École Doctorale Sciences pour l'Ingénieur
Inria Lille - Nord Europe
Université Lille 1

Thèse de Doctorat

présentée pour obtenir le grade de
DOCTEUR EN SCIENCES DE L'UNIVERSITÉ LILLE 1

Spécialité : **Informatique**

présentée par
Marta SOARE

Sequential Resource Allocation in Linear Stochastic Bandits

sous la direction de M. Rémi **MUNOS**
et le co-encadrement de M. Alessandro **LAZARIC**

Rapporteurs:	Mme. Michèle **SEBAG**	CNRS, Université Paris Sud
	M. Csaba **SZEPESVÁRI**	University of Alberta

Soutenue publiquement le **14 décembre 2015** devant le jury composé de :

M. Olivier **CAPPÉ**	CNRS, Télécom ParisTech	Examinateur
M. Rémi **GILLERON**	Université de Lille	Examinateur
M. Alessandro **LAZARIC**	INRIA	Co-encadrant
M. Rémi **MUNOS**	INRIA & Google DeepMind	Directeur
M. Liva **RALAIVOLA**	Université Aix-Marseille	Examinateur
Mme. Michèle **SEBAG**	CNRS, Université Paris Sud	Rapportrice
M. Csaba **SZEPESVÁRI**	University of Alberta	Rapporteur

Sequential Allocation of Resources in Linear Stochastic Bandits

This thesis is dedicated to the study of resource allocation problems in uncertain environments, where an agent can sequentially select which action to take. After each step, the environment returns a noisy observation of the value of the selected action. These observations guide the agent in adapting his resource allocation strategy towards reaching a given objective. In the most typical setting of this kind, the stochastic multi-armed bandit (MAB), it is assumed that each observation is drawn from an unknown probability distribution associated with the selected action and gives no information on the expected value of the other actions. The MAB setting has been widely studied and optimal allocation strategies were proposed to solve various objectives under the MAB assumptions. Here, we consider a variant of the MAB setting where there exists a global linear structure in the environment and by selecting an action, the agent also gathers information on the value of the other actions. Therefore, the agent needs to adapt his resource allocation strategy to exploit the structure in the environment. In particular, we study the design of sequences of actions that the agent should take to reach objectives such as: (i) identifying the best value with a fixed confidence and using a minimum number of pulls, or (ii) minimizing the prediction error on the value of each action. In addition, we investigate how the knowledge gathered by a bandit algorithm in a given environment can be transferred to improve the performance in other similar environments.

Keywords: sequential learning, bandit games, adaptive sampling, stochastic optimization

Allocation Séquentielle de Ressources dans le Modèle de Bandit Linéaire

Dans cette thèse nous étudions des problèmes d'allocation de ressources dans des environnements incertains où un agent choisit ses actions séquentiellement. Après chaque pas, l'environnement fournit une observation bruitée sur la valeur de l'action choisie et l'agent doit utiliser ces observations pour allouer ses ressources de façon optimale. Dans le cadre le plus classique, dit modèle du bandit à plusieurs bras (MAB), on fait l'hypothèse que chaque observation est tirée aléatoirement d'une distribution de probabilité associée à l'action choisie et ne fournit aucune information sur les valeurs espérées des autres actions disponibles dans l'environnement. Ce modèle a été largement étudié dans la littérature et plusieurs stratégies optimales ont été proposées, notamment pour le cas où le but de l'agent est de maximiser la somme des observations. Ici, nous considérons une version du MAB où les actions ne sont plus indépendantes, mais chaque observation peut être utilisée pour estimer les valeurs de l'ensemble des actions de l'environnement. Plus précisément, nous proposons des stratégies d'allocation de ressources qui sont efficaces et adaptées à un environnement caractérisé par une structure linéaire globale. Nous étudions notamment les séquences d'actions qui mènent à : (i) identifier la meilleure action avec une précision donnée et en utilisant un nombre minimum d'observations, ou (ii) maximiser la précision d'estimation de la valeur de chaque action. De plus, nous étudions les cas où les observations provenant d'un algorithme de bandit dans un environnement donné peuvent améliorer par la suite la performance de l'agent dans d'autres environnements similaires.

Mots-clés: apprentissage séquentiel, jeux de bandits, échantillonnage adaptatif, optimisation stochastique

Acknowledgements

I take this opportunity to express my deepest gratitude and appreciation to my supervisors, Rémi Munos and Alessandro Lazaric. I am extremely grateful for your careful and thoughtful guidance, for your patience and understanding. For your great research insights, vision, enthusiasm, and rigour that you generously shared with me. For your investment and the great help and support you offered me, fundamental to this thesis. I am also thankful for your constant encouragement and kindness. I feel extremely lucky for having had the chance to work with you and learn from you, I cannot thank you enough for it.

I would also like to thank Michèle Sebag and Csaba Szepesvári for having kindly accepted to review my PhD. I am grateful for your interest and your careful reading. Special thanks also to Liva Ralaivola, Olivier Cappé, and Rémi Gilleron, for having accepted to come at my defense.

I am also grateful to Joelle Pineau for the warm welcome during my research visit at McGill University. I also thank Joelle and Ouais Alsharif for our collaboration on transfer in bandits.

Je remercie les membres (passés ou présents) de l'équipe SequeL, toujours à l'écoute, prêts à aider et à partager de bons moments. Merci aussi à l'Inria pour le soutien logistique, ainsi qu'au projet CompLACS et au Conseil Régional du Nord-Pas de Calais pour le soutien financier.

Merci à l'équipe pédagogique du département d'Informatique de Polytech Lille pour leur accueil, leur confiance, et pour la bonne collaboration.

Dernier mot pour transmettre ma plus vive reconnaissance et mes plus chaleureux remerciements à mes parents, Adrian si Elena, à ma soeur Cristina, et à Radu.

Contents

Chapter 1
Introduction

Overview of the Studied Problem

This dissertation is dedicated to the study of the sequential resource allocation problem in an unknown environment, with a fixed and finite set of possible actions to take. While the per-step cost of choosing an action is constant over actions and time, the **value of an action** strictly depends on the information about the environment gained by choosing that action at a certain time step.

In this setting, the goal of a learner is to design an **optimal sequence of actions** (here seen as an optimal sample allocation strategy) that allows to acquire the information on the environment needed to reach a predefined optimality criterion. For this, the learner will develop **active learning** strategies, allowing to sequentially update the estimated value of each option and to choose in an optimal manner what actions to take at the next time step.

Adaptive design

When being faced with a choice, a decision maker can only rely on a limited number of observations (or evidence) on the possible choices offered to him. If the learner can control the collection of the observations giving information on the value of the choices, then he has a chance to make **more informed decisions**. Whether we refer to the process as *data collection*, *resource allocation*, or *experimental design*, it is well known that one can provably obtain better solutions by **adapting** to the previously obtained information about the unknown environment.

In the most classical setting, known as *passive data collection* or *fixed experimental design*, all decisions concerning the sampling procedure are taken prior to the observations about the environment and the learning is performed only when all resources are allocated. In contrast, sequential, adaptive procedures allow the sampling process to be more **flexible**, deciding after each new observation whether the decision maker needs more information about the environment *(Are additional samples needed?)* or on which factors of the environment the uncertainty is still too high *(Where to sample next?)*. The two inherent advantages of this flexibility in adjusting the strategy as new information about the environment is gathered are (i) the more **efficient use of resources**, thus reducing the costs of the experiment/sampling process and (ii) the potential **improvement in the accuracy of the solution**, due to the additional

knowledge about the environment, potentially obtained by controlling the information collection.

To be able to define a *decision rule* or *design*, a learner is provided with some prior information on the environment (for instance, the number of available options, the number of allowed observations, or the existence of a global structure connecting the value of the options). More importantly, the learner will be guided by an **optimality criterion**, which is the basis for the exploration of the space (as it defines what is the needed information) and which also gives the measure of performance for his decisions.

Theoretical Setting

The settings we consider in the thesis come from Bandit Theory and also take inspiration from Optimal Experimental Design, both fields providing well suited tools for managing and analyzing the trade-off between gaining information on the environment and attaining a given optimality criterion.

Departing from the traditional, widely studied multi-armed bandit (MAB) setting, where the value of each choice is independent, we consider throughout the dissertation the **Linear Stochastic Bandits** setting. In this scenario, we assume the available options (arms) are d-dimensional vectors and their values are given by the linear combination between the arms and an unknown vector of parameters θ^* characterizing the underlying linear function. This global structure of the environment is such that by observing a noisy reward from an arm, we also (indirectly) gain information on the value of the other arms. Knowing that after each arm pull (allocated sample) we obtain a noisy observation of the value of the chosen arm and that the observations are expensive (limited sampling budget), the goal of a learner is to exploit the global linear structure to design adaptive allocation strategies that only sample the **most informative arms**. More precisely, the aim of the decision-maker is to allocate samples only to the arms whose observed rewards allow to improve the estimation of the features of θ^* with accuracy levels that allow to meet a specific optimization criterion.

Given the linear stochastic setting, for the objectives we consider the exploration-exploitation trade-off will differ from the MAB setting with independent arms. Thus, we will design and analyze adaptive algorithms for several objectives, starting from the traditional cumulative regret minimization and extending the performance measure to optimality criteria coming from the Optimal Design of Experiments literature. More precisely, we design and analyze **adaptive allocation strategies** meant to minimize the accuracy of predicting the arm values.

Motivation

The stochastic linear bandit problem (LB), introduced in [Auer, 2002], is a natural extension of the typical MAB setting, mostly relevant in applications where the number of available arms is very large or observations are very expensive (limited sampling budget). In fact, the uncertainty of the environment in this setting is concentrated

in the unknown parameter $\theta^* \in \mathbb{R}^d$ characterizing the linear function. Therefore, the estimation of the value of each arms is replaced by the estimation of the d features of θ^* and the problem changes fundamentally.

