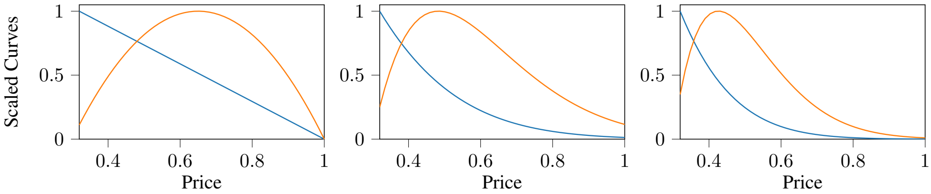


Figure A.3: *Demand curves used in the non-stationary experiment and corresponding reward functions.*

A.3.3 Algorithm Settings

In the first scenario, the demand curve have been estimated using Bernstein's Polynomial with $Z = 75$. The priors for the Lognormal and Gaussian distribution of the BRL model have been set with $\sigma_h = 0.75$ and $\sigma_h = 0.5$, respectively. The values for the hyper-parameters have been chosen basing on an independent data set. The sampling procedure described in Section 4.5 have been applied to the set of margins $\mathcal{M}$ of evenly spaced values over the domain $[0.05, 1.5]$, where $|\mathcal{M}| = 50$.

In the second scenario, we use the same configurations for the Bernstein's Polynomial and the sampling procedure. Conversely, the prior parameters for the Lognormal and Gaussian priors were set to $\sigma_h = 0.75$ and $\sigma_h = 2$, respectively.

Notice that the clairvoyant solution to the problem of maximizing the profits is non-trivial even knowing the real volume functions, due to the fact that the introduction of noise and outliers do not allow to compute it in a closed form solution. We estimated the optimal solution using Monte Carlo approach, i.e., we simulated $10,000$ samples from each one of the margins used in the experiments and averaged the values of the profit gained with such a margin. Then we took the maximum over the computed profits as the optimal solution for the problem. Thanks to this approach, the empirical regret is computed as the difference between this value and the one obtained using the analyzed algorithms.

A.4 Algorithm Running Time

The algorithm running time can be analyzed by dividing the process into two phases: first, the distance estimation and the tree structure generation, then, the proper optimal price estimation.

A.4.1 Similarity and Tree Structure

This phase is required to be performed only when there are changes in the catalog of the products. The running time for the distance estimation algorithm is $\mathcal{O}(|\mathcal{J}|^2)$ for what concerns the operations required to construct the distance matrix. Building the agglomerative clustering tree structure requires a running time $\mathcal{O}(|\mathcal{J}|^2 \log |\mathcal{J}|)$ when using single linkage, and $\mathcal{O}(|\mathcal{J}|^3)$ in the general case. It is worth noting that adding a new product to the catalog corresponds to an incremental update of the distance matrix, i.e., adding a new row and column to the matrix consisting of the distance of the new products w.r.t. the previous ones.

A.4.2 Optimal Pricing

The proper estimate of the optimal price must be performed at every time t, as well as the association of a product j with the related meta-product $\mathcal{K}$. This is because the cluster stopping condition is defined over transactions data, which changes over time. Given $|\mathcal{J}|$ products, we must estimate at most (worst-case scenario) $|\mathcal{J}|$ BLR models.

APPENDIX $\mathcal{B}$

Omitted Proofs of Chapter 5

Theorem 5.2.1 (Optimal Policy)**.** *Under Assumption 5.1.a, for every round $t \in \mathbb{N}$, the optimal policy $\pi_t^*(H_{t-1})$ satisfies:*

$$\pi_t^*(H_{t-1}) \in \arg\max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \mathbf{z}_{t-1} \rangle. \tag{5.5}$$

Proof. We first prove an intermediate result auxiliary to get to the final statement. Let us denote with $J_T^*(\mathbf{z})$ the expected cumulative reward when the initial observations vector is $\mathbf{z} = (1, x_0, x_{-1}, \dots, x_{-n+1})$. Let us denote with $\succeq$ the element-wise inequality. We show that for every $T \in \mathbb{N}$, if $\mathbf{z} \succeq \overline{\mathbf{z}}$, then $J_T^*(\mathbf{z}) \geqslant J_T^*(\overline{\mathbf{z}})$.
We proceed by induction.
For $T = 1$, we have $J_1^*(\mathbf{z}) = \max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \mathbf{z} \rangle = \langle \boldsymbol{\gamma}(a_1^*), \mathbf{z} \rangle$, where $a_1^* \in \arg\max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \mathbf{z} \rangle$ and $J_1^*(\overline{\mathbf{z}}) = \max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \overline{\mathbf{z}} \rangle = \langle \boldsymbol{\gamma}(\overline{a}_1^*), \overline{\mathbf{z}} \rangle$, where $\overline{a}_1^* \in \arg\max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \overline{\mathbf{z}} \rangle$. Thus, we have:

$$J_1^*(\mathbf{z}) = \langle \boldsymbol{\gamma}(a_1^*), \mathbf{z} \rangle \geqslant \langle \boldsymbol{\gamma}(\overline{a}_1^*), \mathbf{z} \rangle \overset{\text{(a)}}{\geqslant} \langle \boldsymbol{\gamma}(\overline{a}_1^*), \overline{\mathbf{z}} \rangle = J_1^*(\overline{\mathbf{z}}),$$

where inequality (a) follows from Assumption 5.1.a.
Suppose the statement holds for $T - 1$, we prove it for $T > 1$. To this end, we consider the *transition operator* $P : \mathcal{Z} \times \mathcal{A} \times \mathbb{R} \to \mathcal{Z}$, defined for every

observations vector $\mathbf{z}_t = (1, x_{t-1}, x_{t-2}, \dots, x_{t-n}) \in \mathcal{Z}$, action $a \in \mathcal{A}$, and noise $\xi \in \mathbb{R}$ as follows:

$$P(\mathbf{z}_t, a, \xi) = P\left(\begin{pmatrix} 1 \\ x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-n} \end{pmatrix}, a, \xi\right) = \begin{pmatrix} 1 \\ x_{t} \\ x_{t-1} \\ \vdots \\ x_{t-n+1} \end{pmatrix} = \mathbf{z}_{t+1},$$

where $x_t = \langle \boldsymbol{\gamma}(a), \mathbf{z}_t \rangle + \xi$. Thus, we can look at the stochastic process as a Markov decision process (Puterman, 2014) with $\mathbf{z}_t$ as state representation. We immediately observe that if $\mathbf{z} \succeq \overline{\mathbf{z}}$, we have that $P(\mathbf{z}, a, \xi) \succeq P(\overline{\mathbf{z}}, a, \xi)$, for every action $a \in \mathcal{A}$ and noise $\xi \in \mathbb{R}$. By applying the Bellman equation, we obtain:

$$\begin{aligned} J_T^*(\mathbf{z}) &= \max_{a \in \mathcal{A}} \left\{ \langle \boldsymbol{\gamma}(a), \mathbf{z} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\mathbf{z}, a, \xi_T)) \right] \right\} \\ &= \langle \boldsymbol{\gamma}(a_T^*), \mathbf{z} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\mathbf{z}, a_T^*, \xi_T)) \right], \\ J_T^*(\overline{\mathbf{z}}) &= \max_{a \in \mathcal{A}} \left\{ \langle \boldsymbol{\gamma}(a), \overline{\mathbf{z}} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\overline{\mathbf{z}}, a, \xi_T)) \right] \right\} \\ &= \langle \boldsymbol{\gamma}(\overline{a}_T^*), \overline{\mathbf{z}} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\overline{\mathbf{z}}, \overline{a}_T^*, \xi_T)) \right], \end{aligned}$$

where the actions are defined as:

$$a_T^* \in \arg\max_{a \in \mathcal{A}} \left\{ \langle \boldsymbol{\gamma}(a), \mathbf{z} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\mathbf{z}, a, \xi_T)) \right] \right\},$$

and:

$$\overline{a}_T^* \in \arg\max_{a \in \mathcal{A}} \left\{ \langle \boldsymbol{\gamma}(a), \overline{\mathbf{z}} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\overline{\mathbf{z}}, a, \xi_T)) \right] \right\}.$$

Thus, we have:

$$\begin{aligned} J_T^*(\mathbf{z}) &= \langle \boldsymbol{\gamma}(a_T^*), \mathbf{z} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\mathbf{z}, a_T^*, \xi_T)) \right] \\ &\geqslant \langle \boldsymbol{\gamma}(\overline{a}_T^*), \mathbf{z} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\mathbf{z}, \overline{a}_T^*, \xi_T)) \right] \\ &\overset{(b)}{\geqslant} \langle \boldsymbol{\gamma}(\overline{a}_T^*), \overline{\mathbf{z}} \rangle + \mathbb{E}_{\xi_T}\left[J_{T-1}^*(P(\overline{\mathbf{z}}, \overline{a}_T^*, \xi_T)) \right] \\ &= J_T^*(\overline{\mathbf{z}}), \end{aligned}$$

where (b) follows from Assumption 5.1.a bounding $\langle \boldsymbol{\gamma}(\overline{a}_T^*), \mathbf{z} \rangle \geqslant \langle \boldsymbol{\gamma}(\overline{a}_T^*), \overline{\mathbf{z}} \rangle$ and by observing that $P(\mathbf{z}, \overline{a}_T^*, \xi_1) \succeq P(\overline{\mathbf{z}}, \overline{a}_T^*, \xi_T)$ and, then, exploiting the inductive hypothesis.

We conclude that the optimal policy is the myopic one by observing that both $\langle \boldsymbol{\gamma}(a), \boldsymbol{z} \rangle$ and $J^*_{T-1}(P(\mathbf{z}, a, \xi))$ are simultaneously maximized by:

$$\arg\max_{a \in \mathcal{A}} \langle \boldsymbol{\gamma}(a), \boldsymbol{z} \rangle.$$

□

Lemma 5.4.1 (Self-Normalized Concentration). *Let $a \in \mathcal{A}$ be an action, let $(\widehat{\boldsymbol{\gamma}}_t(a))_{t \in \mathcal{O}_\infty(a)}$ be the sequence of solutions to the Ridge regression problems computed by Algorithm 5.1. Then, for every regularization parameter $\lambda > 0$, confidence $\delta \in (0,1)$, simultaneously for every round $t \in \mathbb{N}$ and action $a \in \mathcal{A}$, with probability at least $1 - \delta$ it holds that:*

$$\|\widehat{\boldsymbol{\gamma}}_t(a) - \boldsymbol{\gamma}(a)\|_{\mathbf{V}_t(a)} \leqslant \sqrt{\lambda}\|\boldsymbol{\gamma}(a)\|_2 + \sigma\sqrt{2\log\left(\frac{k}{\delta}\right) + \log\left(\frac{\det \mathbf{V}_t(a)}{\lambda^{n+1}}\right)}.$$

Proof. We consider an action at a time; then, the final result is obtained with a union bound over $\mathcal{A} = [\![k]\!]$. Let $a \in \mathcal{A}$. We first observe that the estimates of action a change only when a is pulled. Let $l \in \mathbb{N}$ be an index and let $t_l(a) \in \mathbb{N}$ be the round in which action a is pulled for the l-th time, i.e., $\{t_l(a) \,:\, l \in \mathbb{N}\} = \mathcal{O}_\infty(a)$. Thus, we have:

$$\begin{aligned}
\boldsymbol{\gamma}_{t_l}(a) &= \mathbf{V}^{-1}_{t_l(a)}(a)\mathbf{b}^{-1}_{t_l(a)}(a) \\
&= \left(\lambda \mathbf{I}_{n+1} + \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\mathbf{z}^{\mathrm{T}}_{t_j(a)-1}\right)^{-1} \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1} x_{t_j} \\
&= \left(\lambda \mathbf{I}_{n+1} + \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\mathbf{z}^{\mathrm{T}}_{t_j(a)-1}\right)^{-1} \\
&\qquad\qquad \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\left(\langle \boldsymbol{\gamma}(a), \mathbf{z}_{t_j(a)-1}\rangle + \xi_{t_j(a)}\right) \\
&\overset{\text{(a)}}{=} \boldsymbol{\gamma}(a) - \lambda\left(\lambda \mathbf{I}_{n+1} + \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\mathbf{z}^{\mathrm{T}}_{t_j(a)-1}\right)^{-1} \boldsymbol{\gamma}(a) + \\
&\qquad\qquad + \left(\lambda \mathbf{I}_{n+1} + \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\mathbf{z}^{\mathrm{T}}_{t_j(a)-1}\right)^{-1} \sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\xi_{t_j(a)} \\
&= \boldsymbol{\gamma}(a) - \lambda \mathbf{V}^{-1}_{t_l(a)}(a)\boldsymbol{\gamma}(a) + \mathbf{V}^{-1}_{t_l(a)}(a)\underbrace{\sum_{j=1}^{l} \mathbf{z}_{t_j(a)-1}\xi_{t_j(a)}}_{\mathbf{s}_{t_l(a)}},
\end{aligned}$$

where the passage (a) derives from the observation that:

$$\sum_{j=1}^{l} \mathbf{z}_{t_j-1}(\langle \boldsymbol{\gamma}(a), \mathbf{z}_{t_j-1}\rangle) = \sum_{j=1}^{l} \mathbf{z}_{t_j-1}\mathbf{z}_{t_j-1}^{\mathrm{T}}\boldsymbol{\gamma}(a).$$

Thus, we have:

$$\left\|\boldsymbol{\gamma}_{t_l(a)}(a) - \boldsymbol{\gamma}(a)\right\|_{\mathbf{V}_{t_l(a)}(a)} \leqslant \sqrt{\lambda}\|\boldsymbol{\gamma}(a)\|_2 + \|\mathbf{s}_{t_l(a)}\|_{\mathbf{V}^{-1}_{t_l(a)}(a)}.$$

Let us denote with $\mathcal{F}_{t_l(a)} = \sigma(\mathbf{z}_0, a_1, \mathbf{z}_1, a_2, \dots, \mathbf{z}_{t_l(a)-1}, a_{t_l(a)})$ be the filtration generated by all events realized at round $t_l(a)$. Let us now consider the stochastic processes $(\xi_{t_l(a)})_{l\in\mathbb{N}}$ and $(\mathbf{z}_{t_l(a)-1})_{l\in\mathbb{N}}$. We observe that $\xi_{t_l(a)}$ is $\mathcal{F}_{t_l(a)}$-measurable and conditionally σ^2-subgaussian and that $\mathbf{z}_{t_l(a)-1}$ is $\mathcal{F}_{t_l(a)-1}$-measurable. By applying Theorem 1 of Abbasi-Yadkori et al. (2011), we have that simultaneously for all $l \in \mathbb{N}$, w.p. $1-\delta$:

$$\|\mathbf{s}_{t_l(a)}\|_{\mathbf{V}^{-1}_{t_l(a)}(a)} \leqslant \sigma\sqrt{2\log\frac{1}{\delta} + \log\frac{\det \mathbf{V}_{t_l(a)}(a)}{\lambda^{n+1}}}.$$

Clearly, this hold for the rounds $t \in \mathbb{N}$ in which the action a is not pulled, since the corresponding estimates do not change. □

Lemma 5.4.2 (Policy Regret Decomposition). *Let $(x_t^*)_{t\in[\![T]\!]}$ be the sequence of rewards by executing the optimal policy $\boldsymbol{\pi}^*$ and let $(x_t)_{t\in[\![T]\!]}$ be the sequence of rewards by executing the learner's policy $\boldsymbol{\pi}$. Then, for every $t \in [\![T]\!]$ it holds that:*

$$\begin{aligned} r_t &= x_t^* - x_t \\ &= \sum_{i=1}^{n} \gamma_i(a_t^*)(x_{t-i}^* - x_{t-i}) + \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle \\ &= \sum_{i=1}^{n} \gamma_i(a_t^*) r_{t-i} + \rho_t, \end{aligned} \tag{5.9}$$

where $r_t := x_t^* - x_t$ is the instantaneous policy regret, $\rho_t := \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle$ is the instantaneous external regret, $a_t^* = \pi_t^*(H_{t-1}^*)$, and $r_{t-i} = 0$ if $i \geqslant t$.

Proof. Let $t \in [\![T]\!]$ and let us denote with $\mathbf{z}_{t-1}^* = (1, x_{t-1}^*, \dots, x_{t-n}^*)^{\mathrm{T}}$ the observations vector associated with the execution of the optimal policy and with $\mathbf{z}_{t-1} = (1, x_{t-1}, \dots, x_{t-n})^{\mathrm{T}}$ the observations vector associated with the execution of the learner's policy. We have:

$$r_t = x_t^* - x_t$$

$$
\begin{aligned}
&= \langle \boldsymbol{\gamma}(a_t^*), \mathbf{z}_{t-1}^* \rangle - \langle \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle \\
&= \langle \boldsymbol{\gamma}(a_t^*), \mathbf{z}_{t-1}^* \rangle - \langle \boldsymbol{\gamma}(a_t^*), \mathbf{z}_{t-1} \rangle + \langle \boldsymbol{\gamma}(a_t^*), \mathbf{z}_{t-1} \rangle - \langle \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle \\
&= \langle \boldsymbol{\gamma}(a_t^*), \mathbf{z}_{t-1}^* - \mathbf{z}_{t-1} \rangle + \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle \\
&= \sum_{i=1}^{n} \gamma_i(a_t^*) \underbrace{(x_{t-i}^* - x_{t-i})}_{r_{t-i}} + \underbrace{\langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle}_{\rho_t},
\end{aligned}
$$

where in expanding the inner product we made the summation start from $i = 1$ as the two vectors $\mathbf{z}_{t-1}^*$ and $\mathbf{z}_{t-1}$ have the same first component equal to 1. □

Lemma 5.4.3 (External-to-Policy Regret Bound)**.** *Let $\boldsymbol{\pi}$ be the learner's policy and $T \in \mathbb{N}$ be the horizon. Under Assumptions 5.1.a and 5.1.b, it holds that:*

$$
\mathbb{E}[R(\boldsymbol{\pi}, T)] = \mathbb{E}\left[\sum_{t=1}^{T}\left[\sum_{i=1}^{n} \gamma_i(a_t^*) r_{t-i} + \rho_t\right]\right] \leqslant \left(\frac{\Gamma n}{1-\Gamma} + 1\right) \varrho(\boldsymbol{\pi}, T), \tag{5.10}
$$

where $\varrho(\boldsymbol{\pi}, T) := \mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right]$ is the cumulative expected external regret.

Proof. We start from the decomposition of Lemma 5.4.2. To prove the result we employ the so-called "superposition principle", which allows us to decompose the linear recurrence as follows:

$$
r_t = \sum_{i=1}^{n} \boldsymbol{\gamma}_i(a_t^*) r_{t-i} + \rho_t = \sum_{\tau=0}^{+\infty} \rho_\tau \widetilde{r}_{t,\tau},
$$

where if $\tau > t$ we set $\widetilde{r}_{t,\tau} = 0$ and if $\tau \leqslant t$ we have that $\widetilde{r}_{t,\tau}$ is given by the recurrence:

$$
\widetilde{r}_{t,\tau} = \sum_{i=1}^{n} \boldsymbol{\gamma}_i(a_t^*) \widetilde{r}_{t-i,\tau} + \delta_{t,\tau} \qquad \text{where} \qquad \delta_{t,\tau} := \begin{cases} 1 & t = \tau \\ 0 & t \neq \tau \end{cases}.
$$

This way, we decompose the exogenous term ρ_τ as a linear combination of unitary impulses. Then by Assumption 5.1.a and 5.1.b, recalling that $\widetilde{r}_{t,\tau} = 0$ if $\tau > t$ and that $\widetilde{r}_{\tau,\tau} = 1$, we have that for every $t > \tau$ it holds that:

$$
\widetilde{r}_{t,\tau} \leqslant \Gamma \max_{i \in [\![n]\!]} \widetilde{r}_{t-i,\tau} \leqslant \Gamma^2 \max_{i \in [\![n]\!]} \max_{j \in [\![n]\!]} \widetilde{r}_{t-i-j,\tau} \leqslant \cdots \leqslant \Gamma^{\lceil (t-\tau)/n \rceil},
$$

since we will encounter the $1 = \delta_{\tau,\tau}$ after $\lceil (t-\tau)/n \rceil$ steps of unfolding.

Now, we can manipulate this formula to have an expression of the full regret:

$$\begin{aligned}\sum_{t=1}^{T} r_t &\leqslant \sum_{t=1}^{T}\left(\rho_t + \sum_{\tau=1}^{t-1} \Gamma^{\lceil (t-\tau)/n \rceil}\rho_\tau\right)\\ &= \sum_{\tau=1}^{T}\left(1 + \rho_\tau \sum_{t=\tau+1}^{T} \Gamma^{\lceil (t-\tau)/n \rceil}\right)\\ &\overset{(a)}{\leqslant} \sum_{\tau=1}^{T} \rho_\tau \left(1 + \sum_{s=1}^{+\infty} \Gamma^{\lceil s/n \rceil}\right)\\ &\overset{(b)}{=} \sum_{\tau=1}^{T} \rho_\tau \left(1 + \sum_{l=1}^{+\infty} n\Gamma^{l}\right)\\ &= \left(1 + \frac{\Gamma n}{1-\Gamma}\right)\sum_{\tau=1}^{T} \rho_\tau,\end{aligned}$$

where (a) follows from bounding the summation with the series and changing the index $s = t - \tau$ and (b) is obtained by observing that the exponent $\lceil s/n \rceil$ changes only when s is divisible by n. □

Counterexample to show that this bound is tight There are k arms:

$$\boldsymbol{\gamma}(a_1) := [\Gamma, 0 \ldots 0], \quad \boldsymbol{\gamma}(a_2) := [0, \Gamma, 0 \ldots 0], \ \ldots \ \boldsymbol{\gamma}(a_k) := [0, \ldots 0, \Gamma].$$

All these arms have non-negative coefficients whose sum is bounded by Γ. If the sequence of internal regrets is:

$$\rho_t = \begin{cases} 1 & t = 1 \\ 0 & t > 1 \end{cases},$$

and the sequence of arms is $a_1^* = 1$, and $a_t^* = a_{t-1 \ (mod \ k)+1}$ (which means $a_1, a_2, \ldots, a_k, a_1, a_2, \ldots$), we have:

$$r_1 = 1, r_2 = \Gamma, \ r_3 = \Gamma, \ \ldots, \ r_{k+1} = \Gamma,$$

and then, we start again with the same sequence of arms:

$$r_{k+2} = \Gamma^2, \ r_{k+3} = \Gamma^2, \ \ldots, \ r_{2k+1} = \Gamma^2.$$

Making the sum of these terms for t from one to infinity, we get:

$$\sum_{t=1}^{\infty} r_t = 1 + k \sum_{t=1}^{\infty} \Gamma^t = 1 + \frac{k\Gamma}{1-\Gamma},$$

which is exactly the bound we get.

Lemma B.0.1. *Let $(\mathbf{z}_t)_{t \in [\![T]\!]}$ be the sequence of observations vectors observed by executing the learner's policy. If $\mathbf{z}_0 = (1, 0, \dots, 0)^T$, then, for every $\delta \in (0, 1)$, with probability at least $1 - \delta$, simultaneously for all $t \in [\![T]\!]$, it holds that:*

$$\|\mathbf{z}_{t-1}\|_2 \leqslant \sqrt{1 + n \left(\frac{m + \eta}{1 - \Gamma} \right)^2},$$

where $\eta = \sqrt{2\sigma^2 \log(T/\delta)}$.

Proof. Let $(\xi_t)_{t \in [\![T]\!]}$ be the sequence of noises. We consider the event $\mathcal{E} = \bigcap_{t=1}^{T} \left\{ |\xi_t| \leqslant \eta \right\}$ prescribing that all noises are smaller than η in absolute value. By union bound, knowing that all the noises are independent σ^2-subgaussian random variables we, can bound the probability of event $\mathcal{E}$:

$$\mathbb{P}(\mathcal{E}) = \mathbb{P}\left(\bigcap_{t=1}^{T} \left\{ |\xi_t| \leqslant \eta \right\} \right) \geqslant 1 - Te^{-\frac{\eta^2}{2\sigma^2}} = 1 - \delta,$$

having set $\eta = \sqrt{2\sigma^2 \log(T/\delta)}$. Under event $\mathcal{E}$ when $\mathbf{z}_0 = (1, 0, \dots, 0)^{\mathrm{T}}$, we prove by induction that all rewards x_t are bounded in absolute value by $\frac{m+\eta}{1-\Gamma}$, regardless the actions played. For $T = 1$, the statement is trivial since $x_1 = \gamma_0(a_1) + \eta_1$ and, thus, $|x_1| \leqslant \gamma_0(a_1) + |\eta_1| \leqslant m + \eta \leqslant \frac{m+\eta}{1-\Gamma}$. Suppose the statement holds for all $s < t$, we prove it for t. We have:

$$\begin{aligned} x_t = \gamma_0(a_t) + \sum_{i=1}^{n} \gamma_i(a_t) x_{t-i} + \eta_t \\ \implies |x_t| &\leqslant \gamma_0(a_t) + \sum_{i=1}^{n} \gamma_i(a_t) |x_{t-i}| + |\eta_t| \\ &\leqslant m + \Gamma \frac{m + \Gamma}{1 - \Gamma} + \eta = \frac{m + \eta}{1 - \Gamma}, \end{aligned}$$

where the first inequality uses Assumption 5.1.a, the second inequality follows from the inductive hypothesis and by Assumptions 5.1.b and 5.1.c. Passing to the observations vector, we have:

$$\|\mathbf{z}_{t-1}\|_2^2 = 1 + \sum_{i=1}^{n} x_{t-i}^2 \leqslant 1 + n \left(\frac{m + \eta}{1 - \Gamma} \right)^2.$$

□

For deriving the regret bound, we make use of the following result, known as *Elliptic Potential Lemma* (Lattimore and Szepesvári, 2020, Lemma 19.4).

Lemma B.0.2 (Elliptic Potential Lemma). *Let* $\mathbf{V}_0 \in \mathbb{R}^{d \times d}$ *be a positive definite matrix and let* $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^d$ *be a sequence of vectors such that* $\|\mathbf{a}_t\|_2 \leqslant L < +\infty$ *for all* $t \in [\![k]\!]$. *Let* $\mathbf{V}_t = \mathbf{V}_0 + \sum_{s=1}^{t} \mathbf{a}_s \mathbf{a}_s^T$, *Then:*

$$\sum_{t=1}^{k} \min\{1, \|\mathbf{a}_s\|_{\mathbf{V}_{t-1}{}^{-1}}\} \leqslant 2d \log\left(\frac{\operatorname{tr}(\mathbf{V}_0) + kL^2}{d \det(\mathbf{V}_0)^{1/d}}\right).$$

Theorem 5.4.4. *Let* $\delta = (2T)^{-1}$. *Under Assumptions 5.1.a, 5.1.b, and 5.1.c,* `AR-UCB` *suffers a cumulative expected (policy) regret bounded by (highlighting the dependence on* m, σ, n, Γ, k, *and* T *only):*

$$\mathbb{E}[R(\texttt{AR-UCB}, T)] \leqslant \widetilde{\mathcal{O}}\left(\frac{(m+\sigma)(n+1)^{3/2}\sqrt{kT}}{(1-\Gamma)^2}\right).$$

Proof. We denote with $(x_t^*)_{t \in [\![T]\!]}$ the sequence of rewards generated by playing the optimal policy and with $(x_t)_{t \in [\![T]\!]}$ the sequence of rewards generated by playing `AR-UCB`. Thanks to Lemma 5.4.3, we have to bound the external regret only. Let $\delta \in (0,1)$, and define for every round $t \in [\![T]\!]$ and action $a \in \mathcal{A}$:

$$\beta_t(a) := \sqrt{\lambda(m^2+1)} + \sigma\sqrt{2\log\left(\frac{n}{\delta}\right) + \log\left(\frac{\det \mathbf{V}_t(a)}{\lambda^{n+1}}\right)}.$$

Let us define the confidence set:

$$\mathcal{C}_t(a) := \{\boldsymbol{\gamma} \in \mathbb{R}^{n+1} : \|\boldsymbol{\gamma} - \widehat{\boldsymbol{\gamma}}_{t-1}(a)\|_{\mathbf{V}_{t-1}(a)} \leqslant \beta_{t-1}(a)\},$$

and the optimistic estimate of the true parameter vector $\boldsymbol{\gamma}(a)$:

$$\widetilde{\boldsymbol{\gamma}}_t(a) \in \operatorname*{arg\,max}_{\boldsymbol{\gamma} \in \mathcal{C}_t(a)} \langle \boldsymbol{\gamma}, \mathbf{z}_{t-1} \rangle.$$

By Theorem 5.4.1, we have that, for every action $a \in \mathcal{A}$ and round $t \in [\![T]\!]$, the true parameter vector satisfies $\boldsymbol{\gamma}(a) \in \mathcal{C}_t(a)$ with probability at least $1-\delta$. Therefore, with the same probability, we have:

$$\langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1} \rangle$$

$$
\begin{aligned}
&= \underbrace{\langle \boldsymbol{\gamma}(a_t^*) - \tilde{\boldsymbol{\gamma}}_t(a_t), \mathbf{z}_{t-1}\rangle}_{\leqslant 0} + \langle \tilde{\boldsymbol{\gamma}}_t(a_t) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle \\
&\leqslant \langle \tilde{\boldsymbol{\gamma}}_t(a_t) - \hat{\boldsymbol{\gamma}}_{t-1}(a_t), \mathbf{z}_{t-1}\rangle + \langle \hat{\boldsymbol{\gamma}}_{t-1}(a_t) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle \\
&\leqslant 2\beta_{t-1}(a_t)\|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(a)^{-1}},
\end{aligned}
$$

where the first inequality follows from the optimism and in the last passage we have used Cauchy-Schwartz inequality, recalling that for every couple of vectors $\mathbf{v}, \mathbf{w}$ it holds $\langle \mathbf{v}, \mathbf{w}\rangle \leqslant \|\mathbf{v}\|_{\mathbf{V}_{t-1}(a)}\|\mathbf{w}\|_{\mathbf{V}_{t-1}(a)^{-1}}$, and having observed that $\boldsymbol{\gamma}(a_t), \tilde{\boldsymbol{\gamma}}_t(a_t) \in \mathcal{C}_t(a_t)$.

Furthermore, we observe that the external regret:

$$
\begin{aligned}
\rho_t &= \langle \boldsymbol{\gamma}(a_t^*) - \boldsymbol{\gamma}(a_t), \mathbf{z}_{t-1}\rangle \\
&\leqslant \|\boldsymbol{z}_{t-1}\|_2 + m,
\end{aligned}
$$

since the coefficients γ_j for $j \neq 0$ have a sum bounded by $\Gamma < 1$ and get multiplied by $\mathbf{z}_{t-1}$, while γ_0, which is bounded by m gets multiplied by 1, then we have $\rho_t \leqslant L + m = \mathcal{O}(m)$. By Lemma B.0.1 with probability of at least $1 - \delta$ we have:

$$
\|\boldsymbol{z}_t\|_2 \leqslant \sqrt{1 + n\left(\frac{m+\eta}{1-\Gamma}\right)^2} =: L,
$$

where $\eta = \sqrt{2\sigma^2 \log(T/\delta)}$ and, consequently:

$$
\rho_t \leqslant m + L =: C_1.
$$

At this point, we proceed as follows:

$$
\begin{aligned}
\rho_t &\leqslant 2\min\{C_1, \beta_{t-1}(a_t)\|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(a_t)^{-1}}\} \\
&\leqslant 2\max\{C_1, \beta_{t-1}(a_t)\}\min\{1, \|\mathbf{z}_{t-1}\|_{\mathbf{V}_{t-1}(a_t)^{-1}}\}.
\end{aligned}
$$

Summing over $t \in [\![T]\!]$, we obtain a bound on the cumulative external regret:

$$
\begin{aligned}
\varrho(\texttt{AR-UCB}, T) &= \sum_{t=1}^{T} \rho_t \\
&= \sum_{t=1}^{T} 1 \cdot \rho_t \\
&\leqslant \sqrt{T\sum_{t=1}^{T} \rho_t^2}
\end{aligned}
$$

$$\leqslant 2\max\{C_1, \beta_{T-1}\}\sqrt{T\sum_{t=1}^{T}\min\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(a_t)^{-1}}\}}$$

where

$$\beta_{T-1} := \max_{a\in\mathcal{A}} \beta_{T-1}(a),$$

where the first inequality follows from an application of Cauchy-Schwartz inequality and the last passage holds since the sequence $\beta_t(a_t)$ is non-decreasing, and so we can bound each of them with their value at $t = T$. Now, we are finally able to use the *Elliptic Potential Lemma* (Lemma B.0.2):

$$\begin{aligned}\sum_{t=1}^{T}\min\{1, \|\mathbf{z}_{t-1}\|^2_{\mathbf{V}_{t-1}(a_t)^{-1}}\} &= \sum_{a\in\mathcal{A}}\sum_{l\in\mathcal{O}_T(a)}\min\{1, \|\mathbf{z}_{l-1}\|^2_{\mathbf{V}_{l-1}(a)^{-1}}\}\\ &\leqslant \sum_{a\in\mathcal{A}} 2(n+1)\log\left(\frac{\lambda(n+1)+|\mathcal{O}_T(a)|L^2}{\lambda(n+1)}\right)\\ &\leqslant 2n(n+1)\log\left(1+\frac{TL^2}{k\lambda(n+1)}\right),\end{aligned}$$

where the first inequality follows from an application of the elliptic potential lemma for each action $a \in \mathcal{A}$ observing that $\mathbf{V}_0 = \lambda\mathbf{I}_{n+1}$ and, consequently, tr$(\mathbf{V}_0) = \lambda(n+1)$ and $\det(\mathbf{V}_0)^{1/(n+1)} = \lambda$. The second inequality follows by observing that $\sum_{a\in\mathcal{A}}|\mathcal{O}_T(a)| = T$ and since the $\log$ is a concave function, the worst allocation of pulls is the uniform one. Now that we have bounded the inner summation, we can state that:

$$\begin{aligned}\varrho(\texttt{AR-UCB}, T) &= \sum_{t=1}^{T}\rho_t\\ &\leqslant 2\max\{C_1, \beta_{T-1}\}\sqrt{2Tk(n+1)\log\left(1+\frac{TL^2}{k\lambda(n+1)}\right)}.\end{aligned}$$

To conclude, we bound the term β_{T-1} as follows:

$$\begin{aligned}\beta_{T-1} &= \sqrt{\lambda(m^2+1)} + \sigma\max_{a\in\mathcal{A}}\sqrt{2\log\left(\frac{k}{\delta}\right)+\log\left(\frac{\det\mathbf{V}_{T-1}(a)}{\lambda^{n+1}}\right)}\\ &\leqslant \sqrt{\lambda(m^2+1)} + \sigma\sqrt{2\log\left(\frac{k}{\delta}\right)+(n+1)\log\left(\frac{\lambda(n+1)+TL^2}{\lambda(n+1)}\right)}.\end{aligned}$$

Therefore, by highlighting the dependences on m, n, σ, and Γ, we have:

$$\beta_{T-1} = \widetilde{\mathcal{O}}\left(m + \sigma\sqrt{n+1}\right), \qquad C_1 = \widetilde{\mathcal{O}}\left(1 + \sqrt{n}\frac{m+\sigma}{1-\Gamma}\right).$$

These results hold with probability $1-2\delta$. We set $\delta = (2T)^{-1}$. Putting all together, we obtain:

$$\varrho(\texttt{AR-UCB}, T) = \sum_{t=1}^{T} \rho_t \leqslant \widetilde{\mathcal{O}}\left(\frac{(m+\sigma)\sqrt{k(n+1)T}}{1-\Gamma}\right),$$

and, applying the previous Lemma 5.4.3, this results in:

$$R(\texttt{AR-UCB}, T) \leqslant \widetilde{\mathcal{O}}\left(\frac{(m+\sigma)(n+1)^{3/2}\sqrt{kT}}{(1-\Gamma)^2}\right).$$

□

B.1 Optimal Policy without Noise

In this section, we derive the optimal policy for the deterministic setting. In the case of no noise, our system writes:

$$x_t = \gamma_0(a_t) + \sum_{i=1}^{n} \gamma_i(a_t) x_{t-i}. \tag{B.1}$$

In this case, the process evolution is deterministic. Therefore, even if it is still true that the optimal policy is given by Theorem 5.2.1, it is possible to say that there is a constant policy that is asymptotically optimal, in the sense that its cumulative regret is bounded by a constant. This policy is given by:

$$a^{+} \in \arg\max_{a \in \mathcal{A}} \frac{\gamma_0(a_t)}{1 - \sum_{i=1}^{n} \gamma_i(a_t)}. \tag{B.2}$$

This result is not surprising. In fact, this action makes the process converge to the highest possible stationary reward, which is of course

$$\arg\max_{a \in \mathcal{A}} \frac{\gamma_0(a_t)}{1 - \sum_{i=1}^{n} \gamma_i(a_t)}.$$

Formally, the following result holds.

Theorem B.1.1. *Let us consider the problem formulation of Equation* (B.1). *Define:*

$$a^+ \in \arg\max_{a \in \mathcal{A}} \frac{\gamma_0(a_t)}{1 - \sum_{i=1}^n \gamma_i(a_t)},$$

as in Equation (B.2). *Then, there exist no policy* $\boldsymbol{\pi}$ *(even non-constant) such that:*

$$\limsup_{t \to +\infty} x_t^{\boldsymbol{\pi}} - x_t^* > 0$$

(where $x_t^{\boldsymbol{\pi}}$ *denotes the sequence obtained with policy* $\boldsymbol{\pi}$*, while* x_t^* *is the one relative to* a^+*). Moreover, the cumulative regret with respect to the actual optimal policy is bounded by:*

$$\gamma_0(a^+) \frac{n}{(1-\Gamma)^2}.$$

Proof. If we play always a^+, we have:

$$\limsup_{t \to +\infty} x_t^* = \frac{\gamma_0(a^+)}{1 - \sum_{i=1}^n \gamma_i(a^+)},$$

by imposing the condition of stationarity. For the rest of the proof, let us denote:

$$x^* := \frac{\gamma_0(a^+)}{1 - \sum_{i=1}^n \gamma_i(a^+)}.$$

Now, we prove that, for any policy $\boldsymbol{\pi}$, we cannot achieve an $x_t > x^*$. By contradiction, if $\limsup_{t \to \infty} x_t^{\boldsymbol{\pi}} - x_t^* > 0$, then the set $\{t \in \mathbb{N} \,:\, x_t > x^*\}$ is non-empty. Let $t_0 = \min\{t \in \mathbb{N} \,:\, x_t > x^*\}$. Then, by definition:

$$x_{t_0} = \gamma_0(a_{t_0}) + \sum_{i=1}^n \gamma_i(a_{t_0}) x_{t_0 - i}.$$

Recalling that t_0 is the first time in which we surpass x^*, we have:

$$x^* < x_{t_0} = \gamma_0(a_{t_0}) + \sum_{i=1}^n \gamma_i(a_{t_0}) x_{t_0-i} \leqslant \gamma_0(a_{t_0}) + \sum_{i=1}^n \gamma_i(a_{t_0}) x^*.$$

This inequality entails that:

$$\Big(1 - \sum_{i=1}^n \gamma_i(a_{t_0})\Big) x^* < \gamma_0(a_{t_0}),$$

and, therefore:

$$\frac{\gamma_0(a^+)}{1 - \sum_{i=1}^n \gamma_i(a^+)} = x^* < \frac{\gamma_0(a_{t_0})}{1 - \sum_{i=1}^n \gamma_i(a_{t_0})},$$

which contradicts the definition of a^+.

For the second part, we start considering that the regret obtained by using constant action a^+ is bounded by:

$$\sum_{t=1}^{+\infty} x^* - x_t,$$

since x^* is the maximum instantaneous reward that every policy can achieve. Now, note that $\gamma_0(a^+) > 0$, otherwise it could not be the optimal action. At this point, we have for $0 < t \leqslant n$ that $x_t \geqslant \gamma_0(a^+)$, by simply using the fact that all the coefficients of the autoregressive model are non-negative. From this fact we have for $n < t \leqslant 2n$ that $x_t \geqslant \gamma_0(a^+)(1 + \sum_{i=1}^n \gamma_i(a^+))$; and generalizing:

$$x_t \geqslant \gamma_0(a^+)\Big(\sum_{\ell=0}^{j}(\Gamma^+)^\ell\Big), \quad \forall j > 0 \quad \text{and} \quad jn - n < t \leqslant jn$$

with:

$$\Gamma^+ = \sum_{i=1}^{n} \gamma_i(a^+).$$

Therefore, we have $x_t \geqslant \gamma_0(a^+)\frac{1-\Gamma^{\lfloor t/n \rfloor}}{1-\Gamma}$, which means:

$$\begin{aligned}
R_t &\leqslant \sum_{t=1}^{+\infty} x^* - x_t \\
&\leqslant \sum_{t=1}^{+\infty} x^* - \gamma_0(a^+)\frac{1-\Gamma^{\lfloor t/n \rfloor}}{1-\Gamma} \\
&= \gamma_0(a^+)\sum_{t=1}^{+\infty} \frac{1}{1-\Gamma} - \frac{1-\Gamma^{\lfloor t/n \rfloor}}{1-\Gamma} \\
&= \gamma_0(a^+)\sum_{t=1}^{+\infty} \frac{\Gamma^{\lfloor t/n \rfloor}}{1-\Gamma} \\
&= \gamma_0(a^+)\frac{n}{(1-\Gamma)^2}.
\end{aligned}$$

□

APPENDIX C

Omitted Proofs of Chapter 7

In this appendix, we provide the proofs we have omitted in Chapter 7.

C.0.1 Proofs of Section 7.2

Before we proceed, we introduce a different notion of regret useful for analysis purposes, that we name *offline regret*. This notion of regret compares J^* with the steady-state performance of the action $\mathbf{u}_t = \boldsymbol{\pi}_t(H_{t-1})$ played at each round $t \in [\![T]\!]$ by the agent:

$$R^{\mathrm{off}}(\underline{\boldsymbol{\pi}}, T) := TJ^* - \sum_{t=1}^{T} J(\mathbf{u}_t). \tag{C.1}$$

We denote with $\mathbb{E}R^{\mathrm{off}}(\underline{\boldsymbol{\pi}}, T)$ the *expected offline regret*, where the expectation is taken w.r.t. the randomness of the reward. Clearly, the two notions of regret coincide when the system has no dynamics.
The following result relates the offline and the (online) expected regret.

Lemma C.0.1. *Under Assumptions 7.1 and 7.2, for any policy $\underline{\boldsymbol{\pi}}$, it holds that:*

$$\left|\mathbb{E}\, R^{\mathit{off}}(\underline{\boldsymbol{\pi}}, T) - \mathbb{E}\, R(\underline{\boldsymbol{\pi}}, T)\right| \leqslant \frac{\Omega\Phi(\mathbf{A})BU}{(1-\rho(\mathbf{A}))^2} + \frac{\Omega\Phi(\mathbf{A})X}{1-\rho(\mathbf{A})}.$$

Proof. First of all, we observe that for any policy, the cumulative effect of the noise components is zero-mean. Thus, it suffices to consider the deterministic evolution of the system. For every $t \in [\![T]\!]$, let us denote with $\mathbb{E}[y_t]$ the expected reward at time t and with $J(\mathbf{u}_t)$ as the steady-state performance when executing action $\mathbf{u}_t$:

$$\begin{aligned}
\mathbb{E}[y_t] &= \sum_{s=0}^{t-1} \langle \mathbf{h}^{\{s\}}, \mathbf{u}_{t-s} \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1} \mathbf{x}_1 \\
&= \boldsymbol{\theta}^{\mathrm{T}} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \sum_{s=1}^{t-1} \mathbf{A}^{s-1} \mathbf{B} \mathbf{u}_{t-s} + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1} \mathbf{x}_1, \\
J(\mathbf{u}_t) &= \boldsymbol{\theta}^{\mathrm{T}} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \left(\mathbf{I}_d - \mathbf{A} \right)^{-1} \mathbf{u}_t \\
&= \boldsymbol{\theta}^{\mathrm{T}} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \sum_{s=0}^{+\infty} \mathbf{A}^{s} \mathbf{u}_t.
\end{aligned}$$

We now proceed by summing over $t \in [\![T]\!]$. First of all, we consider the following preliminary result involving y_t, which is obtained by rearranging the summations:

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}[y_t] &= \boldsymbol{\theta}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \sum_{s=1}^{t-1} \mathbf{A}^{s-1} \mathbf{B} \mathbf{u}_{t-s} + \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{A}^{t-1} \mathbf{x}_1 \\
&= \boldsymbol{\theta}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T-1} \left(\sum_{s=0}^{T-t-1} \mathbf{A}^{s} \right) \mathbf{B} \mathbf{u}_t + \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{A}^{t-1} \mathbf{x}_1.
\end{aligned}$$

Thus, we have:

$$\begin{aligned}
\left| \sum_{t=1}^{T} \left(J(\mathbf{u}_t) - \mathbb{E}[y_t] \right) \right| &= \left| \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \left(\sum_{s=0}^{+\infty} \mathbf{A}^{s} - \sum_{s=0}^{T-t-1} \mathbf{A}^{s} \right) \mathbf{B} \mathbf{u}_t - \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{A}^{t-1} \mathbf{x}_1 \right| \\
&= \left| \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \left(\sum_{s=T-t}^{+\infty} \mathbf{A}^{s} \right) \mathbf{B} \mathbf{u}_t - \boldsymbol{\omega}^{\mathrm{T}} \sum_{t=1}^{T} \mathbf{A}^{t-1} \mathbf{x}_1 \right| \\
&\leqslant \Omega \Phi(\mathbf{A}) B U \sum_{t=1}^{T} \sum_{s=T-t}^{+\infty} \rho(\mathbf{A})^{s} + \Omega \Phi(\mathbf{A}) X \sum_{t=1}^{T} \rho(\mathbf{A})^{t-1} \qquad \text{(C.2)} \\
&\leqslant \frac{\Omega \Phi(\mathbf{A}) B U}{1 - \rho(\mathbf{A})} \sum_{t=1}^{T} \rho(\mathbf{A})^{T-t} + \frac{\Omega \Phi(\mathbf{A}) X}{1 - \rho(\mathbf{A})} \qquad \text{(C.3)} \\
&\leqslant \frac{\Omega \Phi(\mathbf{A}) B U}{(1 - \rho(\mathbf{A}))^2} + \frac{\Omega \Phi(\mathbf{A}) X}{1 - \rho(\mathbf{A})}, \qquad \text{(C.4)}
\end{aligned}$$

where line (C.2) follows from Assumptions 7.1 and 7.2, lines (C.3) and (C.4) follow from bounding the summations with the series. The result follows by observing that:

$$\mathbb{E}\, R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) - \mathbb{E}\, R(\underline{\boldsymbol{\pi}}, T) = \sum_{t=1}^{T} \left(J(\mathbf{u}_t) - \mathbb{E}[y_t] \right).$$

□

Theorem 7.2.2 (Lower Bound)**.** *For any policy $\underline{\boldsymbol{\pi}}$ (even stochastic), there exists a DLB fulfilling Assumptions 7.1 and 7.2, such that for sufficiently large $T \geqslant \mathcal{O}\left(\frac{d^2}{1-\rho(\mathbf{A})}\right)$, policy $\underline{\boldsymbol{\pi}}$ suffers an expected regret lower bounded by:*

$$\mathbb{E} R(\underline{\boldsymbol{\pi}}, T) \geqslant \Omega \left(\frac{d\sqrt{T}}{(1-\rho(\mathbf{A}))^{\frac{1}{2}}} \right).$$

Proof. To derive the lower bound, we take inspiration from the construction of Lattimore and Szepesvári (2020) for linear bandits (Theorem 24.1). We consider a class of DLBs defined in terms of fixed $0 \leqslant \rho < 1$ and $0 \leqslant \epsilon \leqslant \rho$ with $\boldsymbol{\omega} = \mathbf{1}_d$, $\boldsymbol{\theta} = -\frac{2(1-\rho)+\epsilon}{2(1-(\rho-\epsilon))}\mathbf{1}_d$, $\mathbf{B} = (1-\rho)\mathbf{I}_d$ and with a diagonal dynamical matrix $\mathbf{A} = \operatorname{diag}(\mathbf{a})$, defined in terms of the vector $\mathbf{a}$ belonging to the set $\mathcal{A} = \{\rho, \rho-\epsilon\}^d$. The available actions are $\mathcal{U} = \{-1, 1\}^d$. Let us note that $|\mathcal{A}| = |\mathcal{U}| = 2^d$. Thus, in our set of DLBs, the vector $\mathbf{a}$ fully characterizes the problem. Moreover, we observe that, given the diagonal $\mathbf{a} = \operatorname{diag}(\mathbf{A})$, we can compute the cumulative Markov parameter $\mathbf{h}_{\mathbf{a}} = \operatorname{sign}(\mathbf{a})\frac{\epsilon}{2(1-(\rho-\epsilon))}$.[1] As a consequence the optimal action can be defined as $\mathbf{u}^*_{\mathbf{a}} = \operatorname{sign}(\mathbf{a})$, whose performance is given by $J^*_{\mathbf{a}} = \langle \mathbf{h}_{\mathbf{a}}, \mathbf{u}^*_{\mathbf{a}} \rangle = \frac{\epsilon d}{2(1-(\rho-\epsilon))}$. Let us consider the probability distribution over the canonical bandit model induced by executing a policy $\underline{\boldsymbol{\pi}}$ in a DLB characterized by the diagonal of the dynamical matrix $\mathbf{a} \in \mathcal{A}$ and with Gaussian diagonal noise:

$$\mathbb{P}_{\mathbf{a}} = \prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_{t+1} | \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \sigma^2 \mathbf{I}_d) \mathcal{N}(y_t | \langle \boldsymbol{\theta}, \mathbf{u}_t \rangle + \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle, \sigma^2) \pi_t(\mathbf{u}_t | H_{t-1}),$$

where H_{t-1} is the history of observations up to time $t-1$. We denote with $\mathbb{E}_{\mathbf{a}}$ the expectation induced by the distribution $\mathbb{P}_{\mathbf{a}}$. For every $i \in [\![d]\!]$, let

[1] For a vector $\mathbf{v} \in \mathbb{R}^d$, we denote with $\operatorname{sign}(\mathbf{v}) \in \{-1, 1\}^d$ the vector of the signs of the components of $\mathbf{v}$. It is irrelevant how we convene to define the sign of 0.

us now consider an alternative DLB instance that differs on the dynamical matrix only. Specifically:

$$\mathbf{a}'_j = \begin{cases} \mathbf{a}_j & \text{if } j \neq i \\ \rho & \text{if } j = i \text{ and } \mathbf{a}_j = \rho - \epsilon \\ \rho - \epsilon & \text{if } j = i \text{ and } \mathbf{a}_j = \rho \end{cases}, \qquad \forall j \in [\![d]\!].$$

By relative entropy identities (Lattimore and Szepesvári, 2020), let $\mathbf{A} = \operatorname{diag}(\mathbf{a})$ and $\mathbf{A}' = \operatorname{diag}(\mathbf{a}')$, we have:

$$\begin{aligned} & D_{\mathrm{KL}}\left(\mathbb{P}_{\mathbf{a}}, \mathbb{P}_{\mathbf{a}'}\right) \\ & = \mathbb{E}_{\mathbf{a}}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\left(\mathcal{N}(\cdot|\mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \sigma^2 \mathbf{I}_d), \mathcal{N}(\cdot|\mathbf{A}'\mathbf{x}_t + \mathbf{B}\mathbf{u}_t, \sigma^2 \mathbf{I}_d)\right)\right] \\ & = \frac{1}{2\sigma^2}\sum_{t=1}^{T} \mathbb{E}_{\mathbf{a}}\left[\left\|\left(\mathbf{A} - \mathbf{A}'\right)\mathbf{x}_t\right\|_2^2\right] = \epsilon^2 \mathbb{E}_{\mathbf{a}}\left[\mathbf{x}_{t,i}^2\right]. \end{aligned}$$

We proceed at properly bounding the KL-divergence, letting $\mathbf{e}_i$ be the i-th vector of the canonical basis of $\mathbb{R}^d$ and convening that $\mathbf{x}_0 = \mathbf{0}_d$:

$$\begin{aligned} \mathbb{E}_{\mathbf{a}}\left[\mathbf{x}_{t,i}^2\right] & = \mathbb{E}_{\mathbf{a}}\left[\left(\sum_{s=1}^{t-1} \mathbf{e}_i^{\mathrm{T}} \mathbf{A}^s \mathbf{B} \mathbf{u}_{t-s} + \sum_{s=1}^{t-1} \mathbf{e}_i^{\mathrm{T}} \mathbf{A}^s \boldsymbol{\epsilon}_{t-s}\right)^2\right] \\ & = \mathbb{E}_{\mathbf{a}}\left[\left((1-\rho)\sum_{s=1}^{t-1} \mathbf{a}_i^s \mathbf{u}_{t-s,i} + \sum_{s=1}^{t-1} \mathbf{a}_i^s \boldsymbol{\epsilon}_{t-s,i}\right)^2\right] \\ & = \mathbb{E}_{\mathbf{a}}\Bigg[\underbrace{(1-\rho)^2 \sum_{s=1}^{t-1}\sum_{l=1}^{t-1} \mathbf{a}_i^{s+l} \mathbf{u}_{t-s,i} \mathbf{u}_{t-l,i}}_{\text{(a)}} + \\ & \qquad + \underbrace{2(1-\rho)\sum_{s=1}^{t-1}\sum_{l=1}^{t-1} \mathbf{a}_i^{s+l} \mathbf{u}_{t-s,i} \boldsymbol{\epsilon}_{t-l,i}}_{\text{(b)}} + \\ & \qquad + \underbrace{\sum_{s=1}^{t-1}\sum_{l=1}^{t-1} \mathbf{a}_i^{s+l} \boldsymbol{\epsilon}_{t-s,i} \boldsymbol{\epsilon}_{t-l,i}}_{\text{(c)}}\Bigg] \end{aligned}$$

Let us start with (a):

$$(1-\rho)^2\mathbb{E}_{\mathbf{a}}\left[\sum_{s=1}^{t-1}\sum_{l=1}^{t-1}\mathbf{a}_i^{s+l}\mathbf{u}_{t-s,i}\mathbf{u}_{t-l,i}\right] \leqslant (1-\rho)^2\sum_{s=1}^{t-1}\sum_{l=1}^{t-1}\rho^{s+l} \leqslant 1,$$

having observed that $|\mathbf{u}_{t-s,i}|, |\mathbf{u}_{t-l,i}| \leqslant 1$, that $|\mathbf{a}_i| \leqslant \rho$, and bounding the summations with the series.
Let us move to (b):

$$\begin{aligned}
(1-\rho)\mathbb{E}_{\mathbf{a}}&\left[\sum_{s=1}^{t-1}\sum_{l=1}^{t-1}\mathbf{a}_i^{s+l}\mathbf{u}_{t-s,i}\boldsymbol{\epsilon}_{t-l,i}\right]\\
&= (1-\rho)\mathbb{E}_{\mathbf{a}}\left[\sum_{s=1}^{t-1}\sum_{l=s+1}^{t-1}\mathbf{a}_i^{s+l}\mathbf{u}_{t-s,i}\boldsymbol{\epsilon}_{t-l,i}\right]\\
&\quad + \cancelto{0}{(1-\rho)\,\mathbb{E}_{\mathbf{a}}\left[\sum_{l=1}^{t-1}\sum_{s=l}^{t-1}\mathbf{a}_i^{s+l}\mathbf{u}_{t-s,i}\boldsymbol{\epsilon}_{t-l,i}\right]}\\
&\leqslant (1-\rho)\sum_{s=1}^{t-1}\sum_{l=s+1}^{t-1}\rho^{s+l}\mathbb{E}_{\mathbf{a}}\left[|\boldsymbol{\epsilon}_{t-l,i}|\right]\\
&\leqslant \frac{\sigma}{1-\rho}\sqrt{\frac{2}{\pi}},
\end{aligned}$$

having observed that $\mathbf{u}_{t-s,i}$ and $\boldsymbol{\epsilon}_{t-l,i}$ are independent when $s \geqslant l$ and $\boldsymbol{\epsilon}_{t-l,i}$ has zero mean, that $|\mathbf{u}_{t-s,i}| \leqslant 1$, that $\mathbf{a}_i^{s+l} \leqslant \rho^{s+l}$, and that the expectation of the absolute value of random variable normally distributed is given by $\mathbb{E}\left[|\boldsymbol{\epsilon}_{t-l,i}|\right] = \sigma\sqrt{\frac{2}{\pi}}$.
Finally, let us consider (c):

$$\begin{aligned}
\mathbb{E}_{\mathbf{a}}&\left[\sum_{s=1}^{t-1}\sum_{l=1}^{t-1}\mathbf{a}_i^{s+l}\boldsymbol{\epsilon}_{t-s,i}\boldsymbol{\epsilon}_{t-l,i}\right]\\
&= \mathbb{E}_{\mathbf{a}}\left[\sum_{s=1}^{t-1}\mathbf{a}_i^{2s}\boldsymbol{\epsilon}_{t-s,i}\boldsymbol{\epsilon}_{t-s,i}\right] + \cancelto{0}{2\,\mathbb{E}_{\mathbf{a}}\left[\sum_{s=1}^{t-2}\sum_{l=s+1}^{t-1}\mathbf{a}_i^{s+l}\boldsymbol{\epsilon}_{t-s,i}\boldsymbol{\epsilon}_{t-l,i}\right]}\\
&\leqslant \sigma^2\sum_{s=1}^{t-1}\rho^{2s} \leqslant \frac{\sigma^2}{1-\rho^2} \leqslant \frac{\sigma^2}{1-\rho},
\end{aligned}$$

having observed that the noise vectors $\boldsymbol{\epsilon}_{t-l,i}$ and $\boldsymbol{\epsilon}_{t-s,i}$ are independent whenever $s \neq l$, that $\mathbb{E}_{\mathbf{a}}[\boldsymbol{\epsilon}_{t-s,i}^2] = \sigma^2$, and having bounded the sum with the series.