More in general, the linear bandit framework extends the applications of classical multi-armed bandit by offering a formalization proper also for sequential decision making problems where the set of feasible decisions is very large (or even infinite). The possibility of indirectly learning about all possible options after each pull values is particularly relevant for settings where the emphasis is put on obtaining information on the environment. This typically appears in problems where there is a huge cost or risk involved in acquiring information (like in medical trials, or measurements performed in high risk environments) and then the goal of the sampling can be for instance, to allow to rapidly discard uninteresting options, or to be sure to avoid the most risky options.

Nevertheless, there is a second dimension of the problem to be considered. Ideally, among the available options, there are some that can indeed provide the information needed to reach the required optimality criterion. However, in many cases, the learner can only choose among a given, **fixed set of arms** and it might be the case that either the number of arms is very limited (insufficient to learn the needed information about the environment), or there can be plenty of arms that are non-informative for his objective, or even the measurements about the values of the options can suffer from huge variance. Therefore, infeasible optimal design might also appear in certain environments, caused by the additional constraints on the properties of the available options.

Applications

In this section we detail some examples of real-world applications that are relevant for the linear stochastic bandits setting and where the use of adaptive allocation strategies can lead to a better performance.

On-line recommendation. One of the real-world problems that fits into our framework and where linear stochastic bandits are already used in practice is the on-line recommendation of products and on-line advertising. Relying on the contextual information about users [Li et al., 2010], on personal history, and/or similarity of past behavior, one can learn the user preference, which can be seen as the underlying parameter to be estimated. The goal is to learn the user preference, while minimizing the regret of recommending products or displaying advertisements that are not in the direct interest of the user. More specifically, consider for instance a movie recommendation system and assume that all persons in a population of interest put similar weights on the features of the movie to decide whether they see a movie or not. After the user reveals his ratings for some (smartly chosen) movies, we can use this information to infer his rating on the rest of the movies. The problem becomes even more interesting when the recommender system has to deal with several users with similar preferences, in which case the system can transfer the knowledge gained through interactions with past users, thus needing a considerable lower number of interactions with a new user.

Medical trials. Another application example, which is also the historical motivation of stochastic multi-armed bandits [Thompson, 1933], comes from adaptively prescribing medical treatments. It is typically assumed that a drug has the same reaction on all patients and a doctor needs to find out which is the best drug, out of several drugs with unknown effects. The doctor is then faced with a trade-off since on the one hand he needs to explore the different drugs (to get information on their effects) and on the other hand he wants to exploit (administer to all patients) what seems to be the best drug. Assuming in addition that there exists a global relation between the type of drugs and their effects on the patients allows to focus more rapidly on the subset of the most beneficial drugs, possibly discarding some of the drugs without actually testing them, but based only on previous observations on some other drugs.

Active polling. Consider also the situation where a company has to make an accurate prediction of the result of a poll (e.g., an election). The company has relevant information on all the participants (age, profession, etc), but can only question a small number of them on their voting intentions. The obtained responses can then be used to infer a preference model for all the individuals in the group. To obtain an accurate estimation of the result of the election, the company should use a strategy which sequentially selects the most informative members of the group and adapts the repartition of the questionnaires such that even the voting intentions of the more heterogeneous groups are well estimated.

Contributions

In this context of sequential decision making problems where observations are expensive and there exists a linear global structure of the problem, our goal is to **exploit the global linear structure** of the setting to design sampling strategies concentrating on the most informative arms. We focus on three objectives, each relying on a different exploration-exploitation trade-off:

- **Identifying the best arm with a fixed confidence.** Here the objective is to design sample allocation strategies that allow to identify the best arm with a fixed confidence, while minimizing the number of pulled needed. This problem is by now well understood in the MAB setting, here we show how the strategies become very different when moving to the linear bandit setting. We begin with the study of the complexity of the best-arm identification problem in the linear bandit framework and we show the importance of exploiting the global linear structure to improve the estimate of the reward of near-optimal arms. Then, we propose and analyze static and adaptive allocation strategies and compare their empirical performance.

- **Sequential transfer of samples.** Here we consider the case where one needs to solve a sequence of similar linear bandit tasks. In this setting the performance measure is given by the per-task cumulative regret. Under the assumption that one can use very few costly samples from a linear bandit task, we investigate the

reduction in the per-task regret brought by transferring samples from other similar tasks. While the potential gains of these approaches have been well studied in batch learning scenarios, the transfer learning setting has received far less attention for sequential decision making problems with limited feedback. We propose multi-task algorithms that manage to avoid the negative transfer effects and are effective in reducing the per-task regret. We provide a theoretical analysis of the transfer problem in this setting and offer some empirical results for our proposed methods.

- **Minimizing the prediction error for the arm values.** Inspired by the Optimal Design of Experiments (OED) literature, we study two heteroscedastic regression problem, where the noise in the observations depends on the unknown arm-specific variance. Given a limited sampling budget n, our first goal is to direct the sampling allocation such that the obtained parameter estimate, denoted $\hat{\theta}_n$, minimizes the total prediction error of the value of the arms. This objective is also known as the *V-optimality criterion*.

  In addition, we investigate the sample allocation strategies that allow to predict with equal accuracy the value of each arm, the performance of the allocation strategy being given by the worst estimated arm value. This objective is also known as the *G-optimality criterion*.

For each setting, we formalize and study the complexity of the learning and estimation problems. Then, we propose and analyze the theoretical and empirical properties of adaptive allocation algorithms, designed to improve the estimate on the unknown parameters according to the optimization criterion at hand.

Objective \ Structure	**MAB**	**LB**
Cumulative Regret (CR)	[Robbins, 1952]...	[Auer, 2002]...
Simple Regret	[Audibert et al., 2010]...	Chapter 3
OED	[Antos et al., 2010]...	Chapter 4
CR + Transfer	[Gheshlaghi Azar et al., 2013]...	Chapter 5

Figure 1.1: Summary of the topics studied in the thesis.

Outline

The remainder of this thesis is organized as follows:

- In Chapter 2 we provide some preliminary notations, definitions, and tools from the literature. This includes a brief overview of known results in the multi-armed and linear bandit settings with stochastic rewards.

- In Chapter 3 we provide our results on the best arm identification problem in the linear bandits setting with a fixed confidence.

- In Chapter 4 we provide the formulation of two optimal design criteria in a linear bandit setting and provide some experimental results for bandit inspired algorithms.

- Then, in Chapter 5 we investigate the interest that transfer of samples might have when solving a sequence of similar linear bandit tasks, each having a limited per-task sample budget.

- We conclude the thesis in Chapter 6, discussing future directions and providing some closing remarks.

Chapter 2
Preliminaries

In this chapter we introduce the notation and basic assumptions later used in the settings of our study. We present here the concepts and tools from the Bandit Theory literature on which we build our work. We begin by introducing briefly the multi-armed stochastic bandit setting and its methods and results. Then, we focus on the setting where arms are correlated, in particular where a linear structure is assumed in the rewards. We then refer to some of the extensions to the classical objective of minimizing the cumulative regret, based on new performance measures. These extensions will be then theoretically and empirically analyzed in the following chapters of the thesis.

Contents

1 Stochastic Multi-armed Bandits

The stochastic multi-armed bandit (MAB) was first introduced in [Robbins, 1952] and takes the name from a casino slot machine. In this initial formulation, a player (or learner) has a finite gambling *budget* (given number of coins). In the casino there are several slot machines (one-armed bandits) and some of them might be better for the gambler, that is, they might return a higher monetary *reward*. Unsurprisingly, the goal of the gambler is to obtain as much reward as possible using his available gambling budget. For this, he needs to *explore* the casino to identify the slot machines that are the more profitable, but also to *exploit* the machines that seem to be better given the rewards he already observed. A smart coin allocation strategy is one that finds an effective exploration/exploitation trade-off, or in other words, a strategy that allows the gambler to maximize his sum of rewards.

Despite the name, the historical (and more important) motivation behind the MAB model dates back to [Thompson, 1933] and concerns the adaptive design of medical trials. As briefly explained in Chapter 1, the goal in this setting is to identify the best

drug out of a finite number of drugs with unknown effects. To do this, one sequentially selects one of the drug and administers it to the patient at hand. After observing the effects of the drug, one chooses the drug to be administered to the next patient. The assumption is that a drug has the same effects on all patients and the natural goal is to maximize the number of patients that are treated with the best drug. The trade-off that appears here comes on the one hand from the need to explore the different drugs (to get information on their effects) and on the other hand, from the will to exploit (administer to all patients) what seems to be the best drug.

1.1 Formalization

The stochastic multi-armed bandit game is a simple, repeated game, between a learner and the environment. This game can be formalized as follows: A learner is given a finite set of choices *(arms)*, $\mathcal{X} = \{x_1, ..., x_K\}$ with unknown returns and a certain number of allowed interactions with the environment *(budget/time horizon)*, denoted n. The total budget may or may not be known in advance by the user. With the goal of **maximizing the sum of the observed rewards**, the learner chooses at each time step $t = \{1, \dots, n\}$, an arm $x_{i,t}$, where $1 \leq i \leq K$, and observes an independent reward, denoted $r_{i,t}$, drawn from an unknown distribution ν_i of mean μ_i. This reward is the only information that the learner gets at time step t and we call this a *bandit feedback* setting, to stress the difference from the more traditional *full feedback* setting, where after each interaction the learner gets to observe rewards from all arms.

Algorithm 1 The stochastic MAB game

The player knows: K – number of arms $(x_1, \dots, x_i, \dots, x_K)$; n – total budget
Uncertain environment: $\nu_1, \dots, \nu_K$ – unknown reward distributions
for $t = 1, \dots, n$ **do**
  The player selects an arm $x_{i,t}$
  The player observes an independent reward $r_{i,t} \sim \nu_i$
end for
Goal: maximize $\sum_{t=1}^{n} r_{i,t}$.