Coming back to the original bound, we have:

$$\mathbb{E}_{\mathbf{a}}\left[\mathbf{x}_{t,i}^2\right] \leqslant 1 + \frac{1}{1-\rho}\left(\sigma^2 + 2\sigma\sqrt{\frac{2}{\pi}}\right).$$

For $i \in [\![d]\!]$ and $\mathbf{a} \in \mathcal{A}$, we introduce the symbol:

$$p_{\mathbf{a},i} = \mathbb{P}_{\mathbf{a}}\left(\sum_{t=1}^{T} \mathbf{1}\{\operatorname{sign}(\mathbf{u}_{t,i}) \neq \operatorname{sign}(\mathbf{h}_{\mathbf{a},i})\} \geqslant \frac{T}{2}\right).$$

Thus, for $\mathbf{a}$ and $\mathbf{a}'$ defined as above, by the Bretagnolle-Huber inequality (Lattimore and Szepesvári, 2020, Theorem 14.2), we have:

$$\begin{aligned}
p_{\mathbf{a},i} + p_{\mathbf{a}',i} &\geqslant \frac{1}{2}\exp\left(-D_{\mathrm{KL}}\left(\mathbb{P}_{\mathbf{a}}, \mathbb{P}_{\mathbf{a}'}\right)\right) \\
&= \frac{1}{2}\exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{T}\mathbb{E}_{\mathbb{P}}\left[\left\|\left(\mathbf{A}-\mathbf{A}'\right)\mathbf{x}_t\right\|_2^2\right]\right) \\
&\geqslant \frac{1}{2}\exp\left(-\frac{T\epsilon^2}{2}\left(\frac{1}{\sigma^2} + \frac{1}{1-\rho}\left(1 + \frac{2}{\sigma}\sqrt{\frac{2}{\pi}}\right)\right)\right) \\
&\geqslant \frac{1}{2}\exp\left(-\frac{2T\epsilon^2}{1-\rho}\right),
\end{aligned}$$

having selected $\sigma^2 = 1$. We use the notation $\sum_{\mathbf{a}_{-i}}$ to denote the multiple summation $\sum_{\mathbf{a}_1,\dots,\mathbf{a}_{i-1},\mathbf{a}_{i+1},\dots,\mathbf{a}_d \in \{\rho,\rho-\epsilon\}^{d-1}}$:

$$\begin{aligned}
\sum_{\mathbf{a}\in\mathcal{A}} 2^{-d}\sum_{i=1}^{d} p_{\mathbf{a},i} &= \sum_{i=1}^{d}\sum_{\mathbf{a}_{-i}} 2^{-d} \sum_{\mathbf{a}_i \in \{\rho,\rho-\epsilon\}} p_{\mathbf{a},i} \\
&\geqslant \sum_{i=1}^{d}\sum_{\mathbf{a}_{-i}} 2^{-d}\cdot\frac{1}{2}\exp\left(-\frac{2T\epsilon^2}{1-\rho}\right) \\
&= \frac{d}{4}\exp\left(-\frac{2T\epsilon^2}{1-\rho}\right).
\end{aligned}$$

Therefore, with this averaging argument, we can conclude that there exists $\mathbf{a}^* \in \mathcal{A}$ such that $\sum_{i=1}^{d} p_{\mathbf{a}*,i} \geqslant \frac{d}{4}\exp\left(-\frac{2T\epsilon^2}{1-\rho}\right)$. For this choice $\mathbf{a}^*$, we consider $\mathbf{u}^*_{\mathbf{a}*} = \operatorname{sign}(\mathbf{a}^*) \in \mathcal{U}$, we can proceed to the lower bound on the expected offline regret:

$$\mathbb{E}R^{\mathrm{off}}(\underline{\boldsymbol{\pi}}, T)$$

$$
\begin{aligned}
&= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{a}*} \left[\langle \mathbf{h}_{\mathbf{a}*}, \mathbf{u}^*_{\mathbf{a}*} - \mathbf{u}_t \rangle \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{a}*} \left[\sum_{i=1}^{d} \mathbf{1}\{\operatorname{sign}(\mathbf{u}_{t,i}) \neq \operatorname{sign}(\mathbf{h}_{\mathbf{a}*,i})\} \frac{\epsilon}{1-(\rho-\epsilon)} \right] \\
&= \frac{\epsilon}{1-(\rho-\epsilon)} \sum_{t=1}^{T} \sum_{i=1}^{d} \mathbb{P}_{\mathbf{a}*} \left(\operatorname{sign}(\mathbf{u}_{t,i}) \neq \operatorname{sign}(\mathbf{h}_{\mathbf{a}*,i}) \right) \\
&\geqslant \frac{T\epsilon}{2(1-(\rho-\epsilon))} \sum_{i=1}^{d} \mathbb{P}_{\mathbf{a}*} \left(\sum_{t=1}^{T} \mathbf{1}\{\operatorname{sign}(\mathbf{u}_{t,i}) \neq \operatorname{sign}(\mathbf{h}_{\mathbf{a}*,i})\} \geqslant \frac{T}{2} \right) \\
&= \frac{T\epsilon}{2(1-(\rho-\epsilon))} \sum_{i=1}^{d} p_{\mathbf{a}*,i} \geqslant \frac{Td\epsilon}{8(1-(\rho-\epsilon))} \exp\left(-\frac{2T\epsilon^2}{1-\rho} \right).
\end{aligned}
$$

We now maximize over $0 \leqslant \epsilon < \rho$. To this end, we perform the substitution $\epsilon = \frac{(1-\rho)\widetilde{\epsilon}}{1-\widetilde{\epsilon}}$, with $0 \leqslant \widetilde{\epsilon} \leqslant \rho$:

$$
\begin{aligned}
\frac{Td\epsilon}{8(1-(\rho-\epsilon))} \exp\left(-\frac{2T\epsilon^2}{1-\rho} \right) &= \frac{Td\widetilde{\epsilon}}{8} \exp\left(-\frac{2\widetilde{\epsilon}^2 T(1-\rho)}{(1-\widetilde{\epsilon})^2} \right) \\
&\geqslant \frac{Td\widetilde{\epsilon}}{8} \exp\left(-8\widetilde{\epsilon}^2 T(1-\rho) \right),
\end{aligned}
$$

where the last inequality holds for $\widetilde{\epsilon} \leqslant \frac{1}{2}$. We not take $\widetilde{\epsilon} = \frac{1}{\sqrt{8T(1-\rho)}}$ which is smaller than $\frac{1}{2}$ if $T \geqslant \frac{1}{2(1-\rho)}$, to get:

$$
\mathbb{E}R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) \geqslant \frac{d\sqrt{T}}{\sqrt{512e(1-\rho)}}.
$$

Notice that with this choice of $\widetilde{\epsilon}$ (and, consequently, of ϵ), for sufficiently large T, we fulfill Assumption 7.2. Indeed:

$$
\boldsymbol{\theta} = -1 + \frac{1}{\sqrt{32T(1-\rho)}}, \quad J^*_{\mathbf{a}} = \frac{d}{\sqrt{32T(1-\rho)}}.
$$

Thus, we require $T \geqslant \mathcal{O}\left(\frac{d^2}{1-\overline{\rho}} \right)$. Finally, to convert this result to the expected regret, we employ Lemma C.0.1:

$$
\mathbb{E}R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) \geqslant \mathbb{E}R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) - \frac{d}{1-\rho}.
$$

Under the constraint $T \geqslant \mathcal{O}\left(\frac{d^2}{1-\overline{\rho}}\right)$, we observe that:

$$\mathbb{E}R^{\text{off}}(\underline{\boldsymbol{\pi}}, T) \geqslant \Omega\left(\frac{d\sqrt{T}}{(1-\rho)^{\frac{1}{2}}}\right).$$

□

Theorem 7.2.1 (Optimal Policy)**.** *Under Assumptions 7.1 and 7.2, an optimal policy $\underline{\boldsymbol{\pi}}^*$ maximizing the (infinite-horizon) expected average reward $J(\underline{\boldsymbol{\pi}})$ (Equation 7.2), for every round $t \in \mathbb{N}$ and history $H_{t-1} \in \mathcal{H}_{t-1}$ is given by:*

$$\boldsymbol{\pi}_t^*(H_{t-1}) = \mathbf{u}^* \qquad \textit{where} \qquad \mathbf{u}^* \in \operatorname*{arg\,max}_{\mathbf{u} \in \mathcal{U}} J(\mathbf{u}) = \langle \mathbf{h}, \mathbf{u} \rangle. \quad (7.4)$$

Proof. Referring to the notation of Appendix C.1, we first observe that for every policy $\underline{\boldsymbol{\pi}}$, we have $J(\underline{\boldsymbol{\pi}}) = \liminf_{H \to +\infty} J_H(\underline{\boldsymbol{\pi}})$, where $J_H(\underline{\boldsymbol{\pi}}) = \frac{1}{H}\mathbb{E}[\sum_{t=1}^H y_t]$, is the H-horizon expected average reward. Let us start with Equation (C.12), a fixed finite $H \in \mathbb{N}$, and considering the sequence of actions $(\mathbf{u}_1, \mathbf{u}_2, \dots)$ generated by policy $\underline{\boldsymbol{\pi}}$:

$$\begin{aligned} J_H(\underline{\boldsymbol{\pi}}) &= \frac{1}{H}\sum_{s=1}^H \langle \mathbf{h}^{[\![0,H-s]\!]}, \mathbb{E}[\mathbf{u}_s] \rangle + \frac{1}{H}\sum_{t=1}^H \boldsymbol{\omega}^\mathsf{T}\mathbf{A}^{t-1}\,\mathbb{E}[\mathbf{x}_1] \\ &= \frac{1}{H}\sum_{s=1}^H \langle \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle - \frac{1}{H}\sum_{s=1}^H \langle \mathbf{h}^{[\![H-s+1,+\infty)\!)}, \mathbb{E}[\mathbf{u}_s] \rangle \\ &\qquad + \frac{1}{H}\sum_{t=1}^H \boldsymbol{\omega}^\mathsf{T}\mathbf{A}^{t-1}\,\mathbb{E}[\mathbf{x}_1]. \end{aligned}$$

Now, we consider two bounds on $J_H(\underline{\boldsymbol{\pi}})$, obtained by an application of Cauchy-Schwarz inequality on the second addendum:

$$\begin{aligned} J_H(\underline{\boldsymbol{\pi}}) &\leqslant \frac{1}{H}\sum_{s=1}^H \langle \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle + \frac{1}{H}\sum_{s=1}^H \left\| \mathbf{h}^{[\![H-s+1,+\infty)\!)} \right\|_2 \|\,\mathbb{E}[\mathbf{u}_s]\|_2 \\ &\qquad + \frac{1}{H}\sum_{t=1}^H \boldsymbol{\omega}^\mathsf{T}\mathbf{A}^{t-1}\,\mathbb{E}[\mathbf{x}_1] =: J_H^\uparrow(\underline{\boldsymbol{\pi}}), \\ J_H(\underline{\boldsymbol{\pi}}) &\geqslant \frac{1}{H}\sum_{s=1}^H \langle \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle - \frac{1}{H}\sum_{s=1}^H \left\| \mathbf{h}^{[\![H-s+1,+\infty)\!)} \right\|_2 \|\,\mathbb{E}[\mathbf{u}_s]\|_2 \end{aligned}$$

$$+ \frac{1}{H} \sum_{t=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1} \mathbb{E}[\mathbf{x}_1] =: J_H^{\downarrow}(\underline{\boldsymbol{\pi}}).$$

Concerning the term $\| \mathbb{E}[\mathbf{u}_s]\|_2$, we have that $\| \mathbb{E}[\mathbf{u}_s]\|_2 \leqslant \mathbb{E}[\|\mathbf{u}_s\|_2] \leqslant U$, having used Jensen's inequality and under Assumption 7.2. Regarding the second term, using Assumptions 7.1 and 7.2, we obtain:

$$\begin{aligned}
\left\|\mathbf{h}^{[\![H-s+1,+\infty)\!)}\right\|_2 &= \left\| \sum_{l=H-s+1}^{+\infty} \mathbf{B}^{\mathrm{T}} (\mathbf{A}^{l-1})^{\mathrm{T}} \boldsymbol{\omega} \right\|_2 \\
&\leqslant B\Omega \sum_{l=H-s+1}^{+\infty} \Phi(\mathbf{A}) \rho(\mathbf{A})^{l-1} \\
&= B\Omega\Phi(\mathbf{A}) \frac{\rho(\mathbf{A})^{H-s}}{1-\rho(\mathbf{A})}. \qquad \text{(C.5)}
\end{aligned}$$

Plugging this result into the summation over s, we obtain:

$$\frac{1}{H} \cdot \frac{B\Omega\Phi(\mathbf{A})}{1-\rho(\mathbf{A})} \sum_{s=1}^{H} \rho(\mathbf{A})^{H-s} = \frac{B\Omega\Phi(\mathbf{A})(1-\rho(\mathbf{A})^H)}{H(1-\rho(\mathbf{A}))^2}.$$

It is simple to observe that the last term approaches zero as $H \to +\infty$. Moreover, with an analogous argument, it can be proved that:

$$\left\| \frac{1}{H} \sum_{t=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1} \mathbb{E}[\mathbf{x}_1] \right\|_2 \to 0,$$

as $H \to +\infty$.
Thus, we have that:

$$\liminf_{H\to+\infty} J_H^{\downarrow}(\underline{\boldsymbol{\pi}}) = \liminf_{H\to+\infty} J_H^{\uparrow}(\underline{\boldsymbol{\pi}}).$$

Consequently, by the squeezing theorem of limits, we have:

$$\begin{aligned}
J(\underline{\boldsymbol{\pi}}) &= \liminf_{H\to+\infty} J_H^{\uparrow}(\underline{\boldsymbol{\pi}}) = \liminf_{H\to+\infty} J_H^{\downarrow}(\underline{\boldsymbol{\pi}}) \\
&= \liminf_{H\to+\infty} \frac{1}{H} \sum_{s=1}^{H} \langle \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle = \mathbf{h}^{\mathrm{T}} \left(\liminf_{H\to+\infty} \frac{1}{H} \sum_{s=1}^{H} \mathbb{E}[\mathbf{u}_s] \right).
\end{aligned}$$

It follows that an optimal policy is a policy that plays the constant action $\mathbf{u}^* \in \arg\max_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{h}, \mathbf{u} \rangle$. □

C.0.2 Proofs of Section 7.3

Theorem 7.3.1 (Self-Normalized Concentration). *Let $(\widehat{\mathbf{h}}_t)_{t\in\mathbb{N}}$ be the sequence of solutions of the Ridge regression problems of Algorithm 7.1. Then, under Assumption 7.1 and 7.2, for every $\lambda \geqslant 0$ and $\delta \in (0,1)$, with probability at least $1-\delta$, simultaneously for all rounds $t \in \mathbb{N}$, it holds that:*

$$\left\|\widehat{\mathbf{h}}_t - \mathbf{h}\right\|_{\mathbf{V}_t} \leqslant \frac{c_1}{\sqrt{\lambda}}\log(e(t+1)) + c_2\sqrt{\lambda} + \sqrt{2\widetilde{\sigma}^2\left(\log\left(\frac{1}{\delta}\right)+\frac{1}{2}\log\left(\frac{\det\left(\mathbf{V}_t\right)}{\lambda^d}\right)\right)},$$

where:

$$c_1 = U\Omega\Phi(\mathbf{A})\left(\frac{UB}{1-\rho(\mathbf{A})}+X\right),$$
$$c_2 = \Theta + \frac{\Omega B\Phi(\mathbf{A})}{1-\rho(\mathbf{A})},$$
$$\widetilde{\sigma}^2 = \sigma^2\left(1+\frac{\Omega^2\Phi(\mathbf{A})^2}{1-\rho(\mathbf{A})^2}\right).$$

Proof. First of all, let us properly relate the round $t \in [\![T]\!]$ and the index of the epoch $m \in [\![M]\!]$. For every epoch $m \in [\![M]\!]$, we denote with t_m the last round of epoch m (i.e., the one in which we update the relevant matrices $\mathbf{V}_t$ and $\mathbf{b}_t$):[2]

$$t_0 = 0, \qquad t_m = t_{m-1} + 1 + H_m.$$

We now proceed to define suitable filtrations. Let $\mathbb{F} = (\mathcal{F}_t)_{t\in[\![T]\!]}$ such that for every $t \geqslant 1$, the random variables $\{\mathbf{u}_1, y_1, \dots, \mathbf{u}_{t-1}, y_{t-1}, \mathbf{u}_t\}$ are $\mathcal{F}_{t-1}$-measurable, i.e., $\mathcal{F}_{t-1} = \sigma(\mathbf{u}_1, y_1, \dots, \mathbf{u}_{t-1}, y_{t-1}, \mathbf{u}_t)$. Let us also consider the filtration indexed by m, denoted with $\widetilde{\mathbb{F}} = (\widetilde{\mathcal{F}}_m)_{m\in[\![M]\!]}$ and defined for all $m \in [\![M]\!]$ as $\widetilde{\mathcal{F}}_m = \mathcal{F}_{t_{m+1}-1}$. Thus, the random variables $\widetilde{\mathcal{F}}_{m-1}$-measurable are those realized until the end of epoch m except for y_{t_m}.
Since the estimates $\widehat{\mathbf{h}}_t$ do not change within an epoch, we need to guarantee the statement for all rounds $\{t_m\}_{m\in[\![M]\!]}$ only. For these rounds, we define the following quantities:

$\widetilde{y}_m = y_{t_m}$,
$\widetilde{\mathbf{u}}_m = \mathbf{u}_{t_m}$, (or any $\mathbf{u}_l$ with $l \in [\![t_{m-1}+1, t_m]\!]$ since they are all equal)

[2] It is worth noting that the variables t_m are deterministic.

$$
\begin{aligned}
&\widetilde{\xi}_m = \eta_{t_m} + \sum_{s=1}^{H_m+1} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \boldsymbol{\epsilon}_{t_m-s}, \\
&\widetilde{\mathbf{x}}_{m-1} = \mathbf{x}_{t_{m-1}}, \\
&\widetilde{\mathbf{h}}_m = \widehat{\mathbf{h}}_{t_m}, \\
&\widetilde{\mathbf{V}}_m = \mathbf{V}_{t_m}, \\
&\widetilde{\mathbf{b}}_m = \mathbf{b}_{t_m}.
\end{aligned}
$$

We prove that $(\widetilde{\xi}_m)_{m \in [\![M]\!]}$ is a martingale difference process adapted to the filtration $\widetilde{\mathbb{F}}$. To this end, we recall that, by construction, $(\eta_t)_{t \in [\![T]\!]}$ and $(\boldsymbol{\epsilon}_t)_{t \in [\![T]\!]}$ are martingale difference processes adapted to the filtration $\mathbb{F}$. It is clear that $\widetilde{\xi}_m$ is $\mathcal{F}_m$-measurable and, being σ^2-subgaussian it is absolutely integrable. Furthermore, using the tower law of expectation:

$$
\begin{aligned}
\mathbb{E}\left[\widetilde{\xi}_m | \widetilde{\mathcal{F}}_{m-1}\right] &= \mathbb{E}\left[\eta_{t_m} + \sum_{s=1}^{H_m+1} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \boldsymbol{\epsilon}_{t_m-s} | \mathcal{F}_{t_m-1}\right] \\
&= \mathbb{E}\left[\eta_{t_m} | \mathcal{F}_{t_m-1}\right] \\
&\quad + \mathbb{E}\left[\sum_{s=1}^{H_m+1} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \, \mathbb{E}[\boldsymbol{\epsilon}_{t_m-s} | \mathcal{F}_{t_m-s-1}] | \mathcal{F}_{t_m-1}\right] \\
&= 0,
\end{aligned}
$$

since the system is operating by persisting the action after having decided it at the beginning of the epoch. Thus, by exploiting the decomposition in Equation (7.1), we can write:

$$
\begin{aligned}
\widetilde{y}_m &= y_{t_m} \\
&= \langle \mathbf{h}^{[\![0,H_m+1]\!]}, \widetilde{\mathbf{u}}_m \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_m+1} \mathbf{x}_{t_{m-1}} + \eta_{t_m} + \sum_{s=1}^{H_m+1} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \boldsymbol{\epsilon}_{t_m-s} \\
&= \langle \mathbf{h}^{[\![0,H_m+1]\!]}, \widetilde{\mathbf{u}}_m \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_m+1} \widetilde{\mathbf{x}}_{m-1} + \widetilde{\xi}_m \\
&= \langle \mathbf{h}, \widetilde{\mathbf{u}}_m \rangle - \langle \mathbf{h}^{[\![H_m+2,\infty)\!)}, \widetilde{\mathbf{u}}_m \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_m+1} \widetilde{\mathbf{x}}_{m-1} + \widetilde{\xi}_m,
\end{aligned} \tag{C.6}
$$

where we simply exploit the identity $\mathbf{h} = \mathbf{h}^{[\![0,H_m+1]\!]} + \mathbf{h}^{[\![H_m+2,\infty)\!)}$. We now introduce the following vectors and matrices:

$$
\widetilde{\mathbf{U}}_m = \begin{pmatrix} \widetilde{\mathbf{u}}_1^{\mathrm{T}} \\ \vdots \\ \widetilde{\mathbf{u}}_m^{\mathrm{T}} \end{pmatrix} \in \mathbb{R}^{m \times d}, \qquad \widetilde{\mathbf{y}}_m = \begin{pmatrix} \widetilde{y}_1 \\ \vdots \\ \widetilde{y}_m \end{pmatrix} \in \mathbb{R}^{m},
$$

$$\widetilde{\boldsymbol{\xi}}_m = \begin{pmatrix} \widetilde{\xi}_1 \\ \vdots \\ \widetilde{\xi}_m \end{pmatrix} \in \mathbb{R}^m, \qquad \widetilde{\boldsymbol{\nu}}_m = \begin{pmatrix} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_1+2} \widetilde{\mathbf{x}}_0 \\ \vdots \\ \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_m+2} \widetilde{\mathbf{x}}_{m-1} \end{pmatrix} \in \mathbb{R}^m,$$

$$\widetilde{\mathbf{g}}_m = \begin{pmatrix} \langle \mathbf{h}^{\llbracket H_1+1,\infty \rrparenthesis}, \widetilde{\mathbf{u}}_1 \rangle \\ \vdots \\ \langle \mathbf{h}^{\llbracket H_m+1,\infty \rrparenthesis}, \widetilde{\mathbf{u}}_m \rangle \end{pmatrix} \in \mathbb{R}^m.$$

Using the vectors and matrices above, we observe that $\widetilde{\mathbf{V}}_m = \lambda \mathbf{I} + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m$ and $\widetilde{\mathbf{b}}_m = \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{y}}_m$. Furthermore, by exploiting Equation (C.6), we can write:

$$\widetilde{\mathbf{y}}_m = \widetilde{\mathbf{U}}_m \mathbf{h} - \widetilde{\mathbf{g}}_m + \widetilde{\boldsymbol{\nu}}_m + \widetilde{\boldsymbol{\xi}}_m.$$

Let us consider the estimate at $m \in \llbracket M \rrbracket$:

$$\begin{aligned} \widetilde{\mathbf{h}}_m &= \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{b}}_m \\ &= \left(\lambda \mathbf{I} + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \right)^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{y}}_m \\ &= \left(\lambda \mathbf{I} + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \right)^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \left(\widetilde{\mathbf{U}}_m \mathbf{h} - \widetilde{\mathbf{g}}_m + \widetilde{\boldsymbol{\nu}}_m + \widetilde{\boldsymbol{\xi}}_m \right) \\ &= \mathbf{h} + \left(\lambda \mathbf{I} + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \right)^{-1} \left(-\lambda \mathbf{h} - \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{g}}_m + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m + \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\xi}}_m \right). \end{aligned}$$

We now proceed at bounding the $\| \cdot \|_{\widetilde{\mathbf{V}}_m}$-norm, and exploit the triangle inequality:

$$\begin{aligned} \left\| \widetilde{\mathbf{h}}_m - \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_m} &\leqslant \lambda \left\| \widetilde{\mathbf{V}}_m^{-1} \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_m} + \left\| \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{g}}_m \right\|_{\widetilde{\mathbf{V}}_m} \\ &\quad + \left\| \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m \right\|_{\widetilde{\mathbf{V}}_m} + \left\| \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\xi}}_m \right\|_{\widetilde{\mathbf{V}}_m} \\ &= \underbrace{\lambda \left\| \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_m^{-1}}}_{\text{(a)}} + \underbrace{\left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{g}}_m \right\|_{\widetilde{\mathbf{V}}_m^{-1}}}_{\text{(b)}} + \underbrace{\left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m \right\|_{\widetilde{\mathbf{V}}_m^{-1}}}_{\text{(c)}} + \underbrace{\left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\xi}}_m \right\|_{\widetilde{\mathbf{V}}_m^{-1}}}_{\text{(d)}}, \end{aligned}$$

where we simply exploited the identity $\|\mathbf{V}^{-1}\mathbf{x}\|_{\mathbf{V}}^2 = \mathbf{x}^{\mathrm{T}} \mathbf{V}^{-1} \mathbf{V} \mathbf{V}^{-1} \mathbf{x} = \mathbf{x}^{\mathrm{T}} \mathbf{V}^{-1} \mathbf{x} = \|\mathbf{x}\|_{\mathbf{V}^{-1}}^2$. We now bound one term at a time.
Let us start with (a):

$$\begin{aligned} \text{(a)}^2 = \lambda^2 \left\| \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_m^{-1}}^2 &= \lambda^2 \mathbf{h}^{\mathrm{T}} \widetilde{\mathbf{V}}_m^{-1} \mathbf{h} \\ &\leqslant \lambda^2 \left\| \widetilde{\mathbf{V}}_m^{-1} \right\|_2 \left\| \mathbf{h} \right\|_2^2 \end{aligned}$$

$$
\begin{aligned}
&\leqslant \lambda \left\|\mathbf{h}\right\|_2^2 \\
&\leqslant \lambda \left(\Theta + \frac{\Omega B \Phi(\mathbf{A})}{1-\rho(\mathbf{A})} \right)^2,
\end{aligned}
$$

where we observed that $\left\|\widetilde{\mathbf{V}}_m^{-1}\right\|_2 \leqslant \left\|\widetilde{\mathbf{V}}_m\right\|_2^{-1} \leqslant \lambda^{-1}$. Finally, we have bounded the norm of $\mathbf{h}$:

$$
\begin{aligned}
\left\|\mathbf{h}\right\|_2 &= \left\| \sum_{s=0}^{+\infty} \mathbf{h}^{\{s\}} \right\|_2 \\
&\leqslant \sum_{s=0}^{+\infty} \left\|\mathbf{h}^{\{s\}}\right\|_2 \\
&\leqslant \|\boldsymbol{\theta}\|_2 + \|\boldsymbol{\omega}\|_2 \|\mathbf{B}\|_2 \sum_{s=1}^{+\infty} \|\mathbf{A}\|^{s-1} \\
&\leqslant \Theta + \frac{\Omega B \Phi(\mathbf{A})}{1-\rho(\mathbf{A})},
\end{aligned}
$$

where we have exploited Assumptions 7.1 and 7.2.

We now move to term (b):

$$
\begin{aligned}
(\mathrm{b})^2 = \left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{g}}_m \right\|_{\widetilde{\mathbf{V}}_m^{-1}}^2 &= \widetilde{\mathbf{g}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\mathbf{g}}_m \\
&\leqslant \frac{1}{\lambda} \left\| \widetilde{\mathbf{g}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \right\|_2^2 \\
&= \frac{1}{\lambda} \left\| \sum_{l=1}^{m} \langle \widetilde{\mathbf{u}}_l, \mathbf{h}^{[\![H_l+2,\infty)\!)} \rangle \widetilde{\mathbf{u}}_l \right\|_2^2 \\
&\leqslant \frac{1}{\lambda} \left(\sum_{l=1}^{m} \|\widetilde{\mathbf{u}}_l\|_2^2 \left\| \mathbf{h}^{[\![H_l+2,\infty)\!)} \right\|_2 \right)^2 \\
&\leqslant \frac{U^4 \Omega^2 B^2 \Phi(\mathbf{A})^2}{\lambda (1-\rho(\mathbf{A}))^2} \cdot \left(\sum_{l=1}^{m} \rho(\mathbf{A})^{H_l+1} \right)^2,
\end{aligned}
$$

where we have employed the following inequality:

$$
\begin{aligned}
\left\| \mathbf{h}^{[\![H_l+2,\infty)\!)} \right\|_2 &= \left\| \boldsymbol{\omega}^{\mathrm{T}} \sum_{j=H_l+2}^{+\infty} \mathbf{A}^{j-1} \mathbf{B} \right\|_2 \\
&\leqslant \|\boldsymbol{\omega}\|_2 \|\mathbf{B}\|_2 \sum_{j=H_l+2}^{+\infty} \left\| \mathbf{A}^{j-1} \right\|_2
\end{aligned}
$$

$$\leqslant \Omega B \Phi(\mathbf{A}) \frac{\rho(\mathbf{A})^{H_l+1}}{1-\rho(\mathbf{A})}.$$

Let us now consider term (c):

$$\begin{aligned}
(\mathrm{c})^2 = \left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m \right\|^2_{\widetilde{\mathbf{V}}_m^{-1}} &= \widetilde{\boldsymbol{\nu}}_m^{\mathrm{T}} \widetilde{\mathbf{U}}_m \widetilde{\mathbf{V}}_m^{-1} \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m \\
&\leqslant \frac{1}{\lambda} \left\| \widetilde{\mathbf{U}}_m^{\mathrm{T}} \widetilde{\boldsymbol{\nu}}_m \right\|_2^2 \\
&= \frac{1}{\lambda} \left\| \sum_{s=1}^{m} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{H_l+1} \widetilde{\mathbf{x}}_{l-1} \widetilde{\mathbf{u}}_l \right\|_2^2 \\
&\leqslant \frac{1}{\lambda} \left(\sum_{s=1}^{m} \|\boldsymbol{\omega}\|_2 \left\| \mathbf{A}^{H_l+1} \right\|_2 \|\widetilde{\mathbf{x}}_{l-1}\|_2 \|\widetilde{\mathbf{u}}_l\|_2 \right)^2 \\
&\leqslant \frac{X^2 \Omega^2 U^2 \Phi(\mathbf{A})^2}{\lambda} \cdot \left(\sum_{l=1}^{m} \rho(\mathbf{A})^{H_l+1} \right)^2 .
\end{aligned}$$

We now bound the summations, exploiting the inequality $\rho(\mathbf{A}) \leqslant \overline{\rho}$, holding by assumption:

$$\begin{aligned}
\sum_{l=1}^{m} \rho(\mathbf{A})^{H_l+1} &= \sum_{l=1}^{m} \rho(\mathbf{A})^{\left\lfloor \frac{\log l}{\log \frac{1}{\overline{\rho}}} \right\rfloor + 1} \\
&\leqslant \sum_{l=1}^{m} \rho(\mathbf{A})^{\frac{\log l}{\log \frac{1}{\overline{\rho}}}} \\
&= \sum_{l=1}^{m} \exp\left(- \frac{\log \frac{1}{\rho(\mathbf{A})}}{\log \frac{1}{\overline{\rho}}} \log l \right) \\
&= \sum_{l=1}^{m} \frac{1}{l} \leqslant \log(m+1) + 1 \leqslant \log(t+1) + 1 = \log(e(t+1)),
\end{aligned}$$

having exploited the fact that $m \leqslant t$ and the bound with the integral to the harmonic sum.