After the learner consumes all the budget, we compare the sum of the rewards he observed with that of an oracle strategy that knows in advance the arm distributions $\nu_1, \dots, \nu_K$. In fact, when the distributions are known, the best choice is to pull at each time step the arm with the largest mean. But since the learner first has to explore the environment to get information about the reward distribution of each arm, unavoidably some suboptimal arms will be pulled.

The goal of the learner is then to design a sequence of arm pulls that leads to a sum of rewards which is as close as possible to that of the oracle that uses all the budget to pull the arm with the largest mean. This difference between the performance of the oracle and the expected sum of rewards obtained by the learner is called **cumulative pseudo-regret** and is the typical performance measure in the MAB setting. Formally, if we

denote the best arm $x^* \in \mathcal{X}$ as the arm having the largest mean $\mu^* = \max_{i=1,\dots,K} \mu_i$, then the cumulative pseudo-regret of the learner after n time steps is defined as:

$$R_n = \mathbb{E}\Big[n\mu^* - \sum_{t=1}^{n} r_{i,t}\Big], \tag{2.1}$$

where the expectation integrates over the randomness in the pull of the rewards. This initial objective of minimizing the cumulative regret in MAB has been widely studied. In the following, we briefly describe the main strategy and its performance.

1.2 Algorithms and Results

The paper [Lai and Robbins, 1985] proposes the first **asymptotically optimal** bandit algorithms in the case of Bernoulli distributions. These algorithms work by constructing arm indices based on the observed rewards. After each update, the arm with the largest index is chosen. To express more formally the properties of asymptotically optimal bandit strategies, we begin by restating in Prop. 2.1 the problem-dependent regret lower bound provided also in [Lai and Robbins, 1985].

Definition 2.1. *Let $N_{i,n}$ denote the number of pulls to arm x_i after consuming an overall budget of n pulls. We say a sampling strategy is **consistent** if for any stochastic bandit problem with a unique best arm, for any budget n and any $\alpha > 0$, the property*

$$\mathbb{E}[N_{i,n}] = o(n^{\alpha}) \tag{2.2}$$

holds true for any suboptimal arm x_i.

Proposition 2.1. *For any consistent strategy, for any stochastic bandit problem with Bernoulli distributions $\mathcal{B}(p)$ of parameters $p < 1$, the cumulative regret R_n is such that*

$$\liminf_{n\to\infty} \frac{R_n}{\log(n)} \geq \sum_{i:\mu_i<\mu^*} \frac{\mu^* - \mu_i}{KL(\mathcal{B}(\mu_i), \mathcal{B}(\mu^*))}, \tag{2.3}$$

where KL is the Kullback-Leibler divergence.

The previous result implies that any efficient allocation strategy samples at least $\Omega(\log(n))$ times each sub-optimal arm. Also, notice that a sub-optimal arm x_i is pulled $\frac{\mu^*-\mu_i}{KL(\mathcal{B}(\mu_i),\mathcal{B}(\mu^*))}\log(n)$ times. Thus, the closer μ_i is to μ^*, the more samples are allocated to arm x_i. Finally, we say that a bandit allocation strategy is asymptotically optimal, if it has the property that

$$\limsup_{n\to\infty} \frac{\mathbb{E}[N_{i,n}]}{\log(n)} \leq \frac{1}{KL(\mathcal{B}(\mu_i), \mathcal{B}(\mu^*))} \tag{2.4}$$

for any sub-optimal arm x_i.

The results of [Lai and Robbins, 1985] generated a lot of interest leading to extensions on multiple levels. On the one hand, asymptotically optimal algorithms were proposed for wider classes of distributions, notably the algorithms

in [Honda and Takemura, 2011] are shown to be asymptotically optimal for arbitrary distributions with finite support. On the other hand, in [Agrawal, 1995], a simple and explicit way of constructing the arm indices is proposed, namely the using as indices **upper-confidence bounds** (UCB) based on the empirical mean of arm-specific rewards and the number of arm pulls.

In the UCB class of algorithms, a particularly important breakthrough was the introduction of an algorithm with a **finite-time analysis** in [Auer et al., 2002]. Relying on the Chernoff-Hoeffding inequality, the algorithms in this paper consider arm indices given by a high-probability upper-bound on the expected value of the arm. Specifically, in the UBC1 algorithm of [Auer et al., 2002], after time step t, the score $B_{i,t}$ for arm $x_i \in \mathcal{X}$ is computed as

$$B_{i,t} = \widehat{\mu}_{i,t} + \sqrt{\frac{2\log t}{N_{i,t}}}, \tag{2.5}$$

where t is the number of total rewards observed so far, $N_{i,t}$ is the number of rewards coming from arm x_i and observed up to time t, and $\widehat{\mu}_{i,t}$ is the empirical mean of the $N_{i,t}$ rewards obtained by pulling arm x_i. Thus defined, the arm indices of the UCB algorithms can also be interpreted as the largest statistically plausible mean value of the arm, given the current available observations. As shown in the pseudo-code of

Algorithm 2 The UBC1 algorithm

The player knows: K – number of arms $(x_1, \ldots, x_i, \ldots, x_K)$; n – total budget
Uncertain environment: $\nu_1, \ldots, \nu_K$ – unknown distributions, bounded in $[0, 1]$
Initialization: Pull each arm once
for $t = K + 1, \ldots, n$ **do**
  Compute/Update arm indices: $B_{1,t} = \widehat{\mu}_{1,t} + \sqrt{\frac{2\log t}{N_{1,t}}}, \ldots, B_{K,t} = \widehat{\mu}_{K,t} + \sqrt{\frac{2\log t}{N_{K,t}}}$
  Pull arm $x_{i,t} = \arg\max_i B_{i,t}$
  Observe an independent reward $r_{i,t} \sim \nu_i$
end for
Goal: maximize $\sum_{t=1}^{n} r_{i,t}$.

UBC1 (Alg. 2), after computing the index for all arms, the UCB algorithm follows the principle of *Optimism in the Face of Uncertainty* (OFU) and chooses to pull the arm with the largest index. This principle thus suggests to follow what seems to be the best arm, based on the *optimistically* constructed arm-scores. The same principle is employed in a various sequential decision making problems, ranging from optimization to planning, as shown on the recent survey [Munos, 2014].

It is also important to note that from the definition of the arm index $B_{i,t}$ one can see that an arm x_i might have the largest index at time step t due to its empirical mean $\widehat{\mu}_{i,t}$ (the *exploitation* term) and/or to the uncertainty one has about the true arm value (the *exploration* term), directly dependent on how many rewards from arm x_i were actually observed. In particular, the exploration term in the construction of the $B_{i,t}$ guarantees that all arms will be pulled infinitely often.

Although efficient and achieving cumulative regret of order $\log(n)$ when the arm distributions have bounded support, the strategy introduced in [Auer et al., 2002] is not optimal. Improvements in the regret of finite-time strategies were later obtained by using more refined arm indices based on the empirical variance of the arms in [Audibert et al., 2009]. Still in the finite budget setting, for problems with finite support distributions, the authors of [Cappé et al., 2013] propose algorithms that use Kullback-Leibler confidence bounds to achieve asymptotically optimal sequential allocations.

Besides the direct improvements concerning the optimality of the algorithms, the classical stochastic MAB problem has been followed by multiple extensions, out of which two type of directions were particularly popular: First, there is the **Adversarial Bandits** family, where the assumption that the rewards coming from an arms are no longer i.i.d (as in the stochastic MAB), but chosen by an adversary. Then, there is the **Bandits with Structure** family, where the assumptions of global or partition-wise reward functions is made. Motivated by practical applications and leading to an important number of interesting problems and solutions, these extensions to the MAB setting are at the moment very popular. The various settings and results recently obtained are presented in the resent survey [Bubeck and Cesa-Bianchi, 2012].

In the following section, we are going to look in more detail only to the stochastic linear bandits setting, common to all the problems that we present in the following chapters of the thesis.

2 Stochastic Linear Bandits

An interesting variant of the MAB setup is the stochastic *linear bandit* problem, introduced in [Auer, 2002]. In the linear stochastic bandit (LB) setting, the input set $\mathcal{X} = \{x_1, \dots, x_K\}$, is a subset of $\mathbb{R}^d$. The set $\mathcal{X}$ is fixed and revealed to the learner. When pulling an arm $x \in \mathcal{X}$, the learner observes a noisy reward whose expected value is the inner product between x and an unknown parameter $\theta^* \in \mathbb{R}^d$:

$$r_t = x_t^\top \theta^* + \eta_t, \tag{2.6}$$

where x_t is the arm pulled at time step t, θ^* is the unknown parameter characterizing the underlying linear function and η_t is the noise affecting the reward observation at time step t. Typically in this setting the assumption is that the noise η is centered and has bounded variance. However, more details on the noise models we consider are to be specified in each problem definition.

The linear bandit setting becomes particularly interesting in applications where the number of arms is very large or at least larger than the available budget. In such cases, a learner facing a MAB would not even get to observe a reward from each arm. On the other hand, under the assumption of the global structure, in linear bandits, pulling an arm gives information about the parameter θ^* and indirectly, about the value of other arms. Therefore, the estimation of K mean-rewards in MAB is replaced here by the estimation of the d features of θ^*.