Finally, we consider term (d). In this case, we apply Theorem 1 of (Abbasi-Yadkori et al., 2011), observing that the conditions are satisfied. To this end, we first need to determine the subgaussianity constant for the noise process $\widetilde{\xi}_l$. For every $l \in [\![m]\!]$ and $\zeta \in \mathbb{R}$, and properly using the tower law of expectation:

$$\mathbb{E}\left[\exp\left(\zeta \widetilde{\xi}_l \right) | \widetilde{\mathcal{F}}_{l-1} \right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\exp\left(\zeta\eta_{t_l} + \zeta\sum_{s=1}^{H_m+1}\boldsymbol{\omega}^{\mathrm{T}}\mathbf{A}^{s-1}\boldsymbol{\epsilon}_{t_l-s}\right)|\mathcal{F}_{t_l-1}\right]\\
&= \mathbb{E}\left[\exp\left(\zeta\eta_{t_l}\right)|\mathcal{F}_{t_l-1}\right]\prod_{s=1}^{H_m+1}\mathbb{E}\left[\mathbb{E}\left[\exp\left(\zeta\boldsymbol{\omega}^{\mathrm{T}}\mathbf{A}^{s-1}\boldsymbol{\epsilon}_{t_l-s}\right)|\mathcal{F}_{t_l-1-s}\right]|\mathcal{F}_{t_l-1}\right]\\
&\leqslant \exp\left(\frac{\zeta^2\sigma^2}{2}\right)\prod_{s=1}^{H_m+1}\mathbb{E}\left[\exp\left(\frac{\zeta^2\|\boldsymbol{\omega}^{\mathrm{T}}\mathbf{A}^{s-1}\|_2^2\sigma^2}{2}\right)|\mathcal{F}_{t_l-1}\right]\\
&\leqslant \exp\left(\frac{\zeta^2\sigma^2}{2}\right)\prod_{s=1}^{H_m+1}\exp\left(\frac{\zeta^2\Omega^2\Phi(\mathbf{A})^2\rho(\mathbf{A})^{2(s-1)}\sigma^2}{2}\right)\\
&\leqslant \exp\left(\frac{\sigma^2\zeta^2}{2}\left(1+\Omega^2\Phi(\mathbf{A})^2\sum_{s=1}^{+\infty}\rho(\mathbf{A})^{2(s-1)}\right)\right)\\
&= \exp\left(\frac{\sigma^2\zeta^2}{2}\left(1+\frac{\Omega^2\Phi(\mathbf{A})^2}{(1-\rho(\mathbf{A})^2)}\right)\right).
\end{aligned}
$$

Thus, simultaneously for all $m \in [\![M]\!]$, with probability at least $1-\delta$, it holds that:

$$
\begin{aligned}
(\mathrm{d})^2 &= \left\|\widetilde{\mathbf{U}}_m^{\mathrm{T}}\widetilde{\boldsymbol{\xi}}_m\right\|^2_{\widetilde{\mathbf{V}}_m^{-1}}\\
&\leqslant 2\sigma^2\left(1+\frac{\Omega^2\Phi(\mathbf{A})^2}{(1-\rho(\mathbf{A})^2)}\right)\left(\log\left(\frac{1}{\delta}\right)+\frac{1}{2}\log\left(\frac{\det\left(\widetilde{\mathbf{V}}_m\right)}{\lambda^d}\right)\right).
\end{aligned}
$$

□

We now proceed at bounding the offline regret R^{off} and, then, relating the offline regret R^{off} with the online regret R, as defined in Chapter 7.

Theorem C.0.2 (Offline Regret Upper Bound)**.** *Under Assumptions 7.1 and 7.2, having selected β_t as in Equation* (7.6)*, for every $\delta \in (0,1)$, with probability at least $1-\delta$,* `DynLin-UCB` *suffers an offline regret R^{off} bounded as:*

$$
R^{\mathit{off}}(\underline{\boldsymbol{\pi}}^{\texttt{DynLin-UCB}}, T) \leqslant \sqrt{8dT\beta_{T-1}^2\left(1+\frac{\log T}{\log\frac{1}{\overline{\rho}}}\right)\log\left(1+\frac{TU^2}{d\lambda}\right)}.
$$

Moreover, by setting $\delta = 1/T$, highlighting the dependencies on T, $\overline{\rho}$, d, *and σ only, the expected offline regret $\mathbb{E}\, R^{\text{off}}$ is bounded as:*

$$\mathbb{E}\, R^{\mathit{off}}(\underline{\boldsymbol{\pi}}^{\texttt{DynLin-UCB}}, T) \leqslant \mathcal{O}\left(\frac{d\sigma\sqrt{T}(\log T)^{\frac{3}{2}}}{1-\overline{\rho}} + \frac{\sqrt{dT}(\log T)^2}{(1-\overline{\rho})^{\frac{3}{2}}} \right).$$

Proof. For every epoch $m \in [\![M]\!]$, let us define $\widetilde{\beta}_{m-1} = \beta_{t_{m-1}}$ and define the confidence set $\mathcal{C}_{m-1} = \{\widetilde{\mathbf{h}} \in \mathbb{R}^d : \|\widetilde{\mathbf{h}} - \widetilde{\mathbf{h}}_{m-1}\|_{\widetilde{\mathbf{V}}_{m-1}} \leqslant \widetilde{\beta}_{m-1}\}$. Let us start by considering the instantaneous offline regret $\widetilde{r}_m$ at epoch $m \in [\![M]\!]$. Let $\mathbf{u}^* \in \arg\max_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{h}, \mathbf{u} \rangle$ and let $\widetilde{\mathbf{h}}^{\uparrow}_{m-1} \in \mathcal{C}_{m-1}$ such that $\mathrm{UCB}_{t_{m-1}+1}(\widetilde{\mathbf{u}}_m) = \langle \widetilde{\mathbf{h}}^{\uparrow}_{m-1}, \widetilde{\mathbf{u}}_m \rangle$. Thus, with probability at least $1-\delta$, we have:

$$\begin{aligned}
\widetilde{r}_m &= J^* - J(\widetilde{\mathbf{u}}_m) = \langle \mathbf{h}, \mathbf{u}^* \rangle - \langle \mathbf{h}, \widetilde{\mathbf{u}}_m \rangle \pm \langle \widetilde{\mathbf{h}}^{\uparrow}_{m-1}, \widetilde{\mathbf{u}}_m \rangle \\
&\leqslant \langle \widetilde{\mathbf{h}}^{\uparrow}_{m-1} - \mathbf{h}, \widetilde{\mathbf{u}}_m \rangle && \text{(C.7)} \\
&\leqslant \left\| \widetilde{\mathbf{h}}^{\uparrow}_{m-1} - \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_{m-1}} \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}^{-1}_{m-1}} \\
&\leqslant \left(\left\| \widetilde{\mathbf{h}}^{\uparrow}_{m-1} - \widetilde{\mathbf{h}}_{m-1} \right\|_{\widetilde{\mathbf{V}}_{m-1}} + \left\| \widetilde{\mathbf{h}}_{m-1} - \mathbf{h} \right\|_{\widetilde{\mathbf{V}}_{m-1}} \right) \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}^{-1}_{m-1}} && \text{(C.8)} \\
&\leqslant 2\widetilde{\beta}_{m-1} \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}^{-1}_{m-1}}. && \text{(C.9)}
\end{aligned}$$

where line (C.7) follows from the optimism, line (C.8) derives from triangle inequality, line (C.9) is obtained by observing that $\mathbf{h} \in \mathcal{C}_{m-1}$ with probability at least $1-\delta$, simultaneously for all $m \in [\![M]\!]$, thanks to Theorem 7.3.1, having observed that $\widetilde{\beta}_{m-1}$ is larger than the right hand side of Theorem 7.3.1.

We now move to the cumulative offline regret over the whole horizon T, by decomposing w.r.t. the epochs and recalling that we pay the same instantaneous regret within each epoch:

$$R^{\text{off}}(\texttt{DynLin-UCB}, T) = \sum_{m=1}^{M} (H_m + 1)\widetilde{r}_m \leqslant \sqrt{\sum_{m=1}^{M} (H_m+1)^2} \sqrt{\sum_{m=1}^{M} \widetilde{r}^2_m}.$$

Concerning the first summation, we proceed as follows, recalling that $M \leqslant T$ and $H_m \leqslant H_M$ for all $m \in [\![M]\!]$:

$$\sum_{m=1}^{M} (H_m+1)^2 \leqslant T(H_M + 1) \leqslant T\left(1 + \frac{\log T}{\log \frac{1}{\rho}}\right).$$

For the second summation, we follow the usual derivation for linear bandits, recalling that $\widetilde{\beta}_{M-1} \geqslant \max\{1, \widetilde{\beta}_{m-1}\}$ for all $m \in [\![M]\!]$ and that under Assumption 7.2 we have that $\widetilde{r}_m^2 \leqslant 2$. In particular:

$$\begin{aligned}\widetilde{r}_m^2 &\leqslant \min\left\{2, 2\widetilde{\beta}_{M-1} \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}_{m-1}^{-1}}\right\} \\ &\leqslant 2\widetilde{\beta}_{M-1} \min\left\{1, \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}_{m-1}^{-1}}\right\}.\end{aligned}$$

Plugging this inequality into the second summation, we obtain:

$$\begin{aligned}\sum_{m=1}^{M} \widetilde{r}_m^2 &\leqslant 4\widetilde{\beta}_{M-1}^2 \sum_{m=1}^{M} \min\left\{1, \|\widetilde{\mathbf{u}}_m\|_{\widetilde{\mathbf{V}}_{m-1}^{-1}}^2\right\} \\ &\leqslant 8d\widetilde{\beta}_{M-1}^2 \log\left(1 + \frac{MU^2}{d\lambda}\right) \\ &\leqslant 8d\beta_{T-1}^2 \log\left(1 + \frac{TU^2}{d\lambda}\right),\end{aligned}$$

where the last passage follows from the elliptic potential lemma (Lattimore and Szepesvári, 2020, Lemma 19.4). Putting all together, we obtain the inequality holding with probability at least $1 - \delta$:

$$R^{\text{off}}(\texttt{DynLin-UCB}, T) \leqslant \sqrt{8dT\beta_{T-1}^2 \left(1 + \frac{\log T}{\log \frac{1}{\overline{\rho}}}\right) \log\left(1 + \frac{TU^2}{d\lambda}\right)},$$

having observed that $\widetilde{\beta}_{M-1} \leqslant \beta_{T-1}$ We can also arrive at a problem-dependent regret bound, by setting $\Delta := \inf_{\mathbf{u} \in \mathcal{U} \langle \mathbf{h}, \mathbf{u} \rangle < \langle \mathbf{h}, \mathbf{u}^* \rangle} \langle \mathbf{h}, \mathbf{u}^* - \mathbf{u} \rangle$ (if it exists > 0). Since the instantaneous regret is either 0 or at least Δ, we have:

$$\begin{aligned}R^{\text{off}}(\texttt{DynLin-UCB}, T) &\leqslant \sum_{m=1}^{M} (H_m + 1) \frac{\widetilde{r}_m^2}{\Delta} \\ &\leqslant \frac{H_M + 1}{\Delta} 8d\widetilde{\beta}_{M-1}^2 \log\left(1 + \frac{MU^2}{d\lambda}\right) \\ &\leqslant \frac{8d}{\Delta}\left(1 + \frac{\log T}{\log \frac{1}{\overline{\rho}}}\right) \beta_{T-1}^2 \log\left(1 + \frac{TU^2}{d\lambda}\right).\end{aligned}$$

By setting $\delta = 1/T$, replacing the value of β_{T-1}, we obtain the offline regret in expectation, highlighting the dependence on T, $\overline{\rho}$, d, and σ only:

$$\mathbb{E}\, R^{\text{off}}(\texttt{DynLin-UCB}, T) \leqslant \mathcal{O}\left(\frac{d\sigma\sqrt{T}(\log T)^{\frac{3}{2}}}{1 - \overline{\rho}} + \frac{\sqrt{dT}(\log T)^2}{(1 - \overline{\rho})^{\frac{3}{2}}}\right),$$

where we used the fact that $\frac{1}{\log \frac{1}{\overline{\rho}}} \leqslant \frac{1}{1-\overline{\rho}}$ and $\rho(\mathbf{A}) \leqslant \overline{\rho}$. □

The following lemma relates the expected offline regret with the expected online regret.

Theorem 7.3.2 (Upper Bound)**.** *Under Assumptions 7.1 and 7.2, selecting β_t as in Equation* (7.6) *and $\delta = 1/T$,* `DynLin-UCB` *suffers an expected regret bounded as (highlighting the dependencies on T, $\overline{\rho}$, d, and σ only):*

$$\mathbb{E}[R(\underline{\boldsymbol{\pi}}^{\texttt{DynLin-UCB}},T)] \leqslant \mathcal{O}\left(\frac{d\sigma\sqrt{T}(\log T)^{\frac{3}{2}}}{1-\overline{\rho}} + \frac{\sqrt{d}T(\log T)^2}{(1-\overline{\rho})^{\frac{3}{2}}} + \frac{1}{(1-\rho(\mathbf{A}))^2}\right).$$

Proof. The result is simply obtained by exploiting the offline regret bound of Theorem C.0.2 and by upper bounding the expected regret thanks to Lemma C.0.1. □

C.1 Finite-Horizon Setting

In this section, we compare the finite-horizon setting with the infinite-horizon one presented in Chapter 7. We shall show that under Assumption 7.1, the two settings tend to coincide when the horizon is sufficiently large. Let us start by introducing the *H–horizon expected average reward*, with $H \in \mathbb{N}$ being the optimization horizon:

$$J_H(\underline{\boldsymbol{\pi}}) := \mathbb{E}\left[\frac{1}{H}\sum_{t=1}^{H} y_t\right] \quad \text{where}$$
$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t \\ y_t = \langle \boldsymbol{\omega}, \mathbf{x}_t \rangle + \langle \boldsymbol{\theta}, \mathbf{u}_t \rangle + \eta_t \\ \mathbf{u}_t = \boldsymbol{\pi}_t(H_{t-1}) \end{cases}, \quad t \in [H], \tag{C.10}$$

where the expectation is taken w.r.t. the randomness of the state noise $\boldsymbol{\epsilon}_t$ and reward noise η_t. We now show that the optimal policy for the finite-horizon setting is a non-stationary open-loop policy.

Theorem C.1.1 (Optimal Policy for the H–Horizon Setting)**.** *If $H \in \mathbb{N}$, an optimal policy $\underline{\boldsymbol{\pi}}^*_H = (\boldsymbol{\pi}^*_{H,t})_{t \in [\![H]\!]}$ maximizing the H-horizon expected average reward $J(\underline{\boldsymbol{\pi}})$ as in Equation* (C.10) *is given by:*

$$\forall t \in [\![H]\!], \quad \forall H_{t-1} \in \mathcal{H}_{t-1} : \qquad \boldsymbol{\pi}^*_{H,t}(H_{t-1}) = \mathbf{u}^*_{H,t}$$

where:

$$\mathbf{u}^*_{H,t} \in \arg\max_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{h}^{[\![0,H-t]\!]}, \mathbf{u} \rangle.$$

Proof. We start by expressing for every $t \in [\![H]\!]$ the reward y_t as a function of the sequence of actions $\underline{\mathbf{u}} = (\mathbf{u}_1, \dots, \mathbf{u}_H)$ produced by a generic policy $\underline{\boldsymbol{\pi}}$. By exploiting Equation (7.4) instanced with $H = t-1$, we have:

$$y_t = \sum_{s=0}^{t-1} \langle \mathbf{h}^{\{s\}}, \mathbf{u}_{t-s} \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1} \mathbf{x}_1 + \eta_t + \sum_{s=1}^{t-1} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{s-1} \boldsymbol{\epsilon}_{t-s}.$$

By computing the expectation, using linearity, and recalling that the noises are zero-mean, we obtain:

$$\mathbb{E}[y_t] = \sum_{s=0}^{t-1} \langle \mathbf{h}^{\{s\}}, \mathbb{E}[\mathbf{u}_{t-s}] \rangle + \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1}\, \mathbb{E}[\mathbf{x}_1].$$

By averaging over $t \in [\![H]\!]$, we get the H-horizon expected average reward:

$$\begin{aligned}
J_H(\underline{\boldsymbol{\pi}}) &= \frac{1}{H} \sum_{t=1}^{H} \mathbb{E}[y_t] \\
&= \frac{1}{H} \sum_{t=1}^{H} \sum_{s=0}^{t-1} \langle \mathbf{h}^{\{s\}}, \mathbb{E}[\mathbf{u}_{t-s}] \rangle + \frac{1}{H} \sum_{t=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1}\, \mathbb{E}[\mathbf{x}_1] \\
&= \frac{1}{H} \sum_{s=1}^{H} \left(\sum_{t=s}^{H} \mathbf{h}^{\{t-s\}} \right)^{\mathrm{T}} \mathbb{E}[\mathbf{u}_s] + \frac{1}{H} \sum_{t=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1}\, \mathbb{E}[\mathbf{x}_1] \qquad \text{(C.11)} \\
&= \frac{1}{H} \sum_{s=1}^{H} \langle \mathbf{h}^{[\![0,H-s]\!]}, \mathbb{E}[\mathbf{u}_s] \rangle + \frac{1}{H} \sum_{t=1}^{H} \boldsymbol{\omega}^{\mathrm{T}} \mathbf{A}^{t-1}\, \mathbb{E}[\mathbf{x}_1]. \qquad \text{(C.12)}
\end{aligned}$$

where line (C.11) is obtained by renaming the indexes of the summations, and line (C.12) comes from the definition of cumulative Markov parameter $\mathbf{h}^{[\![0,H-s]\!]}$. It is now simple to see, as no noise is present in the expression, that the performance $J_H(\underline{\boldsymbol{\pi}})$ is maximized by taking at each round $s \in \mathbb{N}$ an action $\mathbf{u}^*_s = \boldsymbol{\pi}^*_s(H_{s-1})$ such that whose expectation satisfies $\mathbb{E}[\mathbf{u}^*_s] = \arg\max_{\mathbb{E}[\mathbf{u}_s]} \langle \mathbf{h}^{[\![0,H-s]\!]}, \mathbb{E}[\mathbf{u}_s] \rangle$. Clearly, we can take the deterministic action such that $\mathbf{u}^*_s = \mathbb{E}[\mathbf{u}^*_s]$. □

We now show that for sufficiently large H, the H-horizon expected average reward J_H tends to coincide with the infinite-horizon expected average reward.

Proposition C.1.2. *Let $H \in \mathbb{N}$. Then, for every policy $\underline{\boldsymbol{\pi}}$ it holds that:*

$$|J_H(\underline{\boldsymbol{\pi}}) - J(\underline{\boldsymbol{\pi}})| \leqslant \frac{BU\Omega\Phi(\mathbf{A})(1-\rho(\mathbf{A})^H)}{H(1-\rho(\mathbf{A}))}.$$

Proof. Consider two horizons $H < H' \in \mathbb{N}$, and let $(\mathbf{u}_1, \mathbf{u}_2, \dots)$ be the sequence of actions played by policy $\underline{\boldsymbol{\pi}}$. Using Equation (C.12), we have:

$$J_H(\underline{\boldsymbol{\pi}}) - J_{H'}(\underline{\boldsymbol{\pi}}) \tag{C.13}$$

$$= \frac{1}{H}\sum_{s=1}^{H} \langle \mathbf{h}^{\llbracket 0, H-s \rrbracket}, \mathbb{E}[\mathbf{u}_s] \rangle - \frac{1}{H'}\sum_{s=1}^{H'} \langle \mathbf{h}^{\llbracket 0, H'-s \rrbracket}, \mathbb{E}[\mathbf{u}_s] \rangle \tag{C.14}$$

$$= \frac{1}{H}\sum_{s=1}^{H} \langle \mathbf{h}^{\llbracket 0, H-s \rrbracket} - \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle - \frac{1}{H'}\sum_{s=1}^{H'} \langle \mathbf{h}^{\llbracket 0, H'-s \rrbracket} - \mathbf{h}, \mathbb{E}[\mathbf{u}_s] \rangle \tag{C.15}$$

$$= -\frac{1}{H}\sum_{s=1}^{H} \langle \mathbf{h}^{\llbracket H-s+1, +\infty \rrparenthesis}, \mathbb{E}[\mathbf{u}_s] \rangle + \frac{1}{H'}\sum_{s=1}^{H'} \langle \mathbf{h}^{\llbracket H'-s+1, +\infty \rrparenthesis}, \mathbb{E}[\mathbf{u}_s] \rangle. \tag{C.16}$$

As shown in Appendix C.0.1, we have that the second addendum vanishes as H' approaches $+\infty$:

$$\frac{1}{H'}\left|\sum_{s=1}^{H'} \langle \mathbf{h}^{\llbracket H'-s+1, +\infty \rrparenthesis}, \mathbb{E}[\mathbf{u}_s] \rangle\right| \to 0 \qquad \text{when} \qquad H' \to +\infty.$$

Concerning the first addendum, we have:

$$\begin{aligned}\frac{1}{H}\left|\sum_{s=1}^{H} \langle \mathbf{h}^{\llbracket H-s+1, +\infty \rrparenthesis}, \mathbb{E}[\mathbf{u}_s] \rangle\right| &\leqslant \frac{U}{H}\sum_{s=1}^{H} \left\| \mathbf{h}^{\llbracket H-s+1, +\infty \rrparenthesis} \right\|_2 \\ &\leqslant \frac{BU\Omega\Phi(\mathbf{A})}{H}\sum_{s=1}^{H} \rho(\mathbf{A})^{H-s} \\ &= \frac{BU\Omega\Phi(\mathbf{A})(1-\rho(\mathbf{A})^H)}{H(1-\rho(\mathbf{A}))}.\end{aligned}$$

□

APPENDIX $\mathcal{D}$

Omitted Proofs of Chapter 8

In this appendix, we provide the proofs we have omitted in Chapter 8.

D.1 Proofs and Derivations

We first provide proofs of the statements discussed in Chapter 8 (Section D.1.1), then we provide some technical lemmas needed in order to prove them (Section D.1.2).

D.1.1 Proofs of the Theorems

Theorem 8.3.1 (Worst-Case Lower Bound)**.** *For every algorithm $\mathfrak{A}$, there exists an FRB $\underline{\boldsymbol{\nu}}$ such that for:*

$$T \geqslant 2\big(1 - 2^{-\frac{1}{d-1}}\big)^{-2}\sigma^2 \max_{i \in [\![d]\!]} k_i, \tag{8.2}$$

$\mathfrak{A}$ suffers an expected cumulative regret of at least:

$$\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] \geqslant \frac{\sigma}{4\sqrt{2}} \sum_{i \in [\![d]\!]} \sqrt{k_i T}.$$

In particular, if $k_i =: k$ for every $i \in [\![d]\!]$, we have:

$$\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] \geqslant \Omega(\sigma d\sqrt{kT}).$$

Proof. Consider an scenario in which $\mu_{\mathbf{a}^*} = 1$ and $\Delta_{i,j} \leqslant \overline{\Delta} = 1 - 2^{-1/(d-1)}, \forall i \in [\![d]\!], j \in [\![k_i]\!]$, then Lemma D.1.3 allow us to rewrite the expected regret as:

$$\begin{aligned}
\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] &= \mathbb{E}\left[\sum_{t \in [\![T]\!]} \left(1 - \prod_{i \in [\![d]\!]} \left(1 - \Delta_{i,a_i(t)}\right)\right)\right] \\
&\geqslant \frac{1}{2}\mathbb{E}\left[\sum_{t \in [\![T]\!]} \sum_{i \in [\![d]\!]} \Delta_{i,a_i(t)}\right] \\
&= \frac{1}{2}\sum_{i \in [\![d]\!]} \mathbb{E}\left[\sum_{t \in [\![T]\!]} \Delta_{i,a_i(t)}\right] \\
&= \frac{1}{2}\sum_{i \in [\![d]\!]} \mathbb{E}\left[R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right], \qquad \text{(D.1)}
\end{aligned}$$

where $R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}})$ is the expected regret generated by pulling suboptimal arms on the component $i \in [\![d]\!]$. This fact implies that if we take sufficiently small $\Delta_{i,j} < \overline{\Delta}, \forall i \in [\![d]\!], j \in [\![k_i]\!]$, we can analyze the expected regret $R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}})$ we pay for each action component $i \in [\![d]\!]$ independently and then summing up the regret we pay as shown above. We will see how the condition of the sufficiently small $\Delta_{i,j}$ implies that we have to add a condition on the minimum time budget T for which this lower bound holds. We can define a set of $\prod_{i \in [\![d]\!]} k_i$ FRB *base instances* as follows. Given a vector $(h_1, \dots, h_d)^\mathsf{T} \in [\![k_1]\!] \times \cdots \times [\![k_d]\!]$ identifying an instance, we define the expected rewards of such an instance as follows, for $\Delta \in (0, 1/2)$:

$$\mu_{i,j} = \begin{cases} 1 & \text{if } j = h_i \\ 1 - \Delta & \text{if } j \in [\![k_i]\!] \backslash \{h_i\} \end{cases}, \quad \forall i \in [\![d]\!]. \qquad \text{(D.2)}$$

We refer as $\underline{\boldsymbol{\nu}}_{(h_1, \dots, h_d)}$ to the instance in which expected values are characterized by the vector $(h_1, \dots, h_d)^\mathsf{T} \in [\![k_1]\!] \times \cdots \times [\![k_d]\!]$ as in Equation (D.2). We now focus on bounding the regret of a single component $i \in [\![d]\!]$. In particular, we focus on component $i = 1$ for the sake of simplicity in the presentation. Then, we can extend the same reasoning to all the others. Let us define a set of *helper instances* which are needed for the analysis. For all the components different from the first, we consider as before a vector $(h_2, \dots, h_d)^\intercal \in [\![k_2]\!] \times \dots \times [\![k_d]\!]$ which characterize the instance $\underline{\boldsymbol{\nu}}_{(0,h_2,\dots,h_d)}$ defined as follows:

$$\mu_{1,j} = 1 - \Delta, \quad \forall j \in [\![k_1]\!]$$
$$\mu_{i,j} = \begin{cases} 1 & \text{if } j = h_i \\ 1 - \Delta & \text{if } j \in [\![k_i]\!] \backslash \{h_i\} \end{cases}, \quad \forall i \in [\![2, d]\!].$$

Before continue with the proof, we need to introduce some additional quantities. Given a vector $(h_1, h_2, \dots, h_d)^\intercal \in (\{0\} \cup [\![k_1]\!]) \times [\![k_2]\!] \times \dots \times [\![k_d]\!]$, we call $\mathbb{P}_{(h_1,h_2,\dots,h_d)}$ the distribution induced by the history of the pulls and the related rewards for the d components over time horizon T in instance $\underline{\boldsymbol{\nu}}_{(h_1,h_2,\dots,h_d)}$. We denote with $\mathbb{P}_h$ for $h \in \{0\} \cup [\![k_1]\!]$ the distribution induced by the history averaged over the other dimensions, formally:

$$\mathbb{P}_h = \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{(h_2,h_3,\dots,h_d) \in [\![k_2]\!] \times \dots \times [\![k_d]\!]} \mathbb{P}_{(h,h_2,\dots,h_d)},$$

and with $\mathbb{E}_h$ the expectation over $\mathbb{P}_h$.

Coming back to the proof, given the definition of the base instances (Equation D.2), the expected regret $\mathbb{E}\left[R_T^{(1)}(\mathfrak{A}, \underline{\boldsymbol{\nu}}_{(h_1,\dots,h_d)})\right]$ related to the first component is given by:

$$\begin{aligned} \mathbb{E}\left[R_T^{(1)}(\mathfrak{A}, \underline{\boldsymbol{\nu}}_{(h_1,\dots,h_d)})\right] &= \Delta \sum_{j \in [\![k_1]\!] \backslash \{h_1\}} \mathbb{E}\left[N_{1,j}(T)\right] \\ &= \Delta \left(T - \mathbb{E}\left[N_{1,h_1}(T)\right]\right). \end{aligned}$$

We now want to use Lemma D.1.4 in order to obtain the following condition:

$$\frac{1}{k_1} \sum_{h \in [\![k_1]\!]} \mathbb{E}_h[T - N_{1,h}(T)] \geqslant \frac{T}{4}. \tag{D.3}$$

To apply Lemma D.1.4, we need an upper bound on the total variation d_{TV} that we can compute $\forall h \in [\![k_1]\!]$ as follows:

$$\begin{aligned} d_{\text{TV}} &= \frac{1}{2} \left\| \mathbb{P}_0 - \mathbb{P}_h \right\|_1 \\ &= \frac{1}{2} \left\| \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \left(\mathbb{P}_{(0,h_2,\dots,h_d)} - \mathbb{P}_{(h,h_2,\dots,h_d)}\right) \right\|_1 \end{aligned}$$

$$\leqslant \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \frac{1}{2} \left\| \mathbb{P}_{(0,h_2,\dots,h_d)} - \mathbb{P}_{(h,h_2,\dots,h_d)} \right\|_1 \tag{D.4}$$

$$\leqslant \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \sqrt{\frac{1}{2} D_{\mathrm{KL}} \left(\mathbb{P}_{(0,h_2,\dots,h_d)} \middle|\middle| \mathbb{P}_{(h,h_2,\dots,h_d)} \right)} \tag{D.5}$$

$$= \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \sqrt{\frac{1}{2} \mathbb{E}_{(0,h_2,\dots,h_d)}[N_{1,h}(T)] D_{\mathrm{KL}} \left(p_0 \middle\| p_h \right)} \tag{D.6}$$

$$= \frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \sqrt{\frac{1}{2} \mathbb{E}_{(0,h_2,\dots,h_d)}[N_{1,h}(T)] \frac{\Delta^2}{2\sigma^2}} \tag{D.7}$$

$$\leqslant \sqrt{\frac{1}{\prod_{i \in [\![2,d]\!]} k_i} \sum_{\substack{(h_2,h_3,\dots,h_d) \\ \in [\![k_2]\!] \times \dots \times [\![k_d]\!]}} \frac{1}{2} \mathbb{E}_{(0,h_2,\dots,h_d)}[N_{1,h}(T)] \frac{\Delta^2}{2\sigma^2}} \tag{D.8}$$

$$\leqslant \frac{1}{4} \sqrt{\frac{\Delta^2}{2\sigma^2} \mathbb{E}_0[N_{1,h}(T)]}, \tag{D.9}$$

where line (D.4) is the triangle inequality for norms, line (D.5) is due the Pinsker's inequality, line (D.6) is due to the divergence decomposition lemma (Lattimore and Szepesvári, 2020, Lemma 15.1) considering that all the component different from the first are equal, line (D.7) is derived by the expression of D_{KL} between Gaussian distributions, line (D.8) is due to the Jensen's inequality, and line (D.9) is obtained by marginalizing w.r.t. the first component.

Given this upper bound to the total variation, we can finally apply Lemma D.1.4 considering $m = k_1$ and $B = \frac{2\sigma^2 k_1}{\Delta^2}$. What we get is:

$$\frac{1}{k_1} \sum_{i \in [\![k_1]\!]} \mathbb{E}_h \left[\frac{2\sigma^2 k_1}{\Delta^2} - N_{1,h}(T) \right] \geqslant \frac{\sigma^2 k_1}{2\Delta^2}. \tag{D.10}$$

We can now select the value of Δ in order to have in Equation (D.10) a bound on T:

$$T = \frac{2\sigma^2 k_1}{\Delta^2}.$$

This implies a choice of Δ in the form of:

$$\Delta = \sqrt{\frac{2\sigma^2 k_1}{T}}.$$

Given such a choice of Δ and the bound given by Equation (D.3), we get that the regret of the first action component can be bounded as:

$$\begin{aligned}\mathbb{E}\left[R_T^{(1)}(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] &\geqslant \Delta\left(T - \mathbb{E}\left[N_{1,h_1}(T)\right]\right)\\ &\geqslant \sqrt{\frac{2\sigma^2 k_1}{T}}\frac{T}{4}\\ &= \sqrt{\frac{\sigma^2 k_1 T}{8}}\\ &= \frac{1}{2\sqrt{2}}\sigma\sqrt{k_1 T}.\end{aligned}$$

The same reasoning can be done for all the others $d-1$ action components and the bound of Equation (D.1):

$$\begin{aligned}\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] &\geqslant \frac{1}{2}\sum_{i \in [\![d]\!]} \mathbb{E}\left[R_T^{(i)}(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]\\ &\geqslant \frac{1}{4\sqrt{2}}\sigma \sum_{i \in [\![d]\!]} \sqrt{k_i T}.\end{aligned}$$

The last point needed is to check that the condition of the choices we made on the Δ is compliant for all the dimensions $i \in [\![d]\!]$ with the one of Lemma D.1.3, i.e., all the Δs are less than $\overline{\Delta}$ defined as:

$$\overline{\Delta} = \sqrt{\frac{2\sigma^2 \max_{i \in [\![d]\!]} k_i}{T}}.$$

This implies a lower bound on the T for which this bound holds:

$$\sqrt{\frac{2\sigma^2 \max_{i \in [\![d]\!]} k_i}{T}} \leqslant 1 - 2^{-1/(d-1)}.$$

Isolating T we get:

$$T \geqslant \frac{2\sigma^2 \max_{i \in [\![d]\!]} k_i}{\left(1 - 2^{-1/(d-1)}\right)^2}.$$

This concludes the proof. □

Theorem 8.3.2 (Worst-Case Lower Bound without Intermediate Observations)**.** *For every algorithm $\mathfrak{A}^\dagger$ that ignores the intermediate observations* $\mathbf{x}(t)$ *and observes the reward* $r(t)$ *only, there exists an FRB* $\underline{\boldsymbol{\nu}}$ *such that for:*

$$T \geqslant 4(\min_{i \in [\![d]\!]} k_i - 1)/d,$$

$\mathfrak{A}^\dagger$ *suffers an expected cumulative regret of at least:*

$$\mathbb{E}\left[R_T(\mathfrak{A}^\dagger, \underline{\boldsymbol{\nu}})\right] \geqslant \frac{\sigma^d}{8}\sqrt{\frac{(\min_{i \in [\![d]\!]} k_i - 1)T}{d}}.$$

In particular, if $k_i =: k$ *for every* $i \in [\![d]\!]$*, we have:*

$$\mathbb{E}\left[R_T(\mathfrak{A}^\dagger, \underline{\boldsymbol{\nu}})\right] \geqslant \Omega(\sigma^d\sqrt{kT/d}).$$

Proof. For simplicity, we consider d even. We consider the following base instance $\underline{\boldsymbol{\nu}}$, parametrized by $\sigma > 1$ and $\Delta \in [0, 1/4]$ with $\Delta \leqslant \sigma^d$, defined for all $i \in [\![d]\!]$ and $j \in [\![k_i]\!]\backslash\{1\}$:

$$\nu_{i,1} = \begin{cases} \sigma & \text{w.p. } \frac{1}{2} + \frac{\Delta^{1/d}}{2\sigma} \\ -\sigma & \text{w.p. } \frac{1}{2} - \frac{\Delta^{1/d}}{2\sigma} \end{cases}, \qquad \nu_{i,j} = \begin{cases} \sigma & \text{w.p. } \frac{1}{2} \\ -\sigma & \text{w.p. } \frac{1}{2} \end{cases}.$$

It is clear that $\mu_{i,1} = \Delta^{1/d}$ and $\mu_{i,j} = 0$. Consequently, the optimal arm is $(1, \ldots, 1)^\top$ with performance $\mu^* = \Delta$ and all the other arms have performance 0. Furthermore, the variance of the suboptimal arm components is given by σ^2 which is also the subgaussian proxy, while for the optimal arm components, the variance is smaller. Consider now for every $i \in [\![d]\!]$:

$$j_i^* \in \operatorname*{arg\,min}_{j \in [\![k_i]\!]\backslash\{1\}} \mathbb{E}_{\underline{\boldsymbol{\nu}}}[N_{i,j}(T)] \implies \mathbb{E}_{\underline{\boldsymbol{\nu}}}[N_{i,j_i^*}(T)] \leqslant \frac{T}{k_i - 1}.$$

We construct the alternative instance $\underline{\boldsymbol{\nu}}$ which is equal to $\underline{\boldsymbol{\nu}}'$ except for the the components (i, j_i^*) for $i \in [\![d]\!]$:

$$\nu_{i,j_i^*} = \begin{cases} \sigma & \text{w.p. } \frac{1}{2} + \frac{(2\Delta)^{1/d}}{2\sigma} \\ -\sigma & \text{w.p. } \frac{1}{2} - \frac{(2\Delta)^{1/d}}{2\sigma} \end{cases},$$

enforcing $\Delta \leqslant \sigma^d/2$. In this alternative instance, the optimal arm corresponds to $(j_1^*, \ldots j_d^*)^\top$, with performance $(\mu^*)' = 2\Delta$.
We are considering algorithms that do not observe individual components. Therefore, the distribution of the product of the individual components has

to be computed. Since they will be used in the computation of the KL-divergence, we just consider the two most dissimilar ones:

$$\nu_{\dagger}^{\otimes} = \begin{cases} \sigma^d & \text{w.p. } \frac{1}{2} + \frac{\Delta}{\sigma^d} \\ -\sigma^d & \text{w.p. } \frac{1}{2} - \frac{\Delta}{\sigma^d} \end{cases}, \qquad \nu_{\ddagger}^{\otimes} = \begin{cases} \sigma^d & \text{w.p. } \frac{1}{2} \\ -\sigma^d & \text{w.p. } \frac{1}{2} \end{cases},$$

where the probability of the first case in which we play, for instance, the arm $(1, \dots, 1)^\top$ in the base instance is obtained by the following reasoning: we get σ^d if the number of σ realizations is even (being d even). Thus, we have:

$$\begin{aligned} \mathbb{P}(\{\sigma^d\}) &= \sum_{l=0}^{d} \mathbb{1}\{l \text{ is even}\} \binom{d}{j} \left(\frac{1}{2} + \frac{(2\Delta)^{1/d}}{2\sigma^d}\right)^j \left(\frac{1}{2} - \frac{(2\Delta)^{1/d}}{2\sigma^d}\right)^{d-j} \\ &= \frac{1}{2} + \frac{\Delta}{\sigma^d}. \end{aligned}$$

The KL divergence becomes, using reverse Pinsker's inequality:

$$\begin{aligned} D_{\mathrm{KL}}(\nu_{\dagger}^{\otimes}, \nu_{\ddagger}^{\otimes}) &\leqslant \frac{1}{\frac{1}{2} - \frac{\Delta}{\sigma^d}} D_{\mathrm{TV}}(\nu_{\dagger}^{\otimes}, \nu_{\ddagger}^{\otimes}) \\ &= 4\left(\frac{\Delta}{\sigma^d}\right)^2 \\ &= \frac{4\Delta^2}{\sigma^{2d}}. \end{aligned}$$

requiring $\Delta \leqslant \sigma^d/4$.
Let us now lower bound the regret with Bretagnolle-Huber's inequality:

$$\begin{aligned} &\max\{\mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})], \mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}}')]\} \\ &\quad \geqslant \frac{\Delta T}{4} \exp\left(-\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}\left[\sum_{t=1}^{T} \mathbb{1}\{\exists i \in [\![d]\!] \,:\, a_i(t) = j_i^*\} D_{\mathrm{KL}}(\nu_{\mathbf{a}(t)}^{\otimes} \| (\nu')_{\mathbf{a}(t)}^{\otimes})\right]\right) \\ &\quad \geqslant \frac{\Delta T}{4} \exp\left(-\sum_{i \in [\![d]\!]} \mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}[N_{i,j_i^*}(T)] \frac{4\Delta^2}{\sigma^{2d}}\right) \\ &\quad \geqslant \frac{\Delta T}{4} \exp\left(-\frac{4dT\Delta^2}{\sigma^{2d}(k^* - 1)}\right), \end{aligned}$$

being $k^* = \min_{i \in [\![d]\!]} k_i$. We set $\Delta = \sqrt{\frac{\sigma^{2d}(k^*-1)}{4dT}}$ with $T \geqslant 4(k^* - 1)/d$. □

Theorem 8.3.3 (Instance-Dependent Lower Bound). *For every consistent*[1] *algorithm $\mathfrak{A}$ and FRB $\underline{\boldsymbol{\nu}}$ with unique optimal arm $\mathbf{a}^* \in \mathcal{A}$ it holds that:*

$$\liminf_{T\to+\infty} \frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T} \geqslant \underline{C}(\underline{\boldsymbol{\nu}}), \tag{8.3}$$

where $\underline{C}(\underline{\boldsymbol{\nu}})$ is defined as the solution to the following optimization problem:

$$\min_{(L_{\mathbf{a}})_{\mathbf{a}\in\mathcal{A}\backslash\{\mathbf{a}^*\}}} \sum_{\mathbf{a}\in\mathcal{A}\backslash\{\mathbf{a}^*\}} L_{\mathbf{a}}\Delta_{\mathbf{a}} \tag{8.4}$$

$$\text{s.t.} \quad L_{i,j} = \sum_{\substack{\mathbf{a}\in\mathcal{A}\backslash\{\mathbf{a}^*\} \\ a_i = j}} L_{\mathbf{a}}, \ \forall i \in \llbracket d \rrbracket, \ j \in \llbracket k_i \rrbracket \backslash \{a_i^*\} \tag{8.5}$$

$$L_{i,j} \geqslant \frac{2\sigma^2}{\Delta_{i,j}^2}, \ \forall i \in \llbracket d \rrbracket, j \in \llbracket k_i \rrbracket \backslash \{a_i^*\} \tag{8.6}$$

$$L_{\mathbf{a}} \geqslant 0, \quad \forall \mathbf{a} \in \mathcal{A}\backslash\{\mathbf{a}^*\}. \tag{8.7}$$

Proof. The proof of this statement is divided into two parts. Part one is dedicated to finding a lower bound on the expected number of pulls of every action component $N_{i,j}(T)$ for each action component $i \in \llbracket d \rrbracket, \ j \in \llbracket k_i \rrbracket \backslash \{a_i^*\}$. Part two is dedicated to understanding how these pulls of the action components can be combined in action vectors in the best way possible.

Part 1: Lower bounding the expected number of pulls for each action component

The proof of the expected number of pulls of a sub-optimal action $j \in \llbracket k_i \rrbracket \backslash \{a_i^*\}$ of action component $i \in \llbracket d \rrbracket$ is inspired by the proof of the asymptotic number of pulls of sub-optimal arms presented in Theorem 16.2 of (Lattimore and Szepesvári, 2020).

We call $\mathcal{M}_{mn}$ the set of distributions referring to the m^{th} component ($m \in \llbracket d \rrbracket$) and the n^{th} arm ($n \in \llbracket k_m \rrbracket$). Then, consider P_{mn} as a specific distribution taken from $\mathcal{M}_{mn}$ to model the reward observations of arm n of component m in a given instance of the setting.

Let $\underline{\boldsymbol{\nu}}$ be an instance of the FRB setting with d components and k_i actions for every $i \in \llbracket d \rrbracket$. We start by selecting a component $\underline{i}$ and a sub-optimal arm $\underline{j}$. Let $\varepsilon > 0 \in \mathbb{R}$ be arbitrary constant. We define a new instance of the FRB setting $\underline{\boldsymbol{\nu}}'$ such that $P'_{ij} = P_{ij}, \forall i \in \llbracket d \rrbracket \backslash \{\underline{i}\}, \forall j \in \llbracket k_i \rrbracket$, and $P'_{\underline{i}j} = P_{\underline{i}j}, \forall j \in \llbracket k_{\underline{i}} \rrbracket \backslash \{\underline{j}\}$, and $P'_{\underline{i},\underline{j}} \in \mathcal{M}_{\underline{i},\underline{j}}$ be such that $D_{KL}(P_{\underline{i},\underline{j}}, P'_{\underline{i},\underline{j}}) \leqslant d_{\underline{i},\underline{j}} + \varepsilon$ and $\mu'_{\underline{i},\underline{j}} > \mu^*_{\underline{i}}$. d_{mn} represents the KL divergence between P_{mn} and P^*_m.

[1] An algorithm $\mathfrak{A}$ is *consistent* if for every FRB $\underline{\boldsymbol{\nu}}$ and $p > 0$, it holds that $\limsup_{T\to+\infty} \mathbb{E}[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})]/T^p = 0$.

The newly defined instance $\underline{\boldsymbol{\nu}}'$ is then identical to $\underline{\boldsymbol{\nu}}$ for every arm of every component different from $\underline{i}$, and in the $\underline{i}^{\text{th}}$ component every arm is identical except for arm $\underline{j}$, which is sub-optimal in $\underline{\boldsymbol{\nu}}$ and is optimal in $\underline{\boldsymbol{\nu}}'$. Following the original proof, we can define, for any event $\mathcal{E}$:

$$\mathbb{P}_{\underline{\boldsymbol{\nu}}}(\mathcal{E}_{\underline{i},\underline{j}}) + \mathbb{P}_{\underline{\boldsymbol{\nu}}'}(\mathcal{E}^{\complement}_{\underline{i},\underline{j}}) \geqslant \frac{1}{2}\exp\left(-\mathbb{E}_{\underline{\boldsymbol{\nu}}}\left[N_{\underline{i},\underline{j}}(T)\right]\left(d_{\underline{i},\underline{j}} + \varepsilon\right)\right).$$

Now, let $\mathcal{E}_{\underline{i},\underline{j}} = \{N_{\underline{i},\underline{j}}(T) > T/2\}$, and let $R_T = R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})$ and $R'_T = R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}}')$. Then:

$$R_T + R'_T \geqslant \frac{T}{2}\left(\mathbb{P}_{\underline{\boldsymbol{\nu}}}(\mathcal{E}_{\underline{i},\underline{j}}) f_{\underline{i}}(\boldsymbol{\mu})\Delta_{\underline{i},\underline{j}} + \mathbb{P}_{\underline{\boldsymbol{\nu}}'}(\mathcal{E}^{\complement}_{\underline{i},\underline{j}}) f_{\underline{i}}(\boldsymbol{\mu})(\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})\right),$$

where $f_{\underline{i}}(\boldsymbol{\mu})$ is obtained by the following observation. Since at every round $t \in [\![T]\!]$, in which we pull $(\underline{i}, \underline{j})$ we suffer the instantaneous regret in the base instance:

$$\prod_{i \in [\![d]\!]} \mu^*_i - \mu_{\underline{i},\underline{j}} \prod_{i \in [\![d]\!] \setminus \{\underline{i}\}} \mu_{i,j(t)} \geqslant (\mu^*_i - \mu_{\underline{i},\underline{j}}) \prod_{i \in [\![d]\!] \setminus \{\underline{i}\}} \mu^*_i = \Delta_{\underline{i},\underline{j}} \prod_{i \in [\![d]\!] \setminus \{\underline{i}\}} \mu^*_i$$

and in the alternative instance:

$$\mu'_{\underline{i},\underline{j}} \prod_{i \in [\![d]\!] \setminus \{\underline{i}\}} \mu^*_i - \prod_{i \in [\![d]\!]} \mu_{i,j(t)} \geqslant (\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}}) \prod_{i \in [\![d]\!] \setminus \{\underline{i}\}} \mu^*_i,$$

we define:

$$f_{\underline{i}}(\boldsymbol{\mu}) \coloneqq \prod_{i \in [\![d]\!] \neq \{\underline{i}\}} \mu^*_i.$$

Since the term $f_{\underline{i}}(\boldsymbol{\mu})$ multiplies both $\Delta_{\underline{i},\underline{j}}$ and $(\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})$, it is straightforward to continue the original proof and write:

$$R_T + R'_T \geqslant \frac{T}{4} f_{\underline{i}}(\boldsymbol{\mu}) \min\{\Delta_{\underline{i},\underline{j}}, (\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})\} \exp\left(-\mathbb{E}_{\underline{\boldsymbol{\nu}}}\left[N_{\underline{i},\underline{j}}(T)\right]\left(d_{\underline{i},\underline{j}} + \varepsilon\right)\right).$$

Rearranging and dividing by $\log T$, we obtain:

$$\frac{\mathbb{E}_{\underline{\boldsymbol{\nu}}}[N_{\underline{i},\underline{j}}(T)]}{\log(T)} \geqslant \frac{\log(T) + \log\left(\frac{f_{\underline{i}}(\boldsymbol{\mu})}{4} \min\{\Delta_{\underline{i},\underline{j}}, (\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})\}\right) - \log(R_T + R'_T)}{(d_{\underline{i},\underline{j}} + \varepsilon)\log(T)}$$

$$= \frac{1}{d_{\underline{i},\underline{j}} + \varepsilon} + \frac{\log\left(\frac{f_{\underline{i}}(\boldsymbol{\mu})}{4}\min\{\Delta_{\underline{i},\underline{j}}, (\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})\}\right) - \log(R_T + R'_T)}{(d_{\underline{i},\underline{j}} + \varepsilon)\log(T)}$$
$$\geqslant \frac{2\sigma^2}{\Delta^2_{\underline{i},\underline{j}}} - h_{\underline{i},\underline{j}}(T),$$

by letting $\varepsilon \to 0$, having exploited the expression of KL-divergence between Gaussians and having set:

$$h_{\underline{i},\underline{j}}(T) := \max\left\{0, \frac{\log\left(\frac{f_{\underline{i}}(\boldsymbol{\mu})}{4}\min\{\Delta_{\underline{i},\underline{j}}, (\mu'_{\underline{i},\underline{j}} - \mu^*_{\underline{i}})\}\right) - \log(R_T + R'_T)}{d_{\underline{i},\underline{j}}\log T}\right\}.$$

Notice that $\limsup_{T\to+\infty} h_{i,j}(T) = 0$ under consistency.
Now, iterating this reasoning over $\underline{i} \in [\![d]\!]$ and over $\underline{j} \in [\![k_{\underline{i}}]\!]$, we get the lower bound on the expected number of pulls for all the arms of all the action components.

Part 2: Understanding how the pulls we have to perform on the action components can be combined

From Part 1 of this proof, we have a result on the expectation of the minimum number of pulls. We can now define the quantity:

$$L_{i,j}(T) := \frac{\mathbb{E}[N_{i,j}(T)]}{\log T}, \qquad \forall i \in [\![d]\!],\ j \in [\![k_i]\!].$$

This quantity can be lower bounded as:

$$L_{i,j}(T) \geqslant \frac{2\sigma^2}{\Delta^2_{ij}} - h_{i,j}(T), \qquad \forall i \in [\![d]\!],\ j \in [\![k_i]\!]\backslash\{a^*_i\}.$$

Now, we want to understand how these pulls of the action's suboptimal components influence the regret. We chose to look at the asymptotic expected regret, defined as follows:

$$\frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T} = \sum_{\mathbf{a}\in\mathcal{A}} \frac{\mathbb{E}\left[N_{\mathbf{a}}(T)\right]}{\log T}\Delta_{\mathbf{a}},$$

and we denote:

$$L_{\mathbf{a}}(T) := \frac{\mathbb{E}[N_{\mathbf{a}}(T)]}{\log T},\ \ \forall \mathbf{a} \in \mathcal{A}.$$

The regret becomes defined as:

$$\frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T} = \sum_{\mathbf{a}\in\mathcal{A}} L_{\mathbf{a}}(T)\Delta_{\mathbf{a}},$$

Now, we want to look at how the pulls of the action vectors $L_{\mathbf{a}}$ and the ones of the action components are related. We can easily observe that the following relation occurs:

$$L_{i,j}(T) = \sum_{\mathbf{a} \in \mathcal{A} : a_i = j} L_{\mathbf{a}}(T), \;\; \forall i \in [\![d]\!], \; j \in [\![k_i]\!].$$

Given that, we can write an optimization problem in which we search for the best combination of pulls of the action vector satisfying the constraints on the minimum number of pulls of the action components.

$$\min_{L_{\mathbf{a}}(T), L_{i,j}(T)} \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}*\}} L_{\mathbf{a}}(T) \Delta_{\mathbf{a}} \tag{D.11}$$

$$\text{s.t. } L_{i,j}(T) = \sum_{\mathbf{a} \in \mathcal{A} : a_i = j} L_{\mathbf{a}}(T), \quad \forall i \in [\![d]\!], \; j \in [\![k_i]\!] \setminus \{a_i^*\} \tag{D.12}$$

$$L_{i,j}(T) \geqslant \frac{2\sigma^2}{\Delta_{i,j}^2} - h_{i,j}(T), \quad \forall i \in [\![d]\!], \; j \in [\![k_i]\!] \setminus \{a_i^*\} \tag{D.13}$$

$$L_{\mathbf{a}}(T) \geqslant 0, \quad \forall \mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}. \tag{D.14}$$

Now, to simplify notation, we define $x(\mathbf{a}) = L_{\mathbf{a}}(T)$, remove the variables $L_{i,j}$ since constraint (D.13) will be satisfied with equality, and reformulate in the unconstrained form using the following indicator function:

$$I_{\mathcal{X}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{X} \\ +\infty & \text{otherwise} \end{cases},$$

as follows:

$$\inf_{x(\boldsymbol{a})} f_T(x) := \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}*\}} x(\mathbf{a}) \Delta_{\mathbf{a}} +$$

$$+ \sum_{i \in [\![d]\!]} \sum_{j \in [\![k_i]\!] \setminus \{a_i^*\}} I_{\mathbb{R}_{\geqslant 0}} \left(\sum_{\mathbf{a} \in \mathcal{A} \,:\, a_i = j} x(\boldsymbol{a}) - \frac{2\sigma^2}{\Delta_{i,j}^2} + h_{i,j}(T) \right) +$$

$$+ \sum_{\mathbf{a} \in \mathcal{A}} I_{\mathbb{R}_{\geqslant 0}}(x(\mathbf{a})).$$

With this notation, we want to characterize the value of the optimization problem as the horizon T grows to infinity, i.e., $\liminf_{T \to +\infty} \inf_{x(\boldsymbol{a})} f_T(x)$. Notice that this is exactly what we need to obtain a lower bound to:

$$\liminf_{T \to +\infty} \frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T}.$$

In the following, we show that:

$$\liminf_{T\to+\infty} \inf_{x(\boldsymbol{a})} f_T(x) = \inf_{x(\boldsymbol{a})} f_\infty(x),$$

where f_∞ is defined as follows:

$$\begin{aligned} f_\infty(x) := &\sum_{\mathbf{a}\in\mathcal{A}} x(\mathbf{a})\Delta_{\mathbf{a}} + \sum_{i\in[\![d]\!]} \sum_{j\in[\![k_i]\!]\setminus\{a_i^*\}} I_{\mathbb{R}_{\geqslant 0}} \left(\sum_{\mathbf{a}\in\mathcal{A}\,:\,a_i=j} x(\boldsymbol{a}) - \frac{2\sigma^2}{\Delta_{i,j}^2} \right) + \\ &+ \sum_{\mathbf{a}\in\mathcal{A}} I_{\mathbb{R}_{\geqslant 0}}(x(\mathbf{a})), \end{aligned}$$

corresponding to the optimization problem in which we remove the $h_{i,j}(T)$ function from the right-hand side of the constraint.
First of all, we observe that for every x and T, we have that $f_T(x) \leqslant f_\infty(x)$. It follows that $\inf_{x(\mathbf{a})} f_T(x) \leqslant \inf_{x(\mathbf{a})} f_\infty(x)$ and, consequently, $\liminf_{T\to+\infty} \inf_{x(\boldsymbol{a})} f_T(x) \leqslant \inf_{x(\boldsymbol{a})} f_\infty(x)$. Thus, it remains to prove that $\liminf_{T\to+\infty} \inf_{x(\boldsymbol{a})} f_T(x) \geqslant \inf_{x(\boldsymbol{a})} f_\infty(x)$. Since the optimization problem is linear and feasible (for sufficiently large T), there must exist x_T^* such that $\inf_{x(\boldsymbol{a})} f_T(x) = f_T(x_T^*)$ for every finite T, but also for $T = \infty$. Now, consider for a fixed x:

$$\begin{aligned} \liminf_{T\to+\infty} f_T(x) = &\sum_{\mathbf{a}\in\mathcal{A}} x(\mathbf{a})\Delta_{\mathbf{a}} + \sum_{\mathbf{a}\in\mathcal{A}} I_{\mathbb{R}_{\geqslant 0}}(x(\mathbf{a})) + \\ &+ \liminf_{T\to+\infty} \sum_{i\in[\![d]\!]} \sum_{j\in[\![k_i]\!]\setminus\{a_i^*\}} I_{\mathbb{R}_{\geqslant 0}} \left(\sum_{\mathbf{a}\in\mathcal{A}\,:\,a_i=j} x(\boldsymbol{a}) - \frac{2\sigma^2}{\Delta_{i,j}^2} + h_{i,j}(T) \right) \\ \geqslant &\sum_{\mathbf{a}\in\mathcal{A}} x(\mathbf{a})\Delta_{\mathbf{a}} + \sum_{\mathbf{a}\in\mathcal{A}} I_{\mathbb{R}_{\geqslant 0}}(x(\mathbf{a})) + \\ &+ \sum_{i\in[\![d]\!]} \sum_{j\in[\![k_i]\!]\setminus\{a_i^*\}} \liminf_{T\to+\infty} I_{\mathbb{R}_{\geqslant 0}} \left(\sum_{\mathbf{a}\in\mathcal{A}\,:\,a_i=j} x(\boldsymbol{a}) - \frac{2\sigma^2}{\Delta_{i,j}^2} + h_{i,j}(T) \right) \\ = &f_\infty(x), \end{aligned}$$

uniformly since $\limsup_{T\to+\infty} h_{i,j}(T) = 0$ and $I_{\mathbb{R}_{\geqslant 0}}$ is a decreasing function in its argument, having also exploited that $\liminf_n (a_n + b_n) \geqslant \liminf_n a_n + \liminf_n b_n$. Indeed, let $c = \sum_{\mathbf{a}\in\mathcal{A}\,:\,a_i=j} x(\boldsymbol{a}) - \frac{2\sigma^2}{\Delta_{i,j}^2}$ and $y_T = h_{i,j}(T)$, we have to compute $\liminf_{T\to+\infty} I_{\mathbb{R}_{\geqslant 0}}(c + y_T)$.
Since $0 \leqslant y_T$ and $\limsup_{T\to+\infty} y_T = 0$, we have $\lim_{T\to+\infty} y_T = 0$. If $c \neq 0$, there exists $T(c)$ such that for $T \geqslant T(c)$, we have that $y_T \leqslant |c|/2$. Consequently, $\liminf_{T\to+\infty} I_{\mathbb{R}_{\geqslant 0}}(c + y_T) = I_{\mathbb{R}_{\geqslant 0}}(c)$. If, instead, $c = 0$,

we have to compute $\lim_{T\to+\infty} I_{\mathbb{R}_{\geqslant 0}}(y_T)$; being $I_{\mathbb{R}_{\geqslant 0}}$ right continuous and $y_T \geqslant 0$ we have that $\lim_{T\to+\infty} I_{\mathbb{R}_{\geqslant 0}}(y_T) = 0$.
This, combined with the fact $f_T(x) \leqslant f_\infty(x)$ leads to $\liminf_{T\to+\infty} f_T(x) = f_\infty(x)$, uniformly. Thus, we have that for every $\varepsilon > 0$ there exists $T(\varepsilon) > 0$ such that for every $T \geqslant T_0(\varepsilon)$ we have uniformly:

$$\left| \inf_{T' \geqslant T} f_{T'}(x) - f_\infty(x) \right| \leqslant \varepsilon.$$

Consequently, we have:

$$\begin{aligned} \inf_{T' \geqslant T} \inf_{x(\mathbf{a})} f_{T'}(x) &= \inf_{T' \geqslant T} f_{T'}(x^*_{T'}) \\ &\geqslant f_\infty(x^*_{T'}) - \varepsilon \\ &\geqslant f_\infty(x^*_\infty) - \varepsilon \\ &= \inf_{x(\mathbf{a})} f_\infty(x(\mathbf{a})) - \varepsilon. \end{aligned}$$

This concludes the proof. □

Theorem 8.3.4 (Instance-Dependent Lower Bound (Explicit))**.** *Let $\underline{C}(\boldsymbol{\nu})$ be the solution of the optimization problem of Theorem 8.3.3. It holds that:*

$$\underline{C}(\boldsymbol{\nu}) = \sum_{\ell=1}^{K-d} \left(M_{\boldsymbol{\pi}(\ell)} - M_{\boldsymbol{\pi}(\ell-1)} \right) \Delta_{\boldsymbol{\alpha}_\ell},$$

that can be computed in $\mathcal{O}(\sum_{i \in [\![d]\!]} k_i \log k_i)$.