In designing sampling strategies for the linear stochastic bandits, we are still going to rely on upper-confidence bounds on the arms values. As in the MAB setting, here the *value* of an arm $x \in \mathcal{X}$ is given by $x^\top \theta^*$, that is, the expected reward that we obtain by pulling arm x. Similarly, the definition of the cumulative regret after n observations becomes:

$$R_n = \sum_{t=1}^{n} (x^{*\top}\theta^* - x_t^\top \theta^*) = \sum_{t=1}^{n} (x^* - x_t)^\top \theta^*, \tag{2.7}$$

where $x^* = \arg\max_{x \in \mathcal{X}} x^\top \theta^*$ is the best arm and x_t is the arm pulled at time step t. How big is the regret depends here on how well the parameter $\theta^* \in \mathbb{R}^d$ is estimated, or more precisely, whether the observed rewards provide enough information on θ^* to identify the best arm(s). Thus, in this setting a learner also has to find the appropriate balance between on the one hand, the *exploration* of the environment, that is, sampling arms that allow to improve the estimation of certain features of θ^* and on the other hand, the *exploitation* of what seems to be the best arm in order to maximize the sum of observed rewards. In the following, we describe how the arm indices are constructed by taking into account the uncertainty in the parameter characterizing the linear function.

2.1 Concentration Inequalities

Since in this setting the arms are correlated and the uncertainty of the environment is concentrated in the unknown parameter θ^*, clearly, to guide the sample allocation strategy we will rely on the estimate of θ^* obtained from the observed rewards. Indeed, after observing a sequence of n rewards, one can define a least-squares estimator for the parameter θ^* as follows. Let $\mathbf{x}_n = (x_1, \ldots, x_n) \in \mathcal{X}^n$ be a sequence of arms and $(r_1, \ldots, r_n)$ the corresponding observed (random) rewards. An unbiased estimate of θ^* can be obtained by an ordinary least-squares (OLS) method as

$$\hat{\theta}_n = A_{\mathbf{x}_n}^{-1} b_{\mathbf{x}_n}, \tag{2.8}$$

where $A_{\mathbf{x}_n} = \sum_{t=1}^{n} x_t x_t^\top \in \mathbb{R}^{d \times d}$ is the **design matrix** taking into account the pulls to arms $x \in \mathcal{X}$ and $b_{\mathbf{x}_n} = \sum_{t=1}^{n} x_t r_t \in \mathbb{R}^d$ is the vector of all the observed rewards. The OLS estimate enjoys a series of interesting properties. First, it is an unbiased estimator since $\mathbb{E}[\hat{\theta}_n | \mathbf{x}_n] = \theta^*$. Furthermore, for any arm $x \in \mathcal{X}$ the mean-squared error of the estimated reward $x^\top \hat{\theta}_n$ is

$$\mathbb{E}\Big[(x^\top \theta^* - x^\top \hat{\theta}_n)^2 | \mathbf{x}_n\Big] = \mathbb{V}[\eta]\ x^\top A_{\mathbf{x}_n}^{-1} x = \mathbb{V}[\eta]\ ||x||^2_{A_{\mathbf{x}_n}^{-1}}, \tag{2.9}$$

where $\mathbb{V}[\eta]$ is the variance of the additive noise and $||x||_M = x^\top M x$ denotes the M-weighted norm of x. More importantly, we can obtain high-probability bounds for the prediction error of $\hat{\theta}_n$ on the expected reward of the arms in the given set. What is important to take into account when constructing the confidence set is the sequence of pulls $\mathbf{x}_n = (x_1, \ldots, x_n) \in \mathcal{X}^n$. Specifically, the sequence of samples used in constructing the estimate $\hat{\theta}_n$ can be obtained either using a *static* or an *adaptive* allocation strategy.

The **static** allocation corresponds to an experimental design fixed in advance, where the sequence of arm pulls is not influenced by the observed rewards. This would correspond to the passive data collection. On the other hand, we say that a strategy is **adaptive** when at time step t, the choice of the arm to be pulled depends on the previously observed rewards: $r_1, r_2, \ldots, r_{t-1}$. For the two kind of strategies, we can obtain high-probability bounds for the prediction error of the least-squares estimate after an arbitrary number of observations n, as presented in the following propositions.

Proposition 2.2. *Let $c = 2H\sqrt{2}$ and $c' = 6/\pi^2$ and $\hat{\theta}_n$ the least-squares estimator obtained using the observed rewards coming from a **fixed** sequence $\mathbf{x}_n$. It holds true that*

$$\mathbb{P}\left(\forall n \in \mathbb{N}, \forall x \in \mathcal{X}, \left|x^\top \theta^* - x^\top \hat{\theta}_n\right| \leq c||x||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log(c' n^2 K/\delta)}\right) \geq 1 - \delta. \tag{2.10}$$

Proposition 2.3 (Thm. 2 in [Abbasi-Yadkori et al., 2011]). *Let $\hat{\theta}_n^\iota$ be the solution to the regularized least-squares problem with regularizer ι and let $A_{\mathbf{x}}^\iota = \iota I_d + A_{\mathbf{x}}$. Then for any $x \in \mathcal{X}$ and any **adaptive** sequence $\mathbf{x}_n$ such that at any step $n > 0$, x_n only depends on $(x_1, r_1, \ldots, x_{n-1}, r_{n-1})$, w.p. $1 - \delta$, we have*

$$\left|x^\top \theta^* - x^\top \hat{\theta}_n^\iota\right| \leq ||x||_{(A_{\mathbf{x}_n}^\iota)^{-1}} \left(H\sqrt{d \log\left(\frac{1 + nL^2/\iota}{\delta}\right)} + \iota^{1/2}||\theta^*||\right), \tag{2.11}$$

where $L = \max_{x \in \mathcal{X}} ||x||_2$.

In both bounds the key component in the prediction of the true value of arm x is the weighted norm $||x||_{A_{\mathbf{x}_n}^{-1}}$. In fact, one can see that the more $x \in \mathbb{R}^d$ is correlated to the design matrix $A_{\mathbf{x}_n}$, the more its norm weighted by the inverse of the design matrix will be small. If we restrict only to $x \in \mathcal{X}$ this follows the simple intuition that the more an arm gets pulled (information captured in the design matrix), the more we will be able to obtain an accurate estimation of its value.

As for the differences between the two bounds, note in particular the presence of an additional factor $\sqrt{d}$ in Eq. 2.11, the price to pay for adapting $\mathbf{x}_n$ to the samples. By definition, bandit algorithms adapt the allocation on the rewards observed over time. Therefore, in the sequel we will mostly rely on the high-probability bound on the prediction error of an estimate obtained through an adaptive allocation of resources.

2.2 Algorithms and Results

It is also by using the aforementioned concentration inequalities and high-probability bounds on the arms values that the bandit algorithms are build. Specifically, based on the empirical estimate $\hat{\theta}_n$, we construct **confidence ellipsoids** E_n, whose center is the empirical estimate and which containing all statistically possible values for θ^*. Importantly, the construction of the confidence set needs to be such that at each time step θ^* belongs to the set with probability $1 - \delta$. Let us define

$$E_n = \left\{\theta \in \mathbb{R}^d : ||\theta - \hat{\theta}_n||_{A_{\mathbf{x}_n}^\iota} \leq \beta_n(\delta)\right\}$$

where the term $\beta_n(\delta)$ is chosen such that

$$\mathbb{P}(\forall n \geq 1, \theta^* \in E_n) \geq 1 - \delta. \tag{2.12}$$

In fact, since the uncertainty here comes from the value of the d features of θ^*, then the width of the ellipsoid in a certain direction is determined by the accuracy in the estimates corresponding to that direction. Thus, while respecting the property in Eq. 2.12, the goal is to be able to shrink as fast as possible the volume of the confidence ellipsoid as we observe more and more samples. Also, based on the consistent confidence

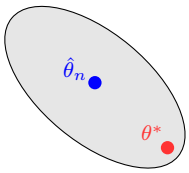


Figure 2.1: Illustration of the confidence ellipsoid E_n built around $\hat{\theta}_n$.

ellipsoid E_n, similarly to the MAB setting, the algorithms for cumulative regret minimization in the linear stochastic setting construct upper-bounds on the arms values using the margins of the confidence ellipsoids. Then, according to the OFU principle, the argorithms pull the arms with the largest index. Thus, at time step $t+1$, the algorithms select arm x_{t+1} defined as

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \left(x^\top \hat{\theta}_t + ||x||_{A_{\mathbf{x}_t}^{-1}} \beta_t \right). \tag{2.13}$$

This is a direct adaptation of the UCB type bounds, that start from the empirical estimate on the arm value $x^\top \hat{\theta}_t$, to which an *exploration* term is added, based on the uncertainty of estimating the reward of x. Specifically, here the uncertainty is captured in the width of the confidence set E_t in direction x. This construction of arm indices was introduced in [Auer, 2002], then the good empirical performance of a simple index following this type of construction was shown in [Li et al., 2010].

The construction of the arm indices can also be seen as a joint optimization problem. As proposed in [Abbasi-Yadkori et al., 2011], the empirical estimate $\hat{\theta}_n$ is used explicitly only for the construction of E_n. Then, the optimistic index for an arm $x \in \mathcal{X}$ will be given by the dot product of x with any convenient θ in E_n. Specifically, the index for arm x at time step t is given by

$$B_t(x) = \arg\max_{\theta \in E_n} x^\top \theta,$$

and the algorithm then selects the arm with the largest index

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} B_t(x) = \arg\max_{x \in \mathcal{X}} \max_{\theta \in E_n} x^\top \theta. \tag{2.14}$$

We now come back to the most important point of the method: the construction of the confidence ellipsoids E_n. Clearly, the tighter the confidence sets, the better the estimation of the arm values and therefore, and the more chances to minimize the overall regret. Several consistent (that is, respecting the property in Eq. 2.12) and efficient (that is, insuring the volume of E_n shrinks at a fast rate) methods have been proposed for the construction of confidence sets. First, in [Dani et al., 2008], the authors propose the CONFIDENCE BALL algorithm where the region of the confidence set is bounded at time step n is computed as

$$\beta_n(\delta) = \sqrt{\max\left(128d\log(n)\log(t^2/\delta), \frac{64}{9}\log^2\left(\frac{n^2}{\delta}\right)\right)}. \tag{2.15}$$

For this algorithm the authors prove that the cumulative regret is at most $O\left(d\log(n)\sqrt{n\log(n/\delta)}\right)$ with high probability. The state of the art result was obtained using a technique based on self-normalized bound for vector-value martingales, introduced in [Abbasi-Yadkori et al., 2011]. Here, the authors propose the OFUL algorithm, based on the confidence set[1]:

$$E_n = \left\{\theta \in \mathbb{R}^d : \left|\left|\theta^* - \hat{\theta}^\iota_n\right|\right|_{A^\iota_{\mathbf{x}_n}} \leq H\sqrt{d\log\left(\frac{1+nL^2/\iota}{\delta}\right)} + \iota^{1/2}||\theta^*||\right\}. \tag{2.16}$$

OFUL was shown to achieve better empirical performance and it also improves the theoretical performance achieving a regret that is at most $O\left(d\log(n)\sqrt{n} + \sqrt{dn\log(n/\delta)}\right)$ with high probability.

Now that we (briefly) introduced the results of stochastic linear bandit algorithms in the typical cumulative regret setting, in the next section we introduce the extensions to this model which considers different performance measures. In Chapter 3 we use the *simple regret*, then in Chapter 4 we consider two *optimality criteria* coming from the OED literature. These measures will then be detailed in their dedicated chapters. We then come back to the cumulative regret as a performance measure in Chapter 5, when another type of extension is considered, namely, the multi-task linear bandit.

3 Performance Measures

In this section we introduce different formulations of bandit problems which extend the typical bandit setting by no longer using as performance measure the cumulative regret, but rather focus on acquiring information on the environment. More specifically, we will present the metrics relevant to the problems studied in the following chapters of the thesis.

[1]The same technique and result were used to define in Eq. 2.11 the prediction error of the least-squares estimate obtained using an adaptive sampling strategy.

3.1 Simple regret

Typically, given a limited number of pulls, the goal of a learner is to maximize the sum of rewards, given rise to the so-called *exploration-exploitation* trade-off, where the exploration of the arms improves the knowledge about the environment, while the exploitation (according to the estimated arm values) leads to maximize the sum of rewards. More recently, a different viewpoint on the same problem has received considerable attention: In what is referred to as the *simple regret* setting, introduced in [Bubeck et al., 2009b], the forecaster's only goal is to identify the best arm. Thus, he suffers no regret for spending his budget on pulling sub-optimal arms. However, it is in his best interest to only pull **informative** arms that help him in identifying the best-arm. There are typically two type of constrains for the user in this setting:

- Either he has a **fixed budget** of pulls, in which case he must design an adaptive sample allocation strategy that identify the best arm with as much confidence as possible.
- Either he has to pull arms until reaching a **fixed confidence** in having identified the best arm. In this case, his goal is to design algorithms that reach this fixed confidence using as few arm pulls as possible, or in other words, algorithms with small sample complexity.

After reaching the fixed budget or the fixed confidence constraints, the design algorithm uses the information gathered in the process to return the **estimated best arm**, denoted $\widehat{x}(n)$ and defined by

$$\widehat{x}(n) = \{x_j \in \mathcal{X} : \mu(x_j(n)) = \max_{i=1,\dots,K} \mu(x_i(n))\}, \tag{2.17}$$

where the index n is the number of observations used by the learning algorithm before returning the estimated best arm. The performance measure in this setting is given by the difference between the expected reward of the true best arm and the expected reward of the arm returned by the adaptive strategy. Thus, by denoting $\mu(\widehat{x}(n))$ the mean of the estimated best arm $\widehat{x}(n)$ we obtain the following definition of the simple regret in the stochastic MAB setting:

$$R_n = \mu^* - \mu(\widehat{x}(n)). \tag{2.18}$$

Since arms are evaluated according to their expected reward, the difficulty of the best-arm identification (BAI) problem comes from being able to estimate the arms values up to a point where we can distinguish the one with the largest expected reward. The difficulty of the BAI problem thus depends on the **value gaps** between arms: the more the arms have similar values (that is, similar expected rewards), the harder it will be for the learner to identify the true best one. Let $\Delta(x, x') = (x - x')^\top \theta^*$ be the value *gap* between two arms, then we denote by $\Delta(x) = \Delta(x^*, x)$ the gap of x with respect to the optimal arm and by $\Delta_{\min} = \min_{x \in \mathcal{X}, x \neq x^*} \Delta(x)$ the minimum gap, where $\Delta_{\min} > 0$. We will rely on the gaps between arm values to define the complexity of the BAI problem.

In MAB, the BAI problem has been widely studied, in various settings: fixed budget [Audibert et al., 2010, Bubeck et al., 2009a], fixed confidence [Even-Dar et al., 2006, Jamieson et al., 2014], identifying the m best arms [Bubeck et al., 2013, Kaufmann and Kalyanakrishnan, 2013], or multi bandit BAI [Gabillon et al., 2011]. In the MAB case where arms are independent, the complexity of BAI is given by the problem-dependent quantity

$$H_{\text{MAB}} = \sum_{i=1}^{K} \frac{1}{\Delta_i^2}, \tag{2.19}$$

the inverse of the gaps between the best arm and the suboptimal arms. In the fixed-budget case, H_{MAB} determines the probability of returning the wrong arm, while in the fixed-confidence case, it characterizes the sample complexity. Also, the connection between the two settings has been studied in [Gabillon et al., 2012] and the recent paper [Kaufmann et al., 2015] introduces refined lower-bounds for both fixed-budget and fixed-confidence settings in the specific case of two armed-bandits, suggesting that the fixed-confidence setting has higher complexity.

In Chapter 3 we study in detail the BAI problem in the linear bandit case. We analyze the complexity of the problem and propose sample allocation strategies that manage to take into account the global structure of the problem to efficiently identify the best arm, given a fixed confidence.

3.2 OED inspired Optimality Criteria

The two performance measures presented so far consider the regret of the learner when his goal is to maximize the sum of rewards or to identify the best arm. Both of these goals require a very good knowledge of the expected rewards when pulling near-optimal arms. For the rest of the arms provided in the setting there is no explicit accuracy goal: either they are pulled only until their sub-optimality is certain (as in the cumulative regret case), or they are pulled because of their informative role in estimating the near-optimal arms (as in the simple regret case). In Chapter 4, we consider a more global approach, where the goal to use the sampling budget to obtain an overall good knowledge on the environment, that is, we aim at knowing with equally good accuracy the arm value for all arms.

These optimality criteria for the adaptive sampling strategies that we wish to study are inspired from the Optimal Experimental Design (OED) literature [Fedorov, 1972, Pukelsheim, 2006], where a limited number of experiments can be performed and the goal is to select the sequence of experiments that leads to maximize the overall information gained over the environment. Typically, the goal is to minimize the prediction error for the results of experiments, where for each arm $x \in \mathcal{X}$ we define the prediction error of $\hat{\theta}_n$ as the expected quadratic loss $L_n(x) = \mathbb{E}[(x^\top \hat{\theta}_n - x^\top \theta)^2]$. Overall, the loss of the allocation strategy corresponds to some global measure of the prediction error over all arms. In this case, the regret of a given adaptive strategy is defined as the difference between the its loss of that of an oracle allocation.

We discuss and analyze the properties of strategies designed for reaching the

G-optimality criterion in heteroscedastic noise scenarios (a setting also studied by [Wong and Cook, 1993]) and the V-optimality criterion. For the latter, we propose a sequential allocation strategy based on the recent paper by [Wiens and Li, 2014].

3.3 Cumulative Regret in Multi-task Linear Bandits

In Chapter 5, we study the case when the same learner is facing a sequence of unknown environments. Under the assumption that these environments have a *similarity* that can be exploited, we investigate whether the learner might improve the per-task cumulative regret by *transferring* observations and knowledge gathered at previous tasks.

4 Conclusion

This chapter gives a succinct overview of the basic notations, tools, and methods specific to the bandit literature to which belong the problems studied in the thesis. We first focused on the most classical setting, where the arms are independent and the goal is to maximize the sum of rewards. Then, getting closer to the settings proposed in our work, we introduced the assumptions specific to the linear stochastic bandits and the main method based on confidence ellipsoids. Lastly, we mentioned some extensions to the classical setting where the goal is no longer the typical maximization of the observed rewards, but instead, to identify the best arm or return an arm with lowest possible simple regret. We end the chapter with a glossary of the most frequent notations and their definitions.