Proof. Let $M = \max_{i\in[\![d]\!]} M_{i,k_i-1}$. For every $i \in [\![d]\!]$, let us define a non-negative function function $f_i : \mathbb{R} \to \{\mu_{i,j}\}_{j\in[\![k_i]\!]} \cup \{0\}$ such that:

$$\begin{aligned} &\int_{\mathbb{R}} \mathbb{1}\{f_i(x) = \mu_{i,j}\} \mathrm{d}x = L_{i,j} \qquad \forall j \in [\![k_i]\!] \backslash \{a_i^*\}, \\ &\int_{\mathbb{R}} \mathbb{1}\{f_i(x) = \mu_{i,a_i^*}\} \mathrm{d}x = M - M_{i,k_i-1}. \end{aligned}$$

Clearly, f_i is not uniquely defined. Indeed, any function f_i satisfying these conditions is measurable (by definition, since the pre-image of any $\mathcal{Y} \subseteq \{\mu_{i,j}\}_{j\in[\![k_i]\!]} \cup \{0\}$ is measurable) and correspond to a possible arrangement of a proportion of pulls of the arm components of dimension i. Specifically, all functions satisfying these conditions are called "equimesurable" meaning that for every f_i, g_i fulfilling the conditions, we have that $\{x : f_i(x) \geqslant y\} = \{x : g_i(x) \geqslant y\}$ for every $y \in \mathbb{R}$. We call this set of functions $\mathcal{F}_i$.

A possible arrangement of the proportion of the pulls for component $i \in [\![d]\!]$, corresponds to a function $f_i \in \mathcal{F}_i$ such that $f_i(x) = 0$ for $x < 0$ or $x > M$. Thus, to minimize the regret as in the optimization problem of Theorem 8.3.3, we maximize the reward as follows:

$$\sup_{f_i \in \mathcal{F}_i,\, f_i(x) = 0 \text{ for } x<0 \text{ or } x>M,\, i \in [\![d]\!]} \int_{\mathbb{R}^d} \prod_{i \in [\![d]\!]} f_i(x_i) \mathrm{d}x_i$$
$$\leqslant \sup_{f_i \in \mathcal{F}_i,\, i \in [\![d]\!]} \int_{\mathbb{R}^d} \prod_{i \in [\![d]\!]} f_i(x_i) \mathrm{d}x_i.$$

Let f_i^* be the *symmetric decreasing rearrangement* of f_i for every $i \in [\![d]\!]$, which, in our specific case, is a piecewise constant symmetric function. Define $x_0 = 0$, $x_{i,1} = (M - M_{i,k_i-1})/2$, $x_{i,l+1} = x_{i,l} + L_{i,\pi_i(k_i-l)}/2$ for $l \in [\![k_i]\!]$, we have:

$$f_i^*(x) = \sum_{l \in [\![k_i]\!]} \mu_{i,\pi_i(k_i-l+1)} \mathbb{1}\{|x| \in [x_{i,l-1}, x_{i,l})\}.$$

From the rearrangement inequality for multiple integrals (Luttinger and Friedberg, 1976), we have:

$$\sup_{f_i \in \mathcal{F}_i,\, i \in [\![d]\!]} \int_{\mathbb{R}^d} \prod_{i \in [\![d]\!]} f_i(x_i) \mathrm{d}x_i = \int_{\mathbb{R}^d} \prod_{i \in [\![d]\!]} f_i^*(x_i) \mathrm{d}x_i.$$

Let us observe that the product of $\int_{\mathbb{R}^d} \prod_{i \in [\![d]\!]} f_i^*(x_i) \mathrm{d}x_i$ actually leads to the solution depicted in the statement of the theorem.
Concerning the computational complexity, we observe that it is dominated by the sorting in each dimension $i \in [\![d]\!]$. □

Theorem 8.4.1 (Worst-Case Upper Bound for `F-UCB`). *For any FRB $\underline{\boldsymbol{\nu}}$,* `F-UCB` *with $\alpha > 2$ suffers an expected regret bounded as:*

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant 4\sigma \sum_{i \in [\![d]\!]} \sqrt{\alpha k_i T \log T} + g(\alpha) \sum_{i \in [\![d]\!]} k_i,$$

where $g(\alpha) = \widetilde{\mathcal{O}}\left((\alpha-2)^{-2}\right)$.[2]
In particular, if $k_i =: k$, for every $i \in [\![d]\!]$, we have:

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant \widetilde{\mathcal{O}}(\sigma d \sqrt{kT}).$$

[2] The complete expression is reported in the proof.

Proof. The proof is composed of two parts. In the first part, we define the probability, given the chosen confidence bounds, that the good event holds, i.e., the probability that all the confidence bounds are valid. The goal is to find an upper bound on the probability that the good event does not hold along the whole time horizon T. In the second part, we aim to characterize the regret under the good event for a specific round $t \in [\![T]\!]$. Finally, we join the two parts to find an upper bound on the expected cumulative regret.

Part 1: Upper bounding the bad event over time horizon T

We start by defining our good event $\mathcal{E}_t$ at round $t \in [\![T]\!]$, which implies that all the confidence bounds of interest hold, i.e., we are not making a severe underestimate of the expected value of the optimal action components, and severely overestimating the expected values of the suboptimal ones. Formally:

$$\mathcal{E}_t := \left\{ \forall i \in [\![d]\!], \forall a_i \in [\![k_i]\!] \backslash \{a_i^*\} : \widehat{\mu}_{i,a_i}(t) - \mu_{i,a_i} \leqslant \sigma \sqrt{\frac{\alpha \log t}{N_{i,a_i}(t)}} \right\}$$
$$\cap \left\{ \forall i \in [\![d]\!] : \mu_{i,a_i^*} - \widehat{\mu}_{i,a_i^*}(t) \leqslant \sigma \sqrt{\frac{\alpha \log t}{N_{i,a_i^*}(t)}} \right\}.$$

We now want to find an upper bound of the probability of the bad event $\mathcal{E}_t^{\complement}$:

$$\begin{aligned}
&\mathbb{P}\left(\mathcal{E}_t^{\complement}\right) \\
&\leqslant \mathbb{P}\left(\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\} : \widehat{\mu}_{i,a_i}(t) - \mu_{i,a_i} > \sigma \sqrt{\frac{\alpha \log t}{N_{i,a_i}(t)}} \right) + \\
&\quad + \mathbb{P}\left(\exists i \in [\![d]\!] \colon \mu_{i,a_i^*} - \widehat{\mu}_{i,a_i^*}(t) > \sigma \sqrt{\frac{\alpha \log t}{N_{i,a_i^*}(t)}} \right) \\
&\leqslant \mathbb{P}\left(\underbrace{\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\}, \exists s \in [\![t]\!] : \widehat{\mu}_{i,a_i}[s] - \mu_{i,a_i(t)} > \sigma \sqrt{\frac{\alpha \log t}{s}}}_{\text{(A)}} \right) \\
&\quad + \mathbb{P}\left(\underbrace{\exists i \in [\![d]\!], \exists s \in [\![t]\!] : \mu_{i,a_i^*} - \widehat{\mu}_{i,a_i^*}[s] > \sigma \sqrt{\frac{\alpha \log t}{s}}}_{\text{(B)}} \right),
\end{aligned} \tag{D.15}$$

having highlighted with the symbols $\widehat{\mu}_{i,a_i}[s]$ and $\widehat{\mu}_{i,a_i^*}[s]$ the dependence of the estimators on the number of pulls s. We now bound (A) and (B) separately. Similar to the proof of Theorem 2.2 proposed by Bubeck (2010), we use a peeling argument together with Hoeffding's maximal inequality. We apply the peeling argument with a geometric grid over the time interval $[1, t]$ to bound the probability of term (A). Given $\beta \in (0, 1)$, we note that if $s \in \{1, \dots, t\}$, then $\exists j \in \left\{0, \dots, \frac{\log t}{\log 1/\beta}\right\} : \beta^{j+1} t < s \leqslant \beta^j t$. As such, we obtain:

$$\begin{aligned}
\mathbb{P}\left((\mathrm{A})\right) &= \mathbb{P}\Bigg(\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\}, \exists s \in [\![t]\!] : \\
&\qquad\qquad \widehat{\mu}_{i,a_i}[s] - \mu_{i,a_i} > \sigma \sqrt{\frac{\alpha \log t}{s}} \Bigg) \\
&= \mathbb{P}\Bigg(\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\}, \exists s \in [\![t]\!] : \\
&\qquad\qquad \sum_{l=1}^{s} \left(x_{i,a_i}[l] - \mu_{i,a_i(t)} \right) > \sigma \sqrt{\alpha s \log t} \Bigg) \\
&\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \mathbb{P}\Bigg(\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\}, \exists s : \beta^{j+1} t < s \leqslant \beta^j t, \\
&\qquad\qquad \sum_{l=1}^{s} \left(x_{i,a_i}[l] - \mu_{i,a_i(t)} \right) > \sigma \sqrt{\alpha s \log t} \Bigg) \\
&\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \mathbb{P}\Bigg(\exists i \in [\![d]\!], \exists a_i \in [\![k_i]\!] \backslash \{a_i^*\}, \exists s : \beta^{j+1} t < s \leqslant \beta^j t, \\
&\qquad\qquad \sum_{l=1}^{s} \left(x_{i,a_i}[l] - \mu_{i,a_i(t)} \right) > \sigma \sqrt{\alpha \beta^{j+1} t \log t} \Bigg),
\end{aligned}$$

having denoted with $x_{i,a_i}[l]$ the l-sample used to compute the sample mean $\widehat{\mu}_{i,a_i}[s]$. Applying a union bound on the summations on i and a_i, and Hoeffding's maximal inequality, we obtain:

$$\mathbb{P}\left((\mathrm{A})\right) \leqslant \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!] \backslash \{a_i^*\}} \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \exp\left(- \frac{\left(\sqrt{\sigma^2 \alpha \beta^{j+1} t \log t}\right)^2}{2\sigma^2 \beta^j t} \right)$$

$$
\begin{aligned}
&= \sum_{i \in \llbracket d \rrbracket} \sum_{a_i \in \llbracket k_i \rrbracket \setminus \{a_i^*\}} \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \exp\left(-\frac{\alpha\beta \log t}{2}\right) \\
&= \sum_{i \in \llbracket d \rrbracket} \sum_{a_i \in \llbracket k_i \rrbracket \setminus \{a_i^*\}} \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} t^{-\frac{\alpha\beta}{2}} \\
&\leqslant \sum_{i \in \llbracket d \rrbracket} \sum_{a_i \in \llbracket k_i \rrbracket \setminus \{a_i^*\}} \left(\frac{\log t}{\log \frac{1}{\beta}} + 1\right) t^{-\frac{\alpha\beta}{2}}.
\end{aligned}
$$

Applying the same procedure, we can bound the probability of term (B) in Equation (D.15) to obtain:

$$
\mathbb{P}\left((\mathrm{B})\right) \leqslant \sum_{i \in \llbracket d \rrbracket} \left(\frac{\log t}{\log \frac{1}{\beta}} + 1\right) t^{-\frac{\alpha\beta}{2}}.
$$

As such, we can write the upper bound of the probability of the bad event as:

$$
\mathbb{P}\left(\mathcal{E}_t^{\complement}\right) = \mathbb{P}\left((\mathrm{A})\right) + \mathbb{P}\left((\mathrm{B})\right) \leqslant \sum_{i \in \llbracket d \rrbracket} k_i \left(\frac{\log t}{\log \frac{1}{\beta}} + 1\right) t^{-\frac{\alpha\beta}{2}}.
$$

Let us now bound the sum of the probabilities of the bad event over the horizon T:

$$
\begin{aligned}
\sum_{t \in \llbracket T \rrbracket} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) &\leqslant \sum_{i \in \llbracket d \rrbracket} k_i \sum_{t \in \llbracket T \rrbracket} \left(\frac{\log t}{\log \frac{1}{\beta}} + 1\right) t^{-\frac{\alpha\beta}{2}} \\
&\leqslant \sum_{i \in \llbracket d \rrbracket} k_i \int_1^T \left(\frac{\log t}{\log \frac{1}{\beta}} + 1\right) t^{-\frac{\alpha\beta}{2}} \mathrm{d}t && \text{(D.16)} \\
&= \sum_{i \in \llbracket d \rrbracket} k_i \Bigg(\left[\left(\frac{\log t}{\log 1/\beta} + 1\right)\left(\frac{2}{2-\alpha\beta} t^{1-\frac{\alpha\beta}{2}}\right)\right]_1^{+\infty} + \\
&\qquad - \frac{4}{(2-\alpha\beta)\log 1/\beta} \int_1^{+\infty} t^{-\frac{\alpha\beta}{2}} \mathrm{d}t \Bigg) && \text{(D.17)} \\
&= \sum_{i \in \llbracket d \rrbracket} k_i \left(-\frac{2}{2-\alpha\beta} - \frac{4}{(2-\alpha\beta)^2 \log(1/\beta)} \left[t^{1-\frac{\alpha\beta}{2}}\right]_1^{+\infty}\right) \\
& && \text{(D.18)}
\end{aligned}
$$

$$= \sum_{i\in\llbracket d\rrbracket} k_i \left(-\frac{2}{2-\alpha\beta} + \frac{4}{(2-\alpha\beta)^2\log(1/\beta)}\right), \tag{D.19}$$

where line (D.16) is obtained by bounding the summation with the integral, line (D.17) is obtained via integration by parts, and the first term of line (D.18) is obtained by imposing $\alpha\beta > 2$. Substituting now $\beta = \frac{4}{\alpha+2}$, which verifies $\beta \in (0,1)$ if $\alpha > 2$, we obtain:

$$\begin{aligned}\sum_{t\in\llbracket T\rrbracket} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) &\leqslant \left(\frac{\alpha+2}{\alpha-2} + \frac{(\alpha+2)^2}{(\alpha-2)^2}\frac{1}{\log\left(\frac{\alpha+2}{4}\right)}\right)\sum_{i\in\llbracket d\rrbracket} k_i \\ &= \widetilde{\mathcal{O}}\left((\alpha-2)^2\right)\sum_{i\in\llbracket d\rrbracket} k_i.\end{aligned}$$

Part 2: Upper bounding the instantaneous regret at time t under the good event

We can now bound the instantaneous regret at time t supposing the good event holds. We define the regret R_t at time t as the difference in expectation between the optimal action and the one performed by `F-UCB`, formally:

$$\begin{aligned}R_t &= \prod_{i\in\llbracket d\rrbracket}\mu_i^* - \prod_{i\in\llbracket d\rrbracket}\mu_{i,a_i(t)} &\text{(D.20)}\\ &= \sum_{l\in\llbracket d\rrbracket}\underbrace{\prod_{i\in\llbracket l-1\rrbracket}\mu_l^*}_{\in[0,1]}\left(\mu_l^* - \mu_{l,a_l(t)}\right)\underbrace{\prod_{i\in\llbracket l+1,d\rrbracket}\mu_{i,a_i(t)}}_{\in[0,1]} &\text{(D.21)}\\ &\leqslant \sum_{l\in\llbracket d\rrbracket}\left(\mu_l^* - \mu_{l,a_l(t)}\right) &\text{(D.22)}\\ &= \sum_{l\in\llbracket d\rrbracket}\left(\mu_l^* - \mu_{l,a_l(t)} \pm \mathrm{UCB}_{l,a_l(t)}(t)\right) &\text{(D.23)}\\ &\leqslant \sum_{l\in\llbracket d\rrbracket}\left(\mathrm{UCB}_{l,a_l(t)}(t) - \mu_{l,a_l(t)}\right) &\text{(D.24)}\\ &= \sum_{l\in\llbracket d\rrbracket}\left(\widehat{\mu}_{l,a_l(t)}(t) + \beta_{l,a_l(t)}(t) - \mu_{l,a_l(t)}\right) &\text{(D.25)}\\ &\leqslant 2\sum_{l\in\llbracket d\rrbracket}\beta_{l,a_l(t)}(t), &\text{(D.26)}\end{aligned}$$

where line (D.21) is obtained by summing and subtracting all mixed terms, line (D.22) follows from bounding the left and right products with 1 being all factors (including the middle one) made of non-negative terms, line (D.24) comes from the optimism under the good event, having denoted with $\beta_{l,a_l}(t)$ the exploration bonus.

Upper bound of the expected cumulative regret $R(\texttt{F-UCB}, T)$

Recalling that we call R_t the instantaneous regret under the good event, can now compute an upper bound on the expected cumulative regret as:

$$
\begin{aligned}
&\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \left(1 \cdot \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + R_t \cdot \mathbb{P}\left(\tilde{\mathcal{E}}_t\right)\right) \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + \sum_{t \in [\![T]\!]} R_t \cdot \mathbb{P}\left(\tilde{\mathcal{E}}_T\right) \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + \sum_{t \in [\![T]\!]} R_t \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + \sum_{t \in [\![T]\!]} 2 \sum_{i \in [\![d]\!]} \beta_{i,a_i(t)}(t) \\
&\quad = \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2 \sum_{t \in [\![T]\!]} \sum_{i \in [\![d]\!]} \sigma \sqrt{\frac{\alpha \log t}{N_{i,a_i(t)}}} \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{t \in [\![T]\!]} \sum_{i \in [\![d]\!]} \sqrt{\frac{1}{N_{i,a_i(t)}}} \\
&\quad = \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!]} \sum_{j \in [\![N_{i,a_i}(T)]\!]} \sqrt{\frac{1}{j}} \qquad \text{(D.27)} \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!]} \sum_{j \in [\![T/k_i]\!]} \sqrt{\frac{1}{j}} \qquad \text{(D.28)} \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!]} \int_1^{T/k_i} \sqrt{\frac{1}{j}} \mathrm{dj} \qquad \text{(D.29)} \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!]} \left(1 + 2\sqrt{\frac{T}{k_i}} - 2\right) \\
&\quad \leqslant \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 2\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sum_{a_i \in [\![k_i]\!]} 2\sqrt{\frac{T}{k_i}} \\
&\quad = \sum_{t \in [\![T]\!]} \mathbb{P}\left(\mathcal{E}_t^{\complement}\right) + 4\sigma\sqrt{\alpha \log T} \sum_{i \in [\![d]\!]} \sqrt{k_i T}
\end{aligned}
$$

$$\leqslant \left(\frac{\alpha+2}{\alpha-2} + \frac{(\alpha+2)^2}{(\alpha-2)^2}\frac{1}{\log\left(\frac{\alpha+2}{4}\right)}\right)\sum_{i\in[\![d]\!]} k_i + 4\sigma\sqrt{\alpha T\log T}\sum_{i\in[\![d]\!]}\sqrt{k_i}.$$

where line (D.27) is obtained by rewriting the series over the arms and the number of pulls for each arm, line (D.28) is derived by considering the worst case, i.e., when all the arms are pulled equally (this is the worst case because we are looking at a concave function), and line (D.29) is obtained by bounding the summation with the corresponding integral. This concludes the proof.

□

Theorem 8.4.2 (Instance-Dependent Upper Bound for `F-UCB`)**.** *For a given FRB $\underline{\boldsymbol{\nu}}$,* `F-UCB` *with $\alpha > 2$ suffers an expected regret bounded as:*

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant \overline{C}(\texttt{F-UCB}, \underline{\boldsymbol{\nu}}),$$

where $\overline{C}(\textit{F-UCB}, \underline{\boldsymbol{\nu}})$ is defined as the solution to the following optimization problem (where $g(\alpha) = \widetilde{\mathcal{O}}((\alpha-2)^{-2})$):

$$\max_{(N_{\mathbf{a}})_{\mathbf{a}\in\mathcal{A}}} \sum_{\mathbf{a}\in\mathcal{A}\setminus\{\mathbf{a}^*\}} N_{\mathbf{a}}\Delta_{\mathbf{a}} \tag{8.9}$$

$$\text{s.t.} \quad N_{i,j} = \sum_{\substack{\mathbf{a}\in\mathcal{A}\setminus\{\mathbf{a}^*\}\\ a_i=j}} N_{\mathbf{a}}, \quad \forall i \in [\![d]\!],\ j \in [\![k_i]\!]\setminus\{a_i^*\} \tag{8.10}$$

$$N_{i,j} \leqslant \frac{4\alpha\sigma^2\log T}{\Delta_{i,j}^2} + g(\alpha), \quad \forall i \in [\![d]\!],\ j \in [\![k_i]\!]\setminus\{a_i^*\} \tag{8.11}$$

$$\sum_{\mathbf{a}\in\mathcal{A}} N_{\mathbf{a}} = T \tag{8.12}$$

$$N_{\mathbf{a}} \geqslant 0, \quad \forall \mathbf{a} \in \mathcal{A} \tag{8.13}$$

Proof. The proof of this statement is divided into two parts. The first part is dedicated to finding an upper bound on the expected number of pulls for each action component N_{ij}. The second part is dedicated to understanding how these pulls can be combined to find an upper bound on the regret.

Part 1: Upper bounding the expected number of pulls for each action component

The proof of the expected number of pulls for σ^2-subgaussian variables comprises three parts, extending and following the proof of Theorem 2.2 proposed by Bubeck (2010).

Given an instance $\underline{\boldsymbol{\nu}}$ of FRB, consider a component $i \in [\![d]\!]$, and a suboptimal action $a_i \in [\![k_i]\!]\setminus\{a_i^*\}$, which suffers a suboptimality gap of Δ_{i,a_i}. In this part, we show that if $I_{i,t} = a_i$ (i.e., the action selected for component i at time t is a_i), then one of the three following equations is true:

$$\mathrm{UCB}_{i,a_i^*}(t) \leqslant \mu_i^*, \tag{D.30}$$

or

$$\hat{\mu}_{i,a_i}(t-1) > \mu_{i,a_i} + \sigma\sqrt{\frac{\alpha \log t}{N_{i,a_i}(t-1)}}, \tag{D.31}$$

or

$$N_{i,a_i}(t-1) < \frac{4\sigma^2\alpha \log T}{\Delta_{i,a_i}^2}, \tag{D.32}$$

where: $\mathrm{UCB}_{i,a_i^*}(t)$ is the confidence bound of the optimal arm for component i at time t, having pulled such an arm for $N_{i,a_i^*}(t-1)$ times in the previous rounds, and $\hat{\mu}_{i,a_i,N_{i,a_i}(t-1)}$ is the estimated value of the mean of arm a_i of component i after $N_{i,a_i}(t-1)$ pulls. For absurd, if we assume that the three equations are false, then we have:

$$\begin{aligned}
\mathrm{UCB}_{i,a_i^*}(t) &> \mu_i^* \\
&= \mu_{i,a_i} + \Delta_{i,a_i} \\
&\geqslant \mu_{i,a_i} + 2\sqrt{\frac{\sigma^2\alpha \log t}{N_{i,a_i}(t-1)}} \\
&\geqslant \hat{\mu}_{i,a_i,N_{i,a_i}(t-1)} + \sqrt{\frac{\sigma^2\alpha \log t}{N_{i,a_i}(t-1)}} \\
&= \mathrm{UCB}_{i,a_i}(t-1),
\end{aligned}$$

which implies that $a_i(t) \neq a_i$. Now, we bound the probability that Equation (D.30) or Equation (D.31) hold true. Similar to the original proof, we use a peeling argument together with Hoeffding's maximal inequality, which is a consequence of Azuma-Hoeffding inequality. Note that:

$$\begin{aligned}
&\mathbb{P}(\text{Eq. (D.30) is true}) \\
&\quad \leqslant \mathbb{P}\left(\exists s \in \{1,\ldots,t\} : \hat{\mu}_{i,a_i^*}[s] + \sqrt{\frac{\sigma^2\alpha \log t}{s}} \leqslant \mu_i^*\right) \\
&\quad = \mathbb{P}\left(\exists s \in \{1,\ldots,t\} : \sum_{l=1}^{s}(x_{i,a_i^*}[l] - \mu_i^*) \leqslant -\sqrt{\sigma^2\alpha s \log t}\right)
\end{aligned}$$

We now apply the peeling argument with a geometric grid over the time interval $[1, t]$. More precisely, given $\beta \in (0, 1)$, we note that if $s \in \{1, \ldots, t\}$, then $\exists j \in \left\{0, \ldots, \frac{\log t}{\log 1/\beta}\right\} : \beta^{j+1} t < s \leqslant \beta^j t$.
As such, we get:

$$\begin{aligned}
&\mathbb{P}(\text{Eq. (D.30) is true}) \\
&\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \mathbb{P}\left(\exists s\colon \beta^{j+1} t < s \leqslant \beta^j t, \sum_{l=1}^{s} (x_{i,a_i^*}[l] - \mu_i^*) \leqslant - \sqrt{\sigma^2 \alpha s \log t}\right) \\
&\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \mathbb{P}\left(\exists s\colon \beta^{j+1} t < s \leqslant \beta^j t, \sum_{l=1}^{s} (x_{i,a_i^*}[l] - \mu_i^*) \leqslant - \sqrt{\sigma^2 \alpha \beta^{j+1} t \log t}\right)
\end{aligned}$$

We now bound this last term using Hoeffding's maximal inequality, which gives:

$$\begin{aligned}
\mathbb{P}(\text{Eq. (D.30) is true}) &\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \exp\left(-\frac{\left(\sqrt{\sigma^2 \alpha \beta^{j+1} t \log t}\right)^2}{2\sigma^2 \beta^j t}\right) \\
&\leqslant \sum_{j=0}^{\frac{\log t}{\log 1/\beta}} \exp\left(-\frac{\alpha\beta \log t}{2}\right) \\
&\leqslant \left(\frac{\log t}{\log 1/\beta} + 1\right) \frac{1}{t^{\frac{\beta\alpha}{2}}}.
\end{aligned}$$

Using the same arguments, it can be proven that:

$$\mathbb{P}(\text{Eq. (D.31) is true}) \leqslant \left(\frac{\log t}{\log 1/\beta} + 1\right) \frac{1}{t^{\frac{\beta\alpha}{2}}}.$$

We can now write:

$$\begin{aligned}
\mathbb{E}\left[N_{i,a_i}(T)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}_{\{I_{i,t} = a_i\}}\right] \\
&\leqslant u + \mathbb{E}\left[\sum_{t=u+1}^{T} \mathbb{1}_{\{I_{i,t} = a_i \text{ and Eq. (D.32) is false}\}}\right]
\end{aligned}$$

$$= u + \mathbb{E}\left[\sum_{t=u+1}^{T} \mathbb{1}_{\{\text{Eq. (D.30) or Eq. (D.31) is true}\}}\right]$$
$$\leqslant u + \sum_{t=u+1}^{T} \left(\mathbb{P}(\text{Eq. } \textbf{(D.30)} \text{ is true}) + \mathbb{P}(\text{Eq. } \textbf{(D.31)} \text{ is true})\right),$$

where $u = \lceil \frac{4\sigma^2 \alpha \log T}{\Delta^2_{i,a_i}} \rceil$.

We can now upper bound the probability of Equations (D.30) and (D.31) holds:

$$\sum_{t=u+1}^{T} \left(\mathbb{P}(\text{Eq. } \textbf{(D.30)} \text{ is true}) + \mathbb{P}(\text{Eq. } \textbf{(D.31)} \text{ is true})\right)$$
$$\leqslant 2 \sum_{t=u+1}^{T} \left(\frac{\log t}{\log 1/\beta} + 1\right) \frac{1}{t^{\frac{\beta\alpha}{2}}}$$
$$\leqslant 2 \int_{1}^{+\infty} \left(\frac{\log t}{\log 1/\beta} + 1\right) \frac{1}{t^{\frac{\beta\alpha}{2}}} dt$$
$$= 2\left[\left(\frac{\log t}{\log 1/\beta} + 1\right)\left(\frac{2}{2-\alpha\beta} t^{1-\frac{\alpha\beta}{2}}\right)\right]_1^{+\infty} +$$
$$- \frac{4}{(2-\alpha\beta)\log 1/\beta} \int_1^{+\infty} t^{-\frac{\alpha\beta}{2}} \mathrm{d}t \tag{D.33}$$
$$= -\frac{4}{2-\alpha\beta} - \frac{8}{(2-\alpha\beta)^2 \log 1/\beta} \left[t^{1-\frac{\alpha\beta}{2}}\right]_1^{+\infty} \tag{D.34}$$
$$= -\frac{4}{2-\alpha\beta} + \frac{8}{(2-\alpha\beta)^2 \log 1/\beta},$$

where line (D.33) is obtained via integration by parts and the first term of line (D.34) is obtained imposing $\alpha\beta > 2$. Substituting now $\beta = \frac{4}{\alpha+2}$, which verifies $\beta \in (0,1)$ if $\alpha > 2$, we obtain:

$$\sum_{t=u+1}^{T} \left(\mathbb{P}(\text{Eq. } \textbf{(D.30)} \text{ is true}) + \mathbb{P}(\text{Eq. } \textbf{(D.31)} \text{ is true})\right)$$
$$\leqslant -\frac{4}{2-\frac{4\alpha}{\alpha+2}} + \frac{8}{\left(2-\frac{4\alpha}{\alpha+2}\right)^2} \frac{1}{\log\left(\frac{\alpha+2}{4}\right)}$$
$$= -\frac{2(\alpha+2)}{2-\alpha} + \frac{2(\alpha+2)^2}{(2-\alpha)^2} \frac{1}{\log\left(\frac{\alpha+2}{4}\right)}$$

$$= \frac{2(\alpha+2)}{\alpha-2} + \frac{2}{\log\left(\frac{\alpha+2}{4}\right)}\left(\frac{\alpha+2}{\alpha-2}\right)^2.$$

Rearranging the upper bound on the expected number of pulls given the three cases presented above, we get:

$$\mathbb{E}[N_{i,j}(T)] \leqslant \frac{4\alpha\sigma^2\log T}{\Delta_{i,j}^2} + \frac{2(\alpha+2)}{\alpha-2} + \frac{2}{\log\left(\frac{\alpha+2}{4}\right)}\left(\frac{\alpha+2}{\alpha-2}\right)^2.$$

We set $g(\alpha) = \frac{2(\alpha+2)}{\alpha-2} + \frac{2}{\log\left(\frac{\alpha+2}{4}\right)}\left(\frac{\alpha+2}{\alpha-2}\right)^2 = \widetilde{\mathcal{O}}\left((\alpha-2)^{-2}\right).$

Part 2: Upper bounding the expected cumulative regret

We now have to understand how the pulls defined in part 1 can be combined. We want to look at the worst combination in which we can pull the suboptimal action components.