Notation	Definitions, properties
$\mathcal{X}$	given set of arms
K	number of arms, $\lvert\mathcal{X}\rvert = K$
d	dimensionality of the problem, $\mathcal{X} \subset \mathbb{R}^d$
n	total sampling budget
t	current number of observed rewards: $1 \leq t \leq n$
x_i	arm i in the set $\mathcal{X}$, column vector of d features
L	upper-bound on the ℓ_2 norm of arms: $\max_{x \in \mathcal{X}} \lvert\lvert x\rvert\rvert_2 = L$
$x_{i,t}$	arm i in $\mathcal{X}$ selected at time step t
$r_{i,t}$	reward observed when pulling arm $x_{i,t}$
η	parameter for the noise in the rewards
x^*	best arm in $\mathcal{X}$
μ^*	expected value of x^*
$\widehat{\mu}_{i,t}$	empirical mean of arm x_i after t overall observations
$\sigma^2_{i,t}$	variance of arm x_i
$\widehat{\sigma^2}_{i,t}$	empirical variance of arm x_i after t overall observations
$N_{i,t}$	times that arm x_i was pulled up to total time t
δ	confidence parameter
$B_{i,t}$	high prob. upper-bound on the value of x_i after t overall observations
θ^*	vector of parameters characterizing the linear function, $\theta^* \in \mathbb{R}^d$
S	upper-bound on the ℓ_2 norm of θ^*
$\hat{\theta}_n$	OLS estimate of θ^* obtained after n observations
$\lvert\lvert x\rvert\rvert_M$	M-weighted norm of vector x, where $M \in \mathbb{R}^{d \times d}$ is a pos. def. matrix
$\mathbf{x}_n$	sequence of n pulls $\mathbf{x}_n = (x_1, \ldots, x_n) \in \mathcal{X}^n$
$A_{\mathbf{x}_n}$	design matrix obtained from the sequence of n pulls $\mathbf{x}_n$
$b_{\mathbf{x}_n}$	reward vector obtained from the sequence of n pulls $\mathbf{x}_n$
$\Delta(x, x')$	value gap between arm x and arm x': $\Delta(x, x') = (x - x')^\top \theta^*$
$\Delta(x)$	value gap between best arm x^* and arm x: $\Delta(x) = (x^* - x)^\top \theta^*$
$\Delta_{\min}$	smallest gap between x^* and the second best arm; $\Delta_{\min} > 0$
$\mathcal{Y}$	set of directions $y = x - x'$, where $x \neq x' \in \mathcal{X}$
$\mathcal{Y}^*$	set of directions $y = x^* - x'$, where $x' \neq x* \in \mathcal{X}$
m	number of tasks considered in the transfer scenario

Table 2.1: Table of Notations

Chapter 3

Best-arm Identification in Linear Bandits

We dedicate this chapter to the study of the best-arm identification (BAI) problem in the linear stochastic bandit[1]. We begin with the characterization of the complexity of the best-arm identification task when the environment has a global linear structure and the comparison with the known complexity results in the MAB case. Then, we introduce static and adaptive sample allocation strategies designed to identify the best arm with a **fixed confidence**, while minimizing the number of samples.

In our analysis, we show the importance of exploiting the global linear structure to improve the estimate of the reward of near-optimal arms. In particular, as opposed to stochastic linear bandits algorithms designed for cumulative regret, here we show how pulls to sub-optimal arms might be critical to obtain the information needed to distinguish among near-optimal arms.

We give sample complexity guarantees for the proposed strategies and also provide an empirical evaluation of their performance. Finally, we point out the connection to the G-optimal allocation strategy from the Optimal Experimental Design literature, which provides a worst-case optimal allocation for the best-arm identification problem.

Contents

[1] This chapter is a joint work with Alessandro Lazaric and Rémi Munos. A part of the chapter was presented in our NIPS paper [Soare et al., 2014b].

1 Introduction

While most of the literature in bandit theory focused on the problem of maximization of cumulative rewards, where the learner needs to trade-off exploration and exploitation, recently the *pure exploration* setting has gained a lot of attention. As briefly presented in Sect. 3.1, in the simple regret setting the learner uses the available budget to identify as accurately as possible the best arm, without trying to maximize the sum of rewards.

In the MAB setting, best-arm identification strategies start by considering all arms as potentially optimal, then proceed by sequentially discarding the arms which no longer fulfill the optimality condition. When *fixed budget* settings are considered (see e.g., [Audibert et al., 2010]), the budget is divided in $K-1$ rounds and at the end of each round, the arm with the lowest empirical mean is eliminated. On the other hand, in the *fixed confidence* settings (see e.g., [Even-Dar et al., 2006]), the decision to discard an arm depends on the probability of that arm to be dominated by another arm in the input set.

If in the MAB setting the best-arm identification (BAI) problem is by now well understood, with multiple available results in both fixed-budget and fixed-confidence constraints, in the linear stochastic bandit setting this problem is mostly unexplored in the literature. The fundamental difference between the MAB and the linear bandit best-arm identification strategies comes from the fact that in MAB an arm is no longer pulled as soon as its sub-optimality is evident (in high probability), while in the linear bandit setting even a sub-optimal arm may offer valuable information about the parameter vector θ^* and thus improve the accuracy of the estimation in discriminating among near-optimal arms. For instance, consider the situation when $K-2$ out of K arms are already discarded. In order to identify the best arm, MAB algorithms would concentrate the sampling on the two remaining arms to increase the accuracy of the estimate of their mean-rewards until the discarding condition is met for one of them. On the contrary, a linear bandit pure-exploration strategy would seek to pull the arm $x \in \mathcal{X}$ whose observed reward allows to refine the estimate θ^* along the dimensions which are more suited in discriminating between the two remaining arms.

In the rest of the chapter, we present our study on the sample complexity required to identify the best-linear arm with a given confidence. For this, we design strategies that explicitly take into account the geometry of the space. In Sect. 2 we introduce the details of the problem formulation and we restate the tools needed. In Sect. 3 we characterize the complexity of the best-arm identification in linear bandits. Then, we design both static (Sect. 4) and adaptive strategies (Sect. 5) for the BAI problem and we analyze their theoretical performance. We also show the practical effectiveness of our proposed strategy, by providing in Sect. 6 an illustration of their empirical performance. Finally, in Sect. 7 we draw conclusions and discuss future research directions.

2 Preliminaries

We consider the standard linear bandit model. $\mathcal{X} \subseteq \mathbb{R}^d$ denotes the finite set of arms, where $|\mathcal{X}| = K$ and we assume the ℓ_2-norm of any arm $x \in \mathcal{X}$, denoted by $||x||$, is upper-bounded by a constant L. Given an unknown parameter $\theta^* \in \mathbb{R}^d$, we assume that each time an arm $x \in \mathcal{X}$ is pulled, a random reward $r(x)$ is generated according to the linear model

$$r(x) = x^\top \theta^* + \eta, \tag{3.1}$$

where η is a zero-mean, i.i.d. noise bounded[2] in $[-\sigma; \sigma]$. Arms are evaluated according to their expected reward $x^\top \theta^*$ and we denote by $x^* = \arg\max_{x \in \mathcal{X}} x^\top \theta^*$ the best arm in $\mathcal{X}$. We assume that there is only one best arm in the set $\mathcal{X}$. Also, we introduce the notation $\Pi(\theta) = \arg\max_{x \in \mathcal{X}} x^\top \theta$ to refer to the best arm corresponding to an arbitrary parameter θ.

Let $\Delta(x, x') = (x - x')^\top \theta^*$ be the *value gap* between two arms, also we denote by $\Delta(x) = \Delta(x^*, x)$ the gap of x with respect to the optimal arm and by $\Delta_{\min} = \min\{\Delta(x) > 0, x \in \mathcal{X}\}$ the minimum gap. We also introduce the sets $\mathcal{Y} = \{y = x - x', \forall x, x' \in \mathcal{X}\}$ and $\mathcal{Y}^* = \{y = x^* - x, \forall x \in \mathcal{X}\}$ containing all the directions obtained as the difference of two arms (or an arm and the optimal arm) and we redefine accordingly the gap of a direction as $\Delta(y) = \Delta(x, x')$ whenever $y = x - x'$.

2.1 The problem

We study the best-arm identification problem. Let $\hat{x}(n)$ be the estimated best arm returned by a bandit algorithm after n steps. We evaluate the *quality* of $\hat{x}(n)$ by the simple regret

$$R_n = (x^* - \hat{x}(n))^\top \theta^*. \tag{3.2}$$

While different settings can be defined (see [Gabillon et al., 2012] for an overview), here we focus on the (ϵ, δ)-best-arm identification problem (the so-called PAC setting), where given ϵ and $\delta \in (0, 1)$ as input, the objective is to design an allocation strategy and a stopping criterion so that when the algorithm stops, the returned arm $\hat{x}(n)$ is such that

$$\mathbb{P}\Big(R_n \geq \epsilon\Big) \leq \delta, \tag{3.3}$$

within the smallest number of steps as possible. More specifically, we will focus on the case of $\epsilon = 0$ and we will provide high-probability bounds on the *sample complexity* n of the algorithm, that is, the number of pulls to the arms needed to identify the best arm with a confidence at least $1 - \delta$.

[2] The assumption can be extended to the sub-Gaussian noise.

2.2 The multi-armed bandit case

In MAB, this problem has been well-studied, in a variety of settings. The complexity of best-arm identification is characterized by the gaps between arm values, following the intuition that the more similar the arms, the more pulls are needed to distinguish between them. More formally, the complexity is given by the problem-dependent quantity

$$H_{\text{MAB}} = \sum_{\substack{i=1 \\ x_i \neq x^*}}^{K} \frac{1}{\Delta(x_i)^2} \ , \tag{3.4}$$

the inverse of the gaps between the best arm and the suboptimal arms. In the following paragraphs we give an overview of the algorithms proposed in the literature for this *pure-exploration* problem in the stochastic MAB setting.

2.2.1 Fixed-Budget BAI

In the fixed budget setting, H_{MAB} determines the probability of returning the wrong arm. There are two families of strategies for BAI in multi-armed bandit. First, the *uniform sampling* strategies that work by phases: during a phase all arms are pulled uniformly and at the end of the phase the arms with the lowest empirical mean are eliminated. Then, there is the *adaptive sampling* family of strategies where similarly to the typical cumulative regret scenario, the sampling is guided by arm-index policies.