We recall that regret can be defined by highlighting the dependence on the pulls of the action vectors:

$$\mathbb{E}[R_T(\texttt{F-UCB}, \underline{\nu})] = \sum_{\mathbf{a}\in\mathcal{A}} N_{\mathbf{a}}\Delta_{\mathbf{a}}.$$

As before, we can bind the pulls of the action components N_{ij} and the action vectors $N_{\mathbf{a}}$ as follows:

$$\mathbb{E}[N_{i,j}(T)] = \sum_{\mathbf{a}\in\mathcal{A}:a_i=j} N_{\mathbf{a}}, \quad \forall i \in [\![d]\!],\ j \in [\![k_i]\!].$$

We know that the pulls cannot be negative, and that the total number of pulls of the action vectors sums to T, so we impose these additional constraints. Now, acting on the number of pulls $N_{\mathbf{a}}$, $\forall \mathbf{a} \in \mathcal{A}$ we want to find the worst-case in which we can combine action components in action vectors. So, we solve a maximization problem on the regret defined as a function of the number of pulls, given the constraints defined above, and the upper bound on the expected number of pulls of the action components N_{ij}, $\forall i \in [\![d]\!]$, $j \in [\![k_i]\!]\backslash\{a_i^*\}$ defined in Part 1 of this proof. □

Corollary 8.4.3 (Explicit Instance-Dependent Upper Bound for F-UCB). *For a given FRB $\underline{\nu}$,* F-UCB *with $\alpha > 2$ suffers an expected regret bounded by:*

$$\begin{aligned}\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\nu})\right] &\leqslant \overline{C}(\texttt{F-UCB}, \underline{\nu}) \\ &\leqslant 4\alpha\sigma^2\log T \sum_{i\in[\![d]\!]} \mu_{-i}^* \sum_{j\in[\![k_i]\!]\backslash\{a_i^*\}} \Delta_{i,j}^{-1} + g(\alpha)\sum_{i\in[\![d]\!]} k_i,\end{aligned}$$

where $\mu_{-i}^* = \prod_{l\in[\![d]\!]\backslash\{i\}} \mu_l^* \leqslant 1$ for every $i \in [\![d]\!]$.

Proof. In order to obtain a relaxed solution of the optimization problem in Theorem 8.4.2, we first derive the following upper bound to the suboptimality gaps of the action vector $\mathbf{a} = (a_1, \dots, a_d)^{\mathrm{T}}$:

$$\begin{aligned}
\Delta_{\mathbf{a}} &= \prod_{i \in [\![d]\!]} \mu_i^* - \prod_{i \in [\![d]\!]} \mu_{i,a_i} = \prod_{i \in [\![d]\!]} \mu_i^* \left(1 - \prod_{i \in [\![d]\!]} \frac{\mu_{i,a_i}}{\mu_i^*}\right) && \text{(D.35)}\\
&\leqslant \prod_{i \in [\![d]\!]} \mu_i^* \left(1 - \min_{i \in [\![d]\!]} \frac{\mu_{i,a_i}}{\mu_i^*}\right) && \text{(D.36)}\\
&= \prod_{i \in [\![d]\!]} \mu_i^* \max_{i \in [\![d]\!]} \left(1 - \frac{\mu_{i,a_i}}{\mu_i^*}\right) && \text{(D.37)}\\
&\leqslant \prod_{i \in [\![d]\!]} \mu_i^* \sum_{i \in [\![d]\!]} \left(1 - \frac{\mu_{i,a_i}}{\mu_i^*}\right) && \text{(D.38)}\\
&= \sum_{i \in [\![d]\!]} (\mu_i^* - \mu_{i,a_i}) \prod_{j \in [\![d]\!] \setminus \{j\}} \mu_j^* && \text{(D.39)}\\
&= \sum_{i \in [\![d]\!]} \Delta_{i,a_i} \mu_{-i}^*, && \text{(D.40)}
\end{aligned}$$

where line (D.36) follows from observing that $\prod_{i \in [\![d]\!]} \frac{\mu_{i,a_i}}{\mu_i^*} \leqslant \min_{i \in [\![d]\!]} \frac{\mu_{i,a_i}}{\mu_i^*}$ since $\frac{\mu_{i,a_i}}{\mu_i^*} \in [0,1)$, line (D.39) comes from defining $\mu_{-i}^* := \prod_{j \in [\![d]\!] \setminus \{j\}} \mu_j^* \leqslant 1$. Thus, by considering the objective function in the optimization problem of Theorem 8.4.2, we have:

$$\begin{aligned}
\sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}} N_{\mathbf{a}} \Delta_{\mathbf{a}} &\leqslant \sum_{\mathbf{a} \in \mathcal{A} \setminus \{\mathbf{a}^*\}} N_{\mathbf{a}} \sum_{i \in [\![d]\!]} \Delta_{i,a_i} \mu_{-i}^* \\
&= \sum_{i \in [\![d]\!]} \mu_{-i}^* \sum_{j \in [\![k_i]\!]} \sum_{\mathbf{a} \in \mathcal{A} \,:\, a_i = j} N_{\mathbf{a}} \Delta_{i,a_i} \\
&= \sum_{i \in [\![d]\!]} \mu_{-i}^* \sum_{a_i \in [\![k_i]\!] \setminus \{a_i^*\}} N_{i,a_i} \Delta_{i,a_i}.
\end{aligned}$$

By using the Constraint (8.11) to upper bound N_{i,a_i} and recalling that $\Delta_{i,j} \leqslant 1$, we get the result.

□

Theorem 8.5.1 (Instance-Dependent Upper Bound for `F-Track`)**.** *For any FRB $\underline{\boldsymbol{\nu}}$,* `F-Track` *run with:*

$$N_0 = \left\lceil \sqrt{\log T} \right\rceil \quad \textit{and} \quad \epsilon_T = \sqrt{\frac{2\sigma^2 f_T(1/\log T)}{N_0}},$$

suffers an expected regret of:

$$\limsup_{T\to+\infty} \frac{\mathbb{E}\left[R_T(\texttt{F-Track},\underline{\boldsymbol{\nu}})\right]}{\log T} = \underline{C}(\underline{\boldsymbol{\nu}}).$$

Proof. **Preliminary Results** Let us introduce the symbol:

$$\epsilon_{i,j}(t,\delta) := \sqrt{\frac{2\sigma^2 f_T(\delta)}{N_{i,j}(t)}}.$$

Consider the event $\mathcal{E}(\delta) := \{\exists i \in [\![d]\!],\, \exists j \in [\![k_i]\!],\, \exists t \in [\![T_{\text{warm-up}}, T]\!] \geqslant 1 : |\widehat{\mu}_{i,j}(t) - \mu_{i,j}| > \epsilon_{i,j}(t,\delta)\}$ and let us bound its probability:

$$\mathbb{P}(\mathcal{E}(\delta)) \leqslant \sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]} \mathbb{P}\left(\exists t \in [\![T_{\text{warm-up}}, T]\!] \;:\; |\widehat{\mu}_{i,j}(t) - \mu_{i,j}| > \epsilon_{i,j}(t,\delta)\right) \tag{D.41}$$

$$= \sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]} \mathbb{P}\left(\exists s \in [\![T]\!] \;:\; |\widehat{\mu}_{i,j}[s] - \mu_{i,j}| > \sqrt{\frac{2\sigma^2 f_T(\delta)}{s}}\right) \tag{D.42}$$

$$\leqslant \sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]} \delta = k\delta, \tag{D.43}$$

where line (D.41) follows from a union bound over the values of i and j, line (D.42) follows by rewriting the probability by highlighting the dependence of the estimator on the number of samples s, and line (D.43) follows from Lemma D.1.1, recalling that $s(\widehat{\mu}_{i,j}[s] - \mu_{i,j})$ is a martingale difference sequence and it is σ^2-subgaussian.
We will make use of the following two instantiations of event $\mathcal{E}(\delta)$:

$$\mathcal{E}_1 := \mathcal{E}(1/\log T) \qquad \text{and} \qquad \mathcal{E}_2 := \mathcal{E}(1/T).$$

Clearly, from the previous result, we have that $\mathbb{P}(\mathcal{E}_1) \leqslant k/\log T$ and $\mathbb{P}(\mathcal{E}_2) \leqslant k/T$.
We start decomposing the regret over the phases of the algorithm:

$$\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}[R(\texttt{F-Track}, T)]$$

$$= \underbrace{\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}\left[\sum_{t\in\textit{warm-up}} \Delta_{\mathbf{a}(t)}\right]}_{=:\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{warm-up}}(T)]} + \underbrace{\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}\left[\sum_{t\in\textit{success}} \Delta_{\mathbf{a}(t)}\right]}_{=:\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{success}}(T)]} + \underbrace{\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}\left[\sum_{t\in\textit{recovery}} \Delta_{\mathbf{a}(t)}\right]}_{=:\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)]},$$

where, with little abuse of notation, we denoted with $t \in phase$ denotes the rounds in which phase *phase* is active. We proceed to analyze the three components separately.

Part 1: Regret in Warm-Up Phase $\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{warm-up}}(T)]$ We start by analyzing the regret in the *warm-up* phase, whose duration is given by $T_{\text{warm-up}} = N_0 \max_{i \in [\![d]\!]} k_i = \lceil \sqrt{\log T} \rceil \max_{i \in [\![d]\!]} k_i$. Thus, the corresponding expected cumulative regret can be bounded as follows:

$$\mathop{\mathbb{E}}_{\underline{\boldsymbol{\nu}}}[R_{\text{warm-up}}(T)] \leqslant \Delta_{\mathbf{max}} \left\lceil \sqrt{\log T} \right\rceil \max_{i \in [\![d]\!]} k_i = \mathcal{O}\left(\sqrt{\log T}\right),$$

where $\Delta_{\mathbf{max}} = \max_{\mathbf{a} \in \mathcal{A}} \Delta_{\mathbf{a}}$ and the Big-O notation retains the dependence on T only. Thus, its contribution to the regret is asymptotically negligible:

$$\limsup_{T \to +\infty} \frac{\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{warm-up}}(T)]}{\log T} = 0.$$

Part 2: Regret in the Recovery Phase $\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)]$ We move to the analysis of the regret in the *recovery* phase. We start by showing that if event $\mathcal{E}_1$ does not hold, then, the recovery phase never activates. Indeed, under $\mathcal{E}_1^\complement$ simultaneously for all $i \in [\![d]\!]$, $j \in [\![k_i]\!]$, and $t \in [\![T_{\text{warm-up}}, T]\!]$ we have that:

$$|\widehat{\mu}_{i,j}(t) - \mu_{i,j}| \leqslant \epsilon_{i,j}(t, 1/\log T),$$

which implies simultaneously for all $i \in [\![d]\!]$, $j \in [\![k_i]\!]$, and $t \in [\![T_{\text{warm-up}}, T]\!]$ that:

$$\begin{aligned}|\widehat{\mu}_{i,j}(T_{\text{warm-up}}) - \widehat{\mu}_{i,j}(t-1)| &\leqslant |\widehat{\mu}_{i,j}(T_{\text{warm-up}}) - \mu_{i,j}| + |\widehat{\mu}_{i,j}(t-1) - \mu_{i,j}| \\ &\leqslant \epsilon_{i,j}(T_{\text{warm-up}}, 1/\log T) + \epsilon_{i,j}(t-1, 1/\log T) \\ &\leqslant 2\epsilon_{i,j}(T_{\text{warm-up}}, 1/\log T),\end{aligned}$$

being $\epsilon_{i,j}(t, 1/\log T)$ a decreasing in t. Recalling that $N_{i,j}(T_{\text{warm-up}}) \geqslant N_0$, we have:

$$\begin{aligned}2\epsilon_{i,j}(T_{\text{warm-up}}, 1/\log T) &= 2\sqrt{\frac{2\sigma^2 f_T(1/\log T)}{N_{i,j}(T_{\text{warm-up}})}} \\ &\leqslant 2\sqrt{\frac{2\sigma^2 f_T(1/\log T)}{N_0}} \\ &= 2\epsilon_T.\end{aligned}$$

Thus, we conclude that the termination condition of the while loop never activates and, consequently, the recovery phase activates only when $\mathcal{E}_1$ holds, i.e., with probability at most $1/\log T$.

In the recovery phase, our `F-Track` algorithm plays `F-UCB` that, from Corollary 8.4.3, is proved to suffer logarithmic regret of the form:

$$\rho(T) := 4\alpha\sigma^2 \log T \sum_{i \in [\![d]\!]} \mu^*_{-i} \sum_{j \in [\![k_i]\!] \setminus \{a^*_i\}} \Delta^{-1}_{i,j} + g(\alpha) \sum_{i \in [\![d]\!]} k_i = \mathcal{O}(\log T).$$

Thus, we have that the cumulative regret of the recovery phase is bounded by:

$$\begin{aligned}\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)] &= \mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)|\mathcal{E}_1^\complement]\,\mathbb{P}(\mathcal{E}_1^\complement) + \mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)|\mathcal{E}_1]\,\mathbb{P}(\mathcal{E}_1) \\ &\leqslant 0 + \frac{\rho(T)}{\log T} \\ &= \mathcal{O}(1).\end{aligned}$$

Consequently, its contribution to the expected cumulative regret is asymptotically negligible. Indeed:

$$\limsup_{T \to +\infty} \frac{\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{recovery}}(T)]}{\log T} = 0.$$

Part 3: Regret in the Success Phase $\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{success}}(T)]$ We conclude with the most challenging part consisting of bounding the regret in the success phase. The cumulative regret in the success phase needs to be further decomposed as follows:

$$\begin{aligned}\mathbb{E}_{\underline{\boldsymbol{\nu}}}[R_{\text{success}}(T)] = &\mathbb{E}_{\underline{\boldsymbol{\nu}}}\left[\mathbf{1}\{\mathcal{E}_1^\complement\} \sum_{t \in success} \Delta_{\mathbf{a}(t)}\right] + \\ &+ \mathbb{E}_{\underline{\boldsymbol{\nu}}}\left[\mathbf{1}\{\mathcal{E}_1 \wedge \mathcal{E}_2^\complement\} \sum_{t \in success} \Delta_{\mathbf{a}(t)}\right] + \\ &+ \mathbb{E}_{\underline{\boldsymbol{\nu}}}\left[\mathbf{1}\{\mathcal{E}_2\} \sum_{t \in success} \Delta_{\mathbf{a}(t)}\right]\end{aligned}$$

We analyze each term separately.

<u>Part 3.1: Regret under $\mathcal{E}_1^\complement$</u> In what follows, all estimated quantities are estimated with the samples available at the end of the warm-up phase and, thus, we will omit the dependence on $T_{\text{warm-up}}$. We show that asymptotically, during the success phase and under event $\mathcal{E}_1^\complement$, the algorithm suffers

the optimal regret. To this end, we need to introduce some auxiliary tools. For every $i \in [\![d]\!]$, let us define a *sorting function* as any bijective function $\pi_i : [\![k_i]\!] \to [\![k_i]\!]$ such that:

$$\mu_{i,\pi_i(1)} \leqslant \cdots \leqslant \mu_{i,\pi_i(k_i)}.$$

If all $\mu_{i,j}$ are different, the sorting function is unique. Furthermore, for every $i \in [\![d]\!]$ and $j \in [\![k_i]\!] \backslash \{\pi_i(k_i)\}$ (i.e., excluding the action component with maximum expected reward), let us denote:

$$N_{i,j} = \frac{2\sigma^2 f_T(1/T)}{\Delta_{i,j}^2},$$

where $\Delta_{i,j} = \mu_{i,\pi_i(k_i)} - \mu_{i,j}$. Let us notice that $N_{i,j}$ corresponds approximately to the minimum number of pulls of component (i,j) prescribed by the lower bound in Theorem 8.3.3 and denoted with $L_{i,j} = \frac{2\sigma^2 \log T}{\Delta_{i,j}^2}$. Given the definition of $f_T(1/T)$, we have that $L_{i,j}/N_{i,j} \to 1$ as $T \to +\infty$. Given the sorting function, it is clear that also:

$$N_{i,\pi_i(1)} \leqslant \cdots \leqslant N_{i,\pi_i(k_i)}.$$

Let us define:

$$\beta_i := f_T(1/T)^{-1} \min_{l,l' \in [\![k_i]\!] \,:\, N_{i,\pi_i(l)} \neq N_{i,\pi_i(l')}} \left| N_{i,\pi_i(l)} - N_{i,\pi_i(l')} \right|.$$

It is clear that if for every $i \in [\![b]\!]$ and $j \in [\![k_i]\!]$ we have we have $|\widehat{N}_{i,j} - N_{i,j}| \leqslant \beta_i f_T(1/T)/4$, then, for any sorting function $\widehat{\pi}_i$ of the estimated quantities $\underline{N}_{i,j}$, there exist a sorting function π_i of the true quantities $N_{i,j}$ such that $\widehat{\pi}_i = \pi_i$.

Let us define for every $i \in [\![d]\!]$ and $j \in [\![k_i]\!]$:

$$M_{i,j} := \sum_{l=1}^{j} N_{i,\pi_i(l)}.$$

We define now a sorting function $\pi : [\![k]\!] \to \bigcup_{i \in [\![d]\!]} (\{i\} \times [\![k_i]\!])$ as any bijection such that:

$$M_{\pi(1)} \leqslant \cdots \leqslant M_{\pi(k)},$$

and convene (with a little abuse of notation) that $M_{\pi(0)} = 0$. It is clear that $M_{\pi(k)} = M_{\pi(k-1)} = \cdots = M_{\pi(k-d+1)} = T$. Let $l \in [\![k]\!]$, we define the *active action* as:

$$\boldsymbol{\alpha}(l) := (j_1, \ldots, j_d),$$

where:

$$j_i \text{ s.t. } \pi(l') = (i, j_i) \text{ and } l' = \min\{l'' \geqslant l \text{ and } \pi(l'') = (i, \cdot)\} \text{ with } i \in [\![d]\!].$$

We can now rewrite the regret with this notation:

$$\sum_{\mathbf{a} \neq \mathbf{a}^*} N_{\mathbf{a}} \Delta_{\mathbf{a}} = \sum_{l=1}^{k-d} \left(M_{\pi(l)} - M_{\pi(l-1)} \right) \Delta_{\boldsymbol{\alpha}(l)},$$

having observed that for the $k-d+1$ terms we play the optimal action and the successive ones are zero. Furthermore, given the relation between $L_{i,j}$ and $N_{i,j}$, we have that:

$$\frac{\sum_{\mathbf{a} \neq \mathbf{a}^*} N_{\mathbf{a}}}{f_T(1/T)} = \underline{C} \qquad \text{and} \qquad \limsup_{T \to +\infty} \frac{\sum_{\mathbf{a} \neq \mathbf{a}^*} N_{\mathbf{a}}}{\log T} = \underline{C}.$$

Let us now define:

$$\beta := f_T(1/T)^{-1} \min_{l,l' \in [\![k]\!] \,:\, M_{\pi(l)} \neq M_{\pi(l')}} \left| M_{\pi(l)} - M_{\pi(l')} \right|.$$

It is clear that if for every $i \in [\![b]\!]$ and $j \in [\![k_i]\!]$ we have $|\widehat{M}_{i,j} - M_{i,j}| \leqslant \beta f_T(1/T)/4$, for every sorting function $\widehat{\pi}$ of the estimated quantities $\widehat{M}_{i,j}$, there exist a sorting function π of the true quantities $M_{i,j}$ such that $\widehat{\pi} = \pi$. If this is the case, then, the *active action* $\widehat{\boldsymbol{\alpha}}(l)$ induced by $\widehat{\pi}$ must be the same as $\boldsymbol{\alpha}(l)$ since the active action depends on the sorting function only. We now show that we can always guarantee $|\widehat{N}_{i,j} - N_{i,j}| \leqslant (\beta_i f_T(1/T))/4$ and $|\widehat{M}_{i,j} - M_{i,j}| \leqslant (\beta f_T(1/T))/4$ for sufficiently large T. First of all, let us ensure that we identify the optimal component for every $i \in [\![d]\!]$. This is guaranteed whenever for every $j \in [\![k_i]\!]$ we have:

$$|\widehat{\mu}_{i,j} - \mu_{i,j}| \leqslant \epsilon_{i,j}(T_{\text{warm-up}}, 1/\log T) \leqslant \epsilon_T \leqslant \Delta_{\text{min}}/4,$$

where $\Delta_{\text{min}} = \min_{i \in [\![d]\!]} \min_{j \in [\![k_i]\!] \setminus \{\pi_i(k_i)\}} \mu_{i,\pi_i(k_i)} - \mu_{i,j}$. The inequality is satisfied for sufficiently large T since:

$$\epsilon_T = \sqrt{\frac{2\sigma^2 f_T(1/\log T)}{\lceil \sqrt{\log T} \rceil}} = \mathcal{O}\left(\sqrt{\frac{\sigma^2 \log\log T}{\sqrt{\log T}}} \right) \to 0 \quad \text{as} \quad T \to +\infty.$$

Under this condition, we have that $\pi_i(k_i) = \widehat{\pi}_i(k_i)$ and, consequently:

$$\widehat{\Delta}_{i,j} = \widehat{\mu}_{i,\pi(k_i)} - \widehat{\mu}_{i,j} \qquad \text{and} \qquad \Delta_{i,j} = \mu_{i,\pi(k_i)} - \mu_{i,j}.$$

Thus, under event $\mathcal{E}_1^\complement$, we have $|\widehat{\Delta}_{i,j} - \Delta_{i,j}| \leqslant 2\epsilon_T$. Let us now consider $i \in [\![k]\!]$ and $j \in [\![k_i]\!] \backslash \{\pi_i(k_i)\}$, we have:

$$\begin{aligned}\left|\widehat{N}_{i,j} - N_{i,j}\right| &= \left|\frac{2\sigma^2 f_T(1/T)}{\widehat{\Delta}_{i,j}^2} - \frac{2\sigma^2 f_T(1/T)}{\Delta_{i,j}^2}\right| \\ &= 2\sigma^2 f_T(1/T)\frac{(\Delta_{i,j} + \widehat{\Delta}_{i,j})|\Delta_{i,j} - \widehat{\Delta}_{i,j}|}{\Delta_{i,j}^2 \widehat{\Delta}_{i,j}^2} \\ &\leqslant 8\sigma^2 f_T(1/T)\frac{(2\Delta_{\max} + \Delta_{\min}/2)}{\Delta_{\min}^4}\epsilon_T,\end{aligned}$$

where $\Delta_{\max} = \max_{i \in [\![d]\!]} \max_{j,j' \in [\![k_i]\!]} |\mu_{i,j} - \mu_{i,j'}|$ and having observed that $\widehat{\Delta}_{i,j} \geqslant \Delta_{i,j} - 2\epsilon_T \geqslant \Delta_{\min} - \Delta_{\min}/2 = \Delta_{\min}/2$ and $\widehat{\Delta}_{i,j} \leqslant \Delta_{i,j} + 2\epsilon_T \leqslant \Delta_{\max} + \Delta_{\min}/2 = \Delta_{\min}/2$. Thus, the difference can go below $\beta_i f_T(1/T)$ for sufficiently large T. Let us now move to the $M_{i,j}$ variables. For sufficiently large T such that the sorting function π_i coincide with their estimated counterparts $\widehat{\pi}_i$, we have that for $i \in [\![d]\!]$ and $j \in [\![k_i]\!]$:

$$\begin{aligned}\left|M_{i,j} - \widehat{M}_{i,j}\right| &= \left|\sum_{l=1}^{j} N_{i,\pi_i(l)} - \sum_{l=1}^{j} \widehat{N}_{i,\widehat{\pi}_i(l)}\right| &\text{(D.44)} \\ &\leqslant \sum_{l=1}^{j} \left|\underline{N}_{i,\pi_i(l)} - \widehat{N}_{i,\pi_i(l)}\right| &\text{(D.45)} \\ &\leqslant 8\sigma^2 j f_T(1/T)\frac{(2\Delta_{\max} + \Delta_{\min}/2)}{\Delta_{\min}^4}\epsilon_T. &\text{(D.46)}\end{aligned}$$

Similarly, as before, we can conclude that this difference can be made smaller than β for sufficiently large T, and, consequently, make the estimated sorting function $\widehat{\pi}$ equal the true counterpart π.

Under these conditions, we can bound the cumulative regret under $\mathcal{E}_1^\complement$:

$$\begin{aligned}\sum_{t \in success} \Delta_{\mathbf{a}(t)} &= \sum_{\mathbf{a} \neq \mathbf{a}^*} \widehat{N}_{\mathbf{a}} \Delta_{\mathbf{a}} \\ &= \sum_{l=1}^{k-d} \left(\widehat{M}_{\widehat{\pi}(l)} - \widehat{M}_{\widehat{\pi}(l-1)}\right) \Delta_{\widehat{\boldsymbol{\alpha}}(l)} \\ &= \sum_{l=1}^{k-d} \left(\widehat{M}_{\pi(l)} - \widehat{M}_{\pi(l-1)}\right) \Delta_{\boldsymbol{\alpha}(l)} \\ &= \sum_{l=1}^{k-d} \left(\widehat{M}_{\pi(l)} - M_{\pi(l)} + M_{\pi(l-1)} - \widehat{M}_{\pi(l-1)}\right) \Delta_{\boldsymbol{\alpha}(l)} +\end{aligned}$$

$$
\begin{aligned}
&+\sum_{l=1}^{k-d}\left(M_{\pi(l)}-M_{\pi(l-1)}\right)\Delta_{\boldsymbol{\alpha}(l)}\\
&\leqslant 2\Delta_{\mathbf{max}}\sum_{l=1}^{k-d}\left|\widehat{M}_{\pi(l)}-M_{\pi(l)}\right|+\underline{C}f_T(1/T)\\
&\leqslant 8\sigma^2(k-d)\max_{i\in[\![d]\!]}k_i f_T(1/T)\frac{(2\Delta_{\max}+\Delta_{\min}/2)}{\Delta_{\min}^4}\epsilon_T\\
&\quad+\underline{C}f_T(1/T)\\
&=\mathcal{O}(\epsilon_T f_T(1/T))+\underline{C}f_T(1/T),
\end{aligned}
$$

where we used Equation (D.46). Thus, recalling that $\epsilon_T \to 0$ for $T \to +\infty$, we have:

$$
\limsup_{T\to+\infty}\frac{\mathbb{E}\left[\mathbf{1}\{\mathcal{E}_1^{\complement}\}\sum_{t\in success}\Delta_{\mathbf{a}(t)}\right]}{\log T}=\underline{C}.
$$

Consequently, its contribution to the asymptotic regret is exactly $\underline{C}$.
Part 3.2: Regret under $\mathcal{E}_1 \wedge \mathcal{E}_2^{\complement}$ In this case, we have to prove that the regret remains logarithmic. We consider two cases:
Case 1 We perform the analysis in the first case under the following conditions:

$$
\forall i\in[\![d]\!]:\quad \pi_i(k_i)=\widehat{\pi}_i(k_i)\quad\text{and}\quad \forall j\in[\![k_i]\!]\backslash\{\pi_i(k_i)\}:\quad \widehat{\Delta}_{i,j}\geqslant\Delta_{\min}/4. \tag{D.47}
$$

In such a case, it is simple to show that the regret is at most logarithmic. Indeed, being the optimal arm correctly identified ($\pi_i(k_i) = \widehat{\pi}_i(k_i)$) we have:

$$
\begin{aligned}
\sum_{\mathbf{a}\neq\mathbf{a}^*}\widehat{N}_{\mathbf{a}}\Delta_{\mathbf{a}}&\leqslant 2\Delta_{\mathbf{max}}\sum_{l=1}^{k-d}\widehat{M}_{\widehat{\pi}(l)}\\
&\leqslant 2\Delta_{\mathbf{max}}\sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]\backslash\{\pi_i(k_i)\}}\widehat{N}_{i,\pi_i(j)}\\
&\leqslant 4\sigma^2 f_T(1/T)\Delta_{\mathbf{max}}\sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]\backslash\{\pi_i(k_i)\}}\widehat{\Delta}_{i,\pi_i(j)}^{-2}\\
&\leqslant 64k\sigma^2 f_T(1/T)\Delta_{\mathbf{max}}\Delta_{\min}^{-2}=\mathcal{O}(\log T),
\end{aligned}
$$

where we observed that since the optimal arm is correctly identified, the following inequality holds: $\sum_{l=1}^{k-d}\widehat{M}_{\widehat{\pi}(l)}\leqslant\sum_{i\in[\![d]\!]}\sum_{j\in[\![k_i]\!]\backslash\{\pi_i(k_i)\}}\widehat{N}_{i,\pi_i(j)}$.