For bounded bandit models [Audibert et al., 2010] propose two algorithms for BAI problems with fixed budget. Both proposed strategies are shown to be nearly optimal and each is representative of one of the family of strategies for this problem. The first strategy proposed in [Audibert et al., 2010] is an adaptive sampling algorithm which relies on the problem dependent complexity H_{MAB} to design an exploring strategy based on upper confidence bounds on the arms' values (UCB-E). The second strategy, called Successive Rejects (SR), is parameter-free and belongs to the uniform sampling family of strategies: The given budget is divided in $K-1$ phases and at the end of each phase, a decision is made. During a phase, the arms are pulled uniformly according to a fixed schedule. Then, at the end of the phase, the algorithm dismisses one of the arms, chosen randomly among those with the lowest empirical mean. At the next round, the remaining arms are again uniformly pulled according to the fixed schedule and the empirical means computed over the total number of pulls per arm (over all phases) are considered when deciding which arm to dismiss. The recommended arm is the unique arm remaining after the $K-1$ phase. What determines the performance here is the way in which the length of the phases is fixed. In [Audibert et al., 2010] it is shown that the SR strategy has phases lengths such that the available budget is not exceeded and the probability of error is minimized.

The algorithm (U)GapE, proposed in [Gabillon et al., 2012], belongs to the adaptive sampling family. This algorithm is an extension of UCB-E to the multi-bandit scenario, where multiple bandit problems are considered at the same time and the goal is to identify the best arm in each of them. This algorithm also needs to know the

complexity of the problem, since it relies on the gap between arms to construct the arm indices. A refinement of the (U)GapE, also introduced in [Gabillon et al., 2012] takes into account the variance of the arms ((U)GapE-V) to obtain tighter arm indices. For both algorithms the authors provide upper-bounds on the probability of error which decrease exponentially with the budget.

Another extension in the fixed-budget setting was proposed by [Bubeck et al., 2013]. The algorithm they introduce, called Successive Accepts and Rejects (SAR), is a phased, uniform sampling method, which based on efficiently predefined phase lengths (similar to the ones in SR), returns the m-best arms ($m > 1$) with high probability. For the same m-best arms problem and still under a fixed-budget constraint, [Kaufmann and Kalyanakrishnan, 2013] propose an algorithm belonging to the adaptive sampling family, based on an exploration rate depending on the budget.

2.2.2 Fixed-Confidence BAI

In the fixed-confidence case, H_{MAB} characterizes the sample complexity. This is the setting corresponding to initial BAI works (see for instance, [Bechhofer et al., 1968, Paulson, 1964, Jennison et al., 1982]), where typically uniform sampling strategies were used until reaching a stopping criterion triggered by the probability of returning the wrong arm being under a certain threshold. We distinguish here the Hoeffding Races algorithms, introduced in [Maron and Moore, 1993], where the arm elimination is based on Hoeffding confidence bound on the arms' values. More precisely, an arm x_i gets eliminated from the set of potentially optimal arms as soon as the upper-confidence bound of μ_i is smaller than the lower-confidence bound for the value of another arm $x \in \mathcal{X}$.

More recently, the best arm in the fixed-confidence case was studied in [Even-Dar et al., 2006], where they give upper-bounds on the sample complexity needed by δ-PAC algorithms. They first introduce the Successive-Elimination (SE) method, which eliminates one arm per phase and assumes that the expected rewards of the arms are known (but the matching of the arms to the expected values in not). They show that the lengths of the phases can be adapted such that with probability $1-\delta$ the worst arm is discarded using the smallest number of pulls to the arms, leading to a sample complexity of order $O\left(\log(\frac{K}{\delta})\sum_{i=2}^{K}\frac{1}{\Delta_i^2}\right)$. On the other hand, when the expected values are not given in advance, the SE algorithm needs longer phases, which leads to $O\left(\sum_{i=2}^{K}\frac{1}{\Delta_i^2}\log(\frac{K}{\delta\Delta_i})\right)$ pulls needed to identify the best arm with probability of at least $1-\delta$. Also in [Even-Dar et al., 2006], the Median Elimination (ME) method is proposed, this time for identifying the (ϵ, δ)-PAC arm, where a gap of at most ϵ is allowed between the returned arm and the true best arm. ME is a phased-algorithm, where at the end of each phase half of the arms is discarded (those with the worst empirical mean) based only on the samples observed during the phase. Using ME as a subroutine, the Exponential-Gap Elimination algorithm proposed in [Karnin et al., 2013] for Bernoulli bandits, is shown to obtain an improved sample complexity of

$O\left(\sum_{i=2}^{K} \frac{1}{\Delta_i^2} \log(\frac{1}{\delta} \log \frac{1}{\Delta_i})\right)$. Later on, a similar sample complexity was proven for the LIL'UCB algorithm proposed in [Jamieson et al., 2014].

For the m-best arms identification in a fixed-confidence setting, the LUCB algorithm is proposed in [Kalyanakrishnan et al., 2012]. The algorithm is also based on Hoeffding confidence bounds, but in contrast to the Hoeffding races, it does not sample uniformly the remaining arms. Instead, at each round the algorithm pulls the arms having the lowest upper-bound, and respectively, the largest lower-bound. Here the goal is to obtain an ϵ separation between these two critical arms.

Lastly, the connection between the fixed-budget and fixed-confidence settings has been studied in [Gabillon et al., 2012] and the recent paper [Kaufmann et al., 2015] introduces refined lower-bounds for both fixed-budget and fixed-confidence settings in the specific case of two armed-bandits, suggesting that the latter setting has higher complexity.

As regards the extensions of BAI scenarios in the MAB case to settings with *linear structure*, a first algorithm for the best-arm identification in linear bandits has been proposed in [Hoffman et al., 2014]. The authors consider a linear bandit model with Gaussian noise, specifically, each arm is normally distributed with mean $x_i^\top \theta^*$ and known, homoscedastic variance σ^2. The algorithm they propose is a version of (U)GapE (introduced in [Gabillon et al., 2012]), based on Bayesian confidence bounds. An upper-bound on the probability of error is given for the fixed-budget setting. Nonetheless, the complexity term considered in [Hoffman et al., 2014] is H_{MAB}, suggesting that their proposed algorithm does not take into account the correlations induced by the linear structure.

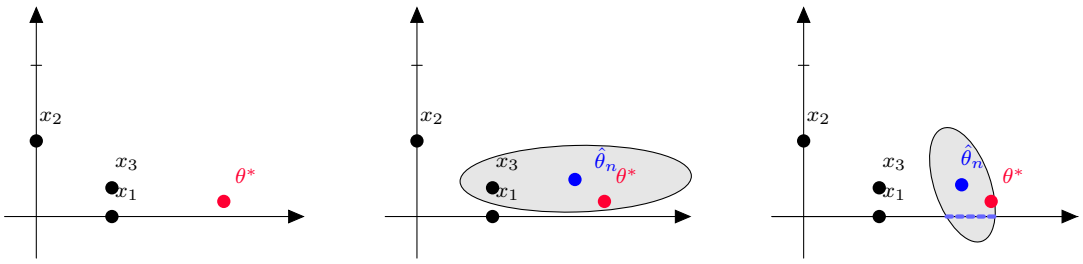
In this Chapter, we provide the first characterization of the complexity of the BAI problem in a linear bandit setting (Sect. 3) and the algorithms we introduce later on (Sect. 4, 5) exploit the global linear structure of the environment for designing the sample allocation. Clearly, the strategies designed for BAI in linear bandits will depend even more on problem dependent quantities, since in designing a strategy one must take into account (or rather, take advantage of) both the values gaps and the correlation between arms. In particular, in some cases in the linear bandit problems, it might be the case that "discarded arms" (that is, the arms that are sub-optimal with high probability), should continue to be pulled. This type of situation is due to the structure of the problem which makes the reward coming from an arm (as sub-optimal as it may be) informative and helpful in reducing the overall uncertainty of the environment. In the following, we present a simple example in support of this observation.

2.3 An Illustrative Example

As already pointed out, the linear structure of the problem leads strategies very different with respect to the in the typical multi-armed bandit, due to the global linear structure of the space that they need to take into account. Indeed, in this case, the observation of the reward for an arm also gives information about the expected rewards of the other arms (through the estimation of the common parameter θ^* which characterizes

the underlying linear function). Depending on the input set, it might happen that a clearly suboptimal arm is also the most informative with respect to θ^*. In the following, we offer a simple illustration of this situation.

Consider an input set in $\mathbb{R}^2$, containing only the following three arms: the two canonical arms, denoted x_1 and x_2, and a third arm, x_3, chosen to be very close to x_1 and having the same value on the X-axis. Also, suppose that the unknown parameter θ^* is very close to the abscissa axis, as depicted in Fig. 3.1(a).



(a) The setting. (b) $\forall \theta \in E_n, \Pi(\theta) = x_3$. (c) $\exists \theta \in \mathbb{E}_n : \Pi(\theta) = x_1$.

Figure 3.1: Illustration of sampling strategies for BAI in a simple setting in $\mathbb{R}^2$.

Given this setting, it is easy to see that x_1 and x_3 have very close values, whereas x_2 is suboptimal. Moreover, we notice that to identify the best arm, one should know whether θ^* belongs to first quadrant (in which case x_3 is the best arm), or to the forth quadrant (implying that x_1 is the best arm). As for the sampling, to be able to identify using as few many as possible whether x_1 or x_3 is the best arm, we need to design a strategy that is able to collect samples that help in better estimating the direction $y = x_1 - x_3$, which roughly corresponds to estimating the second feature of θ.

In Fig. 3.1(b) and Fig. 3.1(c) we illustrate the shape of two confidence sets E_n obtained after n pulls of two different sampling strategies. The sampling leading to the confidence set in Fig. 3.1(b) is efficient because it allows to say that x_3 is the best arm, given that $\forall \theta \in E_n$, we have that $\Pi(\theta) = x_3$. On the other hand, the strategy that led to the confidence set in Fig. 3.1(c) would need more than n samples before identifying with high probability the best arm. In fact, for the $\theta \in E_n$ that also belong to the forth quadrant, x_1 is the best arm. Basically, what makes the strategy in Fig. 3.1(b) more efficient is the fact that the observed rewards of this strategy allowed to reduce the uncertainty in direction $y = x_1 - x_3$ enough to remove the ambiguity in identifying the best arm. Lastly, notice that in the setting in Fig. 3.1(a), the pulls to arm x_2 allow to obtain more direct information about the value of θ^* in direction $y = x_1 - x_3$. Therefore, in this setting, pulling x_2 is the most effective way of getting to identify the best arm.