Case 2 If the condition in Equation (D.47) is violated, we show that the success phase stops after a logarithmic number of rounds. Consider the smallest round $t_{i,j}$ in which for a given $i \in [\![k]\!]$ and $j \in [\![k_i]\!] \backslash \{\widehat{\pi}_i(k_i)\}$, it holds that:

$$N_{i,j}(t_{i,j}) \geqslant \min\left\{\frac{2\sigma^2 f_T(1/T)}{\widehat{\Delta}_{i,j}^2}, \frac{128\sigma^2 f_T(1/T)}{\Delta_{\min}^2}\right\}. \tag{D.48}$$

Since the `F-Track` algorithm in the success phase proceeds with the round robin of at most k arms, we have that:

$$\begin{aligned} t_{i,j} &\leqslant k \min\left\{\frac{2\sigma^2 f_T(1/T)}{\widehat{\Delta}_{i,j}^2}, \frac{128\sigma^2 f_T(1/T)}{\Delta_{\min}^2}\right\} && \text{(D.49)} \\ &\leqslant \frac{128 k\sigma^2 f_T(1/T)}{\Delta_{\min}^2} && \text{(D.50)} \\ &=: t^* = \mathcal{O}(\log T). && \text{(D.51)} \end{aligned}$$

Now, we consider two sub-cases.
Case 2.1 In the first sub-case, we deal with the case in which some optimal components are not correctly identified:

$$\exists i \in [\![d]\!]: \quad \pi_i(k_i) \neq \widehat{\pi}_i(k_i)$$

In such a case, at most at round t^*, we have that:

$$\begin{aligned} \widehat{\mu}_{i,\pi_i(k_i)}(t) &\geqslant \mu_{i,\pi_i(k_i)}(t) - \sqrt{\frac{2\sigma^2 f_T(1/T)}{N_{i,\pi_i(k_i)}(t)}} && \text{(D.52)} \\ &\geqslant \mu_{i,\pi_i(k_i)}(t) - \max\left\{\widehat{\Delta}_{i,\pi_i(k_i)}, \Delta_{\min}/8\right\} && \text{(D.53)} \\ &\geqslant \mu_{i,\pi_i(k_i)}(t) - \widehat{\Delta}_{i,\pi_i(k_i)} - \Delta_{\min}/8 && \text{(D.54)} \\ &\geqslant \mu_{i,\widehat{\pi}_i(k_i)}(t) + \Delta_{i,\widehat{\pi}_i(k_i)} - \Delta_{\min}/8 - \widehat{\Delta}_{i,\pi_i(k_i)} && \text{(D.55)} \\ &\geqslant \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(t) - \sqrt{\frac{2\sigma^2 f_T(1/T)}{N_{i,\widehat{\pi}_i(k_i)}(t)}} + \Delta_{i,\widehat{\pi}_i(k_i)} - \Delta_{\min}/8 - \widehat{\Delta}_{i,\pi_i(k_i)} && \text{(D.56)} \\ &\geqslant \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(t) - \max\{0, \Delta_{\min}/8\} + \Delta_{i,\widehat{\pi}_i(k_i)} - \Delta_{\min}/8 - \widehat{\Delta}_{i,\pi_i(k_i)} && \text{(D.57)} \\ &\geqslant \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(t) - 3/4\Delta_{\min} + \widehat{\mu}_{i,\pi_i(k_i)}(T_{\text{warm-up}}) - \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(T_{\text{warm-up}}). && \text{(D.58)} \end{aligned}$$

where line (D.52) follows from the fact that event $\mathcal{E}_2$ does not hold, line (D.53) follows from Equation (D.48) with $j = \pi_i(k_i)$, line (D.54) is obtained with $\max a, b \leqslant a + b$ for $a, b \geqslant 0$, line (D.55) is obtained from the definition of $\Delta_{i,\widehat{\pi}_i(k_i)}$, line (D.56) follows from the fact that event $\mathcal{E}_2$ does not hold, line (D.57) follows from Equation (D.48) with $j = \widehat{\pi}_i(k_i)$ (whose estimated $\widehat{\Delta}_{i,\widehat{\pi}_i(k_i)} = 0$, and line (D.58) is obtained from the definition of $\widehat{\Delta}_{i,\pi_i(k_i)}$ and from $\Delta_{i,\widehat{\pi}_i(k_i)} \geqslant \Delta_{\min}$.
This implies that at this round:

$$\begin{aligned}\widehat{\mu}_{i,\pi_i(k_i)}(t) - \widehat{\mu}_{i,\pi_i(k_i)}(T_{\text{warm-up}}) + \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(T_{\text{warm-up}}) - \widehat{\mu}_{i,\widehat{\pi}_i(k_i)}(t) &\geqslant 3/4\Delta_{\min} \\ &\geqslant 4\epsilon_T,\end{aligned}$$

where the latter holds for sufficiently large T. Thus, we have that the success phase stops after at most t^* rounds, leading to a regret of:

$$\sum_{t \in \mathit{success}} \Delta_{\mathbf{a}(t)} \leqslant \Delta_{\mathbf{max}} \frac{32k\sigma^2 f_T(1/T)}{\Delta_{\min}^2} = \mathcal{O}(\log T).$$

Case 2.2 In the first sub-case, we deal with the case holding under the condition:

$$\forall i \in [\![d]\!] : \quad \pi_i(k_i) = \widehat{\pi}_i(k_i),$$

and:

$$\exists i \in [\![d]\!] : \quad \exists j \in [\![k_i]\!] \backslash \{\pi_i(k_i)\} : \quad \widehat{\Delta}_{i,j} < \Delta_{\min}/4.$$

At round t^*, for the (i, j) fulfilling the second part of the condition:

$$\begin{aligned}&\widehat{\mu}_{i,\pi_i(k_i)}(t) - \widehat{\mu}_{i,\pi_i(k_i)}(T_{\text{warm-up}}) + \widehat{\mu}_{i,j}(T_{\text{warm-up}}) - \widehat{\mu}_{i,j}(t) \\ &\qquad \geqslant \widehat{\mu}_{i,\pi_i(k_i)}(t) - \widehat{\mu}_{i,j}(t) - \widehat{\Delta}_{i,j} \\ &\qquad \geqslant \mu_{i,\pi_i(k_i)}(t) - \sqrt{\frac{2\sigma^2 f_T(1/T)}{N_{i,\pi_i(k_i)}(t)}} - \mu_{i,j}(t) - \sqrt{\frac{2\sigma^2 f_T(1/T)}{N_{i,j}(t)}} - \widehat{\Delta}_{i,j} \\ &\qquad \geqslant -\max\{0, \Delta_{\min}/8\} - \max\{\widehat{\Delta}_{i,j}, \Delta_{\min}/8\} + \Delta_{i,j} - \widehat{\Delta}_{i,j} \\ &\qquad \geqslant \Delta_{\min}/4,\end{aligned}$$

having exploited $\widehat{\Delta}_{i,j} \leqslant \Delta_{\min}/4$ and $\Delta_{i,j} \geqslant \Delta_{\min}$. Thus, for sufficiently large T, we have that $4\epsilon_T \leqslant \Delta_{\min}/4$ and, consequently, the success phase ends.
Part 3.3: Regret under $\mathcal{E}_2$ We conclude by bounding the regret under event $\mathcal{E}_2$, In this case, we proceed with the following trivial bound, recalling that

$\Pr(\mathcal{E}_2) \leqslant 1/T$.

$$\mathbb{E}\left[\mathbf{1}\{\mathcal{E}_2\} \sum_{t \in success} \Delta_{\mathbf{a}(t)}\right] \leqslant \Delta_{\mathbf{max}} T\, \mathbb{P}(\mathcal{E}_2) \leqslant \Delta_{\mathbf{max}} = \mathcal{O}(1).$$

Consequently, its contribution to the asymptotic regret is negligible. □

D.1.2 Technical Lemmas

Lemma D.1.1. *Let $T \in \mathbb{N}$, $\epsilon > 0$. Let $X_1, \dots, X_T$ be a martingale difference sequence adapted to the filtration $\mathcal{F}_0, \mathcal{F}_1, \dots$, such that for every $t \in [\![T]\!]$, it holds that $\mathbb{E}[e^{\lambda X_t}] \leqslant e^{(\sigma^2\lambda^2)/2}$ a.s. for every $\lambda \in \mathbb{R}$. Then, for every $\delta \in (0,1)$ it holds that:*

$$\mathbb{P}\left(\exists t \in [\![T]\!] : \sum_{s=1}^{t} X_s \geqslant \right.$$
$$\left.\sqrt{2\left(1+(\log T)^{-1}\right)\max\{\epsilon, t\sigma^2\}\left(\log\left(1+\left\lceil\frac{\log(T\sigma^2/\epsilon)}{\log(1+(\log T)^{-1})}\right\rceil\right)+\log\left(\frac{1}{\delta}\right)\right)}\right)$$
$$\leqslant \delta.$$

Furthermore, for sufficiently large T, it holds that:

$$\mathbb{P}\left(\exists t \in [\![T]\!] : \sum_{s=1}^{t} X_s \geqslant \sqrt{2\sigma^2 t f_T(\delta)}\right) \leqslant \delta,$$

where:

$$f_T(\delta) := \left(1 + \frac{1}{\log T}\right)\left(c \log\log T + \log\left(\frac{1}{\delta}\right)\right),$$

and $c > 0$ is a universal constant.

Proof. The first statement is obtained from Lemma 14 of (Lattimore and Szepesvari, 2017) considering that the inequality employed in Equation (19) of that proof applies for σ^2-subgaussian random variables and not for Gaussian variables only. The second statement is obtained by setting $\epsilon = \sigma^2$ and bounding $\frac{1}{\log(1+(\log T)^{-1})} \leqslant \log T$ and $\log(1 + \lceil(\log T)^2\rceil) \leqslant c \log\log T$ for some universal constant c (≈ 2). □

Lemma D.1.2. *Let $\overline{x} \in [0,1)$, $d \in \mathbb{N}$, then if $x_i \in [0, \overline{x})$, $\forall i \in [\![d]\!]$, it holds:*

$$1 - \prod_{i \in [\![d]\!]}(1 - x_i) \geqslant (1 - \overline{x})^{d-1} \sum_{i \in [\![d]\!]} x_i.$$

Proof. We prove this statement by induction.
First, we can observe how for $d = 1$ this result trivially holds:

$$1 - (1 - x_1) = x_1.$$

We can now make the inductive step on d:

$$\begin{aligned}
1 - \prod_{i \in [\![d]\!]} (1 - x_i) &= 1 - (1 - x_d) \prod_{i \in [\![d-1]\!]} (1 - x_i) \\
&= 1 - (1 - x_d) \prod_{i \in [\![d-1]\!]} (1 - x_i) \pm x_d \\
&= (1 - x_d) \left(1 - \prod_{i \in [\![d-1]\!]} (1 - x_i) \right) + x_d \qquad \text{(D.59)} \\
&\geqslant (1 - x_d) \left((1 - \overline{x})^{d-2} \sum_{i \in [\![d-1]\!]} x_i \right) + x_d \\
&\geqslant (1 - \overline{x})^{d-1} \sum_{i \in [\![d]\!]} x_i,
\end{aligned}$$

where line (D.59) is the inductive step on d. □

Lemma D.1.3. *In a FRB, considering $\mu_{\mathbf{a}*} = 1$, if $\Delta_{i,j} \leqslant \overline{\Delta} = 1 - \frac{1}{2^{1/(d-1)}}, \forall i \in [\![d]\!], j \in [\![k_i]\!]$, the regret can be bounded as:*

$$R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}}) = \sum_{t \in [\![T]\!]} \left(1 - \prod_{i \in [\![d]\!]} \left(1 - \Delta_{i,a_i(t)}\right) \right) \geqslant \frac{1}{2} \sum_{t \in [\![T]\!]} \sum_{i \in [\![d]\!]} \Delta_{i,a_i(t)}.$$

Proof. We prove this statement by looking at a single time t. We can rewrite Lemma D.1.2 as:

$$1 - \prod_{i \in [\![d]\!]} (1 - \Delta_{i,a_i(t)}) \geqslant (1 - \overline{\Delta})^{d-1} \sum_{i \in [\![d]\!]} \Delta_{i,a_i(t)},$$

if $\Delta_{i,j} \leqslant \overline{\Delta} \in [0, 1), \forall i \in [\![d]\!], j \in [\![k_i]\!]$.
We make a choice we want to transform this result in order to have:

$$1 - \prod_{i \in [\![d]\!]} (1 - \Delta_{i,a_i(t)}) \geqslant \frac{1}{2} \sum_{i \in [\![d]\!]} \Delta_{i,a_i(t)}.$$

This can be done by imposing:

$$\frac{1}{2} \leqslant (1 - \overline{\Delta})^{d-1}$$

$$\frac{1}{2^{1/(d-1)}} \leqslant (1-\overline{\Delta})$$
$$\overline{\Delta} \leqslant 1 - \frac{1}{2^{1/(d-1)}}.$$

□

Lemma D.1.4 (Wang et al. 2021c)**.** *Suppose m, B are positive integers and $m \geqslant 2$; there are $m+1$ probability distributions $\mathbb{P}_0, \mathbb{P}_1, \dots \mathbb{P}_m$, and m random variables $N_1, \dots, N_m$, such that: (i) Under any of the P_i's, $N_1, \dots, N_m$ are non-negative and $\sum_{i \in [\![m]\!]} N_i \leqslant B$ with probability 1; (ii) $\forall i \in [\![m]\!]$, $d_{\text{TV}} \leqslant \frac{1}{4}\sqrt{\frac{m}{B}\mathbb{E}_0[N_i]}$. Then:*

$$\frac{1}{m}\sum_{i \in [\![m]\!]} \mathbb{E}_i[B - N_i] \geqslant \frac{B}{4}.$$

Proof. For the proof of this Lemma, we refer the reader to Lemma 24 of (Wang et al., 2021c). □

D.2 Additional Theorems and Lemmas

In this section, we provide additional Theorems and Lemmas useful in the discussion of the work.

Lemma D.2.1. *The product $X_1 X_2 \cdots X_n$ of $n \geqslant 3$ independent random variables σ^2-subgaussian is not subgaussian anymore.*

Proof. The proof follows the one proposed by (Pinelis, 2021).
The proof of this statement can be done by verifying that the moment-generating function of the product of n independent Gaussian distributions with unit variance ($X_i \sim \mathcal{N}(0,1), \ \forall i \in [\![n]\!]$) is unbounded:

$$\mathbb{E}\left[\exp\left(c\prod_{i \in [\![n]\!]} X_i\right)\right] = \infty, \qquad \forall c > 0.$$

Given our random variables $X_1, X_2, \dots, X_n$, let us call X the vector composed of our random variables ($X := (X_1, X_2, \dots, X_n)$) and let the vector $(U_1, U_2, \dots U_n)$ be a uniformly distributed unit random vector. For some real $C_n > 0$:

$$\mathbb{E}\left[\exp\left(c\prod_{i \in [\![n]\!]} X_i\right)\right]$$

$$\geqslant \mathbb{E}\left[\exp\left(c\prod_{i\in[\![n]\!]} X_i\right)\mathbb{1}\left\{X_i > \frac{||X||_2}{2\sqrt{n}}, \forall i \in [\![n]\!]\right\}\right] \tag{D.60}$$

$$= C_n \int_0^\infty \exp\left(c\underbrace{\frac{1}{(2\sqrt{n})^n}}_{\text{(A)}} r^n\right)\underbrace{r^{n-1}\exp\left(-\frac{r^2}{2}\right)}_{\text{(B)}}\mathrm{d}r\cdot$$

$$\cdot\underbrace{\mathbb{P}\left(U_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]\right)}_{\text{(C)}} \tag{D.61}$$

$$= C_n\frac{(2\sqrt{n})^n}{cn}\int_0^\infty \underbrace{\exp\left(c\frac{1}{(2\sqrt{n})^n}r^n\right)\frac{cn}{(2\sqrt{n})^n}r^{n-1}}_{g'(r)}\underbrace{\exp\left(-\frac{r^2}{2}\right)}_{f(r)}\,\mathrm{d}r\cdot$$

$$\cdot\mathbb{P}\left(U_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]\right)$$

$$= C_n\frac{(2\sqrt{n})^n}{cn}\cdot$$

$$\cdot\left(\left[\exp\left(c\frac{1}{(2\sqrt{n})^n}r^n\right)\exp\left(-\frac{r^2}{2}\right)\right]_0^\infty + \int_0^\infty \exp\left(\underbrace{c\frac{1}{(2\sqrt{n})^n}}_{(D)}r^n - \frac{r^2}{2}\right) r\,\mathrm{d}r\right)$$

$$\cdot\mathbb{P}\left(U_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]\right) \tag{D.62}$$

$$\geqslant C_n\frac{(2\sqrt{n})^n}{cn}\left([\infty - 0] + \int_0^\infty \exp\left(-\frac{r^2}{2}\right) r\,\mathrm{d}r\right)\cdot$$

$$\cdot\mathbb{P}\left(U_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]\right) \tag{D.63}$$

$$= C_n\frac{(2\sqrt{n})^n}{cn}\left([\infty - 0] - \left[\exp\left(-\frac{r^2}{2}\right)\right]_0^\infty\right)\cdot$$

$$\cdot\mathbb{P}\left(U_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]\right)$$

$$\overset{\substack{C_n>0\\ n\geqslant 3\\ c>0}}{=} \infty.$$

The inequality in Equation (D.60) follows from the fact that the event inside the indicator function happens with a probability $\leqslant 1$. Equation (D.61) is a rewriting of the previous line under the assumption that the indicator function evaluates to 1. We can rewrite the expected value as an integral over the positive real numbers since, according to the indicator function, every random variable X_i must be greater than $\frac{||X||_2}{2\sqrt{n}}$, which is a positive quantity.

Term (A) is a substitution of $\prod_{i\in[\![n]\!]} X_i$ with $\frac{r}{2\sqrt{n}}$ repeated n times, which comes from the indicator function. r is the integration variable and represents the Euclidean norm of vector X.

Term (B) represents the probability density of the Euclidean norm of a Gaussian vector $X \sim \mathcal{N}(0, \mathbf{I}_n)$.

Finally, term (C) represents the probability of the indicator function evaluating to 1. Considering the vector Y whose elements are $Y_i = X_i/||X||_2$, then $||Y||_2 = 1$. The probability that $Y_i > \frac{1}{2\sqrt{n}}, \forall i \in [\![n]\!]$ can be thought of as the probability that the point defined by Y in the n-dimensional space is located on the surface of the n-dimensional hyper-sphere of radius 1 in the region induced by the condition $Y_i > \frac{1}{2\sqrt{n}}$.

Equation (D.62) is an integration by parts of the two functions $f(r)$ and $g'(r)$ identified in the line above.

Equation (D.63) holds under the assumption that $n \geqslant 3$ and $c > 0$. First, the term:

$$\left[\exp\left(c\frac{1}{(2\sqrt{n})^n}r^n\right)\exp\left(-\frac{r^2}{2}\right)\right]_0^\infty \overset{\substack{n\geqslant 3\\ c>0}}{=} \infty - 0$$

under such an assumption. Second, we can write:

$$\exp\left(c\frac{1}{(2\sqrt{n})^n}r^n - \frac{r^2}{2}\right) \geqslant \exp\left(-\frac{r^2}{2}\right)$$
$$\Rightarrow \int_0^\infty \exp\left(c\frac{1}{(2\sqrt{n})^n}r^n - \frac{r^2}{2}\right)\,\mathrm{d}r \geqslant \int_0^\infty \exp\left(-\frac{r^2}{2}\right)\,\mathrm{d}r$$

The final result then holds under the further assumption that $C_n > 0$. □

Lemma D.2.2 (Variance of the product of independent random variables)**.** *Let X_1, X_2, $\ldots X_n$ independent random variables. The variance of their product is:*

$$\mathbb{V}\mathrm{ar}[X_1X_2\cdots X_n] = \prod_{i\in[\![n]\!]}\left(\mathbb{V}\mathrm{ar}[X_i] + (\mathbb{E}[X_i])^2\right) - \prod_{i\in[\![n]\!]}(\mathbb{E}[X_i])^2$$

Proof.

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[X_1 X_2 \cdots X_n] \\
&\qquad = \mathbb{E}[(X_1 X_2 \cdots X_n)^2] - (\mathbb{E}[X_1 X_2 \cdots X_n])^2 \\
&\qquad = \mathbb{E}[X_1^2 X_2^2 \cdots X_n^2] - (\mathbb{E}[X_1])^2 (\mathbb{E}[X_2])^2 \cdots (\mathbb{E}[X_n])^2 \\
&\qquad = \mathbb{E}[X_1^2]\,\mathbb{E}[X_2^2] \cdots \mathbb{E}[X_n^2] - (\mathbb{E}[X_1])^2 (\mathbb{E}[X_2])^2 \cdots (\mathbb{E}[X_n])^2 \\
&\qquad = \prod_{i \in [\![n]\!]} \left(\mathbb{V}\mathrm{ar}[X_i] + (\mathbb{E}[X_i])^2\right) - \prod_{i \in [\![n]\!]} (\mathbb{E}[X_i])^2
\end{aligned}
$$

□

Lemma D.2.3. *Let $X_1,\ X_2,\ \dots, X_n$ independent subgaussian random variables with expected value $\mu_i \in [0,1]$ and subgaussianity parameter $\sigma_i \in [0,+\infty)$. The variance of the product $X_1 X_2 \cdots X_n$ is bounded by:*

$$
\prod_{i \in [\![d]\!]} \sigma_i^2 \leqslant \mathbb{V}\mathrm{ar}[X_1 X_2 \cdots X_n] \leqslant \prod_{i \in [\![n]\!]} \left(1 + \sigma_i^2\right) - 1
$$

Proof. Now, we want to find the worst combination of $\mu_i, i \in [\![n]\!]$, i.e., the combination of expected values which maximizes the variance of the product of such random variables. To do so, we can consider a single $\bar{i} \in [\![n]\!]$, and look at the behavior of the first derivative when we change $\mu_{\bar{i}} \in [0,1]$. We recall from Lemma D.2.2 that:

$$
\begin{aligned}
&\mathbb{V}\mathrm{ar}[X_1 X_2 \cdots X_n] \\
&\qquad = \prod_{i \in [\![n]\!]} \left(\mathbb{V}\mathrm{ar}[X_i] + (\mathbb{E}[X_i])^2\right) - \prod_{i \in [\![n]\!]} (\mathbb{E}[X_i])^2 \\
&\qquad = \prod_{i \in [\![n]\!]} \left(\sigma_i^2 + \mu_i^2\right) - \prod_{i \in [\![n]\!]} \mu_i^2 \\
&\qquad = \left(\sigma_{\bar{i}}^2 + \mu_{\bar{i}}^2\right) \prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \left(\sigma_i^2 + \mu_i^2\right) - \mu_{\bar{i}}^2 \prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \mu_i^2, \qquad \text{(D.64)} \\
&\qquad = \mu_{\bar{i}}^2 \prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \left(\sigma_i^2 + \mu_i^2\right) - \mu_{\bar{i}}^2 \prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \mu_i^2 + \sigma_{\bar{i}}^2 \prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \left(\sigma_i^2 + \mu_i^2\right) \qquad \text{(D.65)} \\
&\qquad = \mu_{\bar{i}}^2 \left(\underbrace{\prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \left(\sigma_i^2 + \mu_i^2\right)}_{\mathtt{A}} - \underbrace{\prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \mu_i^2}_{\mathtt{B}} \right) + \sigma_{\bar{i}}^2 \underbrace{\prod_{i \in [\![n]\!] \setminus \{\bar{i}\}} \left(\sigma_i^2 + \mu_i^2\right)}_{\mathtt{C}} \qquad \text{(D.66)}
\end{aligned}
$$

where lines (D.64), (D.65) and (D.66) are no other than an algebraic step to make explicit in the product the dependence on $\mu_{\bar{i}}$. Now we want to look at the worst case scenario for the variance, i.e., the value of $\mu_{\bar{i}}$ that maximize it.

Recalling the constraints on μ_i which is assumed to be bounded in $[0, 1]$ and σ_i^2 that is defined over $[0, +\infty]$, it trivial to see that term $\mathtt{A}$ is predominant over term $\mathtt{B}$ and so the worst case for element $\bar{i}$ is to consider $\mu_{\bar{i}} = 1$, no matter the other values of $\mu_i, i \in [\![n]\!]\backslash\{\bar{i}\}$. The term $\mathtt{C}$ is not relevant as $\mu_{\bar{i}}$ does not appear. This reasoning applies for all the possible values of $\bar{i} \in [\![n]\!]$, and so the worst case variance is when all the μ_i are equal to 1, for all the components $i \in [\![n]\!]$.

Given that, the variance of the product of independent random variables with expected values in $\mu_i \in [0, 1]$ and variance σ_i^2 can be bounded as:

$$\mathbb{V}\mathrm{ar}[X_1 X_2 \cdots X_n] \leqslant \prod_{i \in [\![n]\!]} \left(1 + \sigma_i^2\right) - 1.$$

A symmetric reasoning leads to the lower bound.

This concludes the proof. □

APPENDIX $\mathcal{E}$

System Identification

This appendix presents a solution to perform system identification in the standard LTI system starting from a singler trajectory. Given a system defined as:

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t, \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{z}_t, \end{aligned}$$

the goal is to identify matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$, in the case in which the state $\mathbf{x}_t$ cannot be observed.

E.1 Proposed Solution

In order to identify such matrices, we adopt a variant of the Ho-Kalman (Ho and Kalman, 1966) algorithm. We start from the identification method proposed by Lale et al. (2020a, Section 3), where authors consider a system of the type (strictly proper):

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t, \\ \widetilde{\mathbf{y}}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{z}_t. \end{aligned} \tag{E.1}$$

Our target setting can be seen as (not strictly proper):

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t,\\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{z}_t,\end{aligned} \tag{E.2}$$

with $\mathbf{x}_t, \boldsymbol{\epsilon}_t \in \mathbb{R}^n$, $\mathbf{u}_t \in \mathbb{R}^p$, and $\mathbf{y}_t, \mathbf{z}_t \in \mathbb{R}^m$. The noise over state transition model $\boldsymbol{\epsilon}_t$ and output $\mathbf{z}_t$ are σ^2-subgaussian random vectors. We consider in this part the standard control problem notation adopted for LTI systems. The mapping to our problem presented in Chapter 7 is straightforward by considering $\mathbf{C} = \boldsymbol{\omega}^{\mathrm{T}}$ and $\mathbf{D} = \boldsymbol{\theta}^{\mathrm{T}}$. In predictive form, the system described in Equation (E.1) is:

$$\begin{aligned}\widehat{\mathbf{x}}_{t+1} &= \bar{\mathbf{A}}\widehat{\mathbf{x}}_t + \mathbf{B}\mathbf{u}_t + \mathbf{F}\widetilde{\mathbf{y}}_t,\\ \widetilde{\mathbf{y}}_t &= \mathbf{C}\widehat{\mathbf{x}}_t + \mathbf{e}_t,\end{aligned}$$

where:

$$\begin{aligned}&\bar{\mathbf{A}} = \mathbf{A} - \mathbf{F}\mathbf{C},\\ &\mathbf{F} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\boldsymbol{\Sigma}\mathbf{C}^{\mathrm{T}} + \sigma^2\mathbf{I})^{-1},\end{aligned}$$

and $\boldsymbol{\Sigma}$ is the solution to the following DARE (Discrete Algebraic Riccati Equation):

$$\boldsymbol{\Sigma} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\boldsymbol{\Sigma}\mathbf{C}^{\mathrm{T}}(\mathbf{C}\boldsymbol{\Sigma}\mathbf{C}^{\mathrm{T}} + \sigma^2\mathbf{I})^{-1}\mathbf{C}\boldsymbol{\Sigma}\mathbf{A}^{\mathrm{T}} + \sigma^2\mathbf{I}.$$

In order to identify this LTI system, we want to detect a matrix $\widetilde{\mathcal{G}}_y$:

$$\widetilde{\mathcal{G}}_y = \begin{bmatrix}\mathbf{C}\mathbf{F} & \mathbf{C}\bar{\mathbf{A}}\mathbf{F} & \dots & \mathbf{C}\bar{\mathbf{A}}^{H-1}\mathbf{F} & \mathbf{C}\mathbf{B} & \mathbf{C}\bar{\mathbf{A}}\mathbf{B} & \dots\mathbf{C}\bar{\mathbf{A}}^{H-1}\mathbf{B}\end{bmatrix}.$$

To identify through least squares method matrix $\widetilde{\mathcal{G}}_y$, we construct for each t, a vector $\widetilde{\phi}_t$:

$$\widetilde{\phi}_t = \begin{bmatrix}\mathbf{y}_{t-1}^{\mathrm{T}} & \cdots & \mathbf{y}_{t-H}^{\mathrm{T}} & \mathbf{u}_{t-1}^{\mathrm{T}} & \cdots & \mathbf{u}_{t-H}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(m+p)H}.$$

The system output $\widetilde{\mathbf{y}}_t$ can be rewritten as:

$$\widetilde{\mathbf{y}}_t = \widetilde{\mathcal{G}}_y\widetilde{\phi}_t + \mathbf{e}_t + \mathbf{C}\mathbf{A}^H\mathbf{x}_{t-H}.$$

The output of the system under analysis (Equation E.2) is:

$$\mathbf{y}(t) = \widetilde{\mathbf{y}}_t + \mathbf{D}\mathbf{u}_t = \widetilde{\mathcal{G}}_y\widetilde{\phi}_t + \mathbf{D}\mathbf{u}_t + \mathbf{e}_t + \mathbf{C}\mathbf{A}^H\mathbf{x}_{t-H}$$

We can incorporate the contribution of $\mathbf{D}\mathbf{u}_t$ in $\widetilde{\mathcal{G}}_y$ obtaining $\mathcal{G}_y$:

$$\mathcal{G}_y = \begin{bmatrix}\mathbf{C}\mathbf{F} & \mathbf{C}\bar{\mathbf{A}}\mathbf{F} & \dots & \mathbf{C}\bar{\mathbf{A}}^{H-1}\mathbf{F} & \mathbf{D} & \mathbf{C}\mathbf{B} & \mathbf{C}\bar{\mathbf{A}}\mathbf{B} & \dots\mathbf{C}\bar{\mathbf{A}}^{H-1}\mathbf{B}\end{bmatrix}.$$

The related vector ϕ_t is:

$$\phi_t = \begin{bmatrix}\mathbf{y}_{t-1}^{\mathrm{T}} & \dots & \mathbf{y}_{t-H}^{\mathrm{T}} & \mathbf{u}_t^{\mathrm{T}} & \mathbf{u}_{t-1}^{\mathrm{T}} & \dots & \mathbf{u}_{t-H}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(m+p)H+p}.$$

The best value of $\mathcal{G}_y$ can be found through regularized least squares as in Lale et al. (2020a, Equation 10):

$$\widehat{\mathcal{G}}_y = \arg\min_X \lambda \|\mathbf{X}\|_F^2 + \sum_{\tau=t-H}^{t} \|\mathbf{y}_\tau - \mathbf{X}\phi_\tau\|_2^2,$$

where $\|\cdot\|_F$ represents the Frobenius norm. The matrix $\mathbf{D}$ can be directly retrieved from $\widehat{\mathcal{G}}_y$. In order to get matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, we remove the values related to $\mathbf{D}$ from $\widehat{\mathcal{G}}_y$ and we retrieve $\widetilde{\mathcal{G}}_y$. From now on, we refer to the algorithm proposed in Lale et al. (2020a, Appendix B).