3 The Complexity of the Linear Best-Arm Identification Problem

As reviewed in Sect. 2, in the MAB case the complexity of the best-arm identification task is characterized by the reward gaps between the optimal and suboptimal arms. In this section, we propose an extension of the notion of complexity to the case of linear best-arm identification. We begin with the derivation of a lower-bound on the sample complexity needed for the best arm identification, in Sect. 3.1. Then, to illustrate the stopping conditions and the properties of efficient sample strategies, in Sect. 3.2 we analyze the sampling design obtained in the case of an *oracle* with access to the parameter θ^*.

3.1 Lower Bound on the Sample Complexity

Let $\nu = (\mathcal{X}, \theta)$ be a linear bandit environment, defined by the finite set of arms $\mathcal{X} = \{x_1, \dots, x_K\}, \mathcal{X} \in \mathbb{R}^d$, and a linear function characterized by the parameter vector $\theta \in \mathbb{R}^d$. We define an alternative environment $\nu' = (\mathcal{X}, \theta')$ on the same set $\mathcal{X}$, with a linear function characterized by a vector $\theta' \in \mathbb{R}^d$, such that the optimal arms are different in the two environments:

$$\Pi(\theta) \neq \Pi(\theta'). \tag{3.5}$$

As the previous definition implies, we consider a second environment characterized by a vector $\theta' \in \nu'$ that is "perturbed enough" with respect to the vector θ to ensure that the two environments do not have the same best arm, that is: $x^*_{\nu'} \neq x^*_{\nu}$. The following theorem gives a lower bound on the number of samples needed by an optimal static algorithm to distinguish between the two environments ν and ν' having the same input set, but with different best arms.

Theorem 3.1. *For any linear bandit environment $\nu = (\mathcal{X}, \theta)$, there exists an alternative environment $\nu' = (\mathcal{X}, \theta')$ having the same input $\mathcal{X}$ set but a different best arm, such that the number of pulls τ needed by any δ-PAC static allocation strategy to distinguish between the two problems is such that*

$$\mathbb{E}[\tau] \geq 2 \log(1/2\delta) \max_{y \in \mathcal{Y}^*} \frac{||y||^2_{A_\lambda^{-1}}}{\Delta(y)^2}, \tag{3.6}$$

where $A_\lambda = \sum_x \lambda(x) x x^\top$ captures the sample repartition over the arms x, as given by the design λ.

Proof. Without loss of generality, we use in the sequel $\Pi(\theta) = x_1$ and $\Pi(\theta') = x_2$. Similarly to the proof in [Kaufmann et al., 2015, Lemma 1,], let us consider the following inequalities. Define $\mathcal{A}$ a δ-PAC algorithm and let A be the event that algorithm $\mathcal{A}$ recommends x_1 as the best arm. Denoting by $d(x, y) = x \log(x/y) + (1-x) \log((1-x)/(1-y))$ the binary relative entropy, we have that the following statements hold true:

$$\mathbb{P}_\nu[A] \geq 1 - \delta \tag{3.7}$$

$$\mathbb{P}_{\nu'}[A] < \delta \tag{3.8}$$
$$d(\mathbb{P}_{\nu}[A], \mathbb{P}_{\nu'}[A]) \geq \log(1/2\delta). \tag{3.9}$$

Eq. 3.7 and Eq. 3.8 follow directly from the definition of PAC algorithms (Eq. 3.3) and the choice of event A. We thus have that the probability of event A in environment ν (denoted $\mathbb{P}_{\nu}[A]$) is higher than $1-\delta$, where x_1 is the true best arm of the environment. Also, the probability of event A in environment ν (denoted $\mathbb{P}'_{\nu}[A]$) smaller than δ for environment ν' with $x_{\nu'} = x_2 \neq x_1$. We also introduce the following helping lemma.

Proposition 3.1 (Lemma 1 in [Kaufmann et al., 2015]). *Let $N_{x_i}(t)$ denote the number of draws of arm x_i up to round t and suppose t is the stooping time of an algorithm $\mathcal{A}$. Also, let ν and ν' be two bandit models and A an event such that $0 < \mathbb{P}[A] < 1$. Then,*

$$\sum_{i=1}^{K} \mathbb{E}_{\nu}[N_{x_i}(t)] KL(\nu, \nu') \geq d(\mathbb{P}_{\nu}[A], \mathbb{P}_{\nu'}[A]). \tag{3.10}$$

Now let us introduce $\tilde{\varepsilon} = \theta' - \theta$ and assume a static algorithm $\mathcal{A}$ that performs the fixed sequence of pulls $\mathbf{x}_t$. Let $\mathbf{x}_t(\mathcal{A}) = (x_1, \dots, x_t) \in \mathcal{X}^t$ be the sequence of t arms pulled by the strategy $\mathcal{A}$ and let $(z_1, \dots, z_t)$ be the corresponding observed (random) rewards, where $z_i = x_i^\top \theta + \eta_i$ with $\eta \sim \mathcal{N}(0,1)$. Then, we can introduce the log-likelihood ratio of the observations up to time t under algorithm $\mathcal{A}$:

$$\begin{aligned}
L_t(z_1, \dots, z_t) &= \log\left(\prod_{s=1}^{t} \frac{\mathbb{P}_{\nu}(z_s|x_s)}{\mathbb{P}_{\nu'}(z_s|x_s)}\right) \\
&= \sum_{s=1}^{t} \log\left(\frac{\mathbb{P}_{\nu}(z_s|x_s)}{\mathbb{P}_{\nu'}(z_s|x_s)}\right) = \sum_{s=1}^{t} \log\left(\frac{\mathbb{P}(x_s^\top\theta + \eta_s|x_s)}{\mathbb{P}(x_s^\top\theta' + \eta'_s|x_s)}\right) \\
&= \sum_{s=1}^{t} \log\left(\frac{\mathbb{P}(z_s - x_s^\top\theta)}{\mathbb{P}(z_s - x_s^\top\theta')}\right) = \sum_{s=1}^{t} \log\left(\frac{\mathbb{P}(\eta_s)}{\mathbb{P}(\eta'_s)}\right) \\
&= \sum_{s=1}^{t} \log\left(\frac{\exp(-\eta_s^2/2)}{\exp(-\eta_s'^2/2)}\right) \qquad \text{(since both } \eta \text{ and } \eta' \sim \mathcal{N}(0,1)\text{)} \\
&= \sum_{s=1}^{t} \log\left(\exp(-\eta_s^2/2 + \eta_s'^2/2)\right) = \sum_{s=1}^{t} \frac{(z_s - x_s^\top\theta')^2 - (z_s - x_s^\top\theta)^2}{2} \\
&= \sum_{s=1}^{t} \frac{z_s^2 - 2z_s x_s^\top\theta' + (x_s^\top\theta')^2 - z_s^2 + 2z_s x_s^\top\theta - (x_s^\top\theta)^2}{2} \\
&= \sum_{s=1}^{t} \frac{2z_s x_s^\top(\theta - \theta') + (x_s^\top\theta' - x_s^\top\theta)(x_s^\top\theta' + x_s^\top\theta)}{2} \\
&= \sum_{s=1}^{t} \frac{-2z_s x_s^\top\tilde{\varepsilon} + x_s^\top\tilde{\varepsilon}(x_s^\top\theta + x_s^\top\tilde{\varepsilon} + x_s^\top\theta)}{2} \\
&= \sum_{s=1}^{t} (x_s^\top\tilde{\varepsilon}) \frac{-2z_s + 2x_s^\top\theta + x_s^\top\tilde{\varepsilon}}{2} = \sum_{s=1}^{t} (x_s^\top\tilde{\varepsilon}) \left(\frac{-2\eta_s + x_s^\top\tilde{\varepsilon}}{2}\right).
\end{aligned}$$

After a simple rewriting, we obtain

$$\mathbb{E}\Big[L_t\Big] = \mathbb{E}\left[\sum_{s=1}^{t}(x_s^\top\tilde{\varepsilon})\left(\frac{-2\eta_s + x_s^\top\tilde{\varepsilon}}{2}\right)\right] = \frac{1}{2}\mathbb{E}\left[\sum_{s=1}^{t}(x_s^\top\tilde{\varepsilon})^2\right] = \frac{1}{2}\mathbb{E}\left[\sum_{s=1}^{t}\tilde{\varepsilon}^\top x_s x_s^\top\tilde{\varepsilon}\right]$$
$$= \frac{1}{2}\mathbb{E}\left[\tilde{\varepsilon}^\top A_{\mathbf{x}_t}\tilde{\varepsilon}\right], \tag{3.11}$$

where $A_{\mathbf{x}_t} = \sum_{s=1}^{t} x_s x_s^\top$ is the design matrix corresponding to the fixed sequence $\mathbf{x}_t$.

Soft allocation. In the sequel we will often resort to the notion of design (or "soft" allocation) $\lambda \in \mathcal{D}^K$, which prescribes the *proportions* of pulls to arms $x \in \mathcal{X}$, where $\mathcal{D}^K$ denotes the simplex $\mathcal{X}$. The counterpart of the design matrix $A_{\mathbf{x}_t}$ for a design λ is the matrix $A_\lambda = \sum_{x\in\mathcal{X}} \lambda(x)xx^\top$. Also, from an allocation $\mathbf{x}_t$ we can derive the corresponding design $\lambda_{\mathbf{x}_t}$ as $\lambda_{\mathbf{x}_t}(x_i) = N_{i,t}/t$, where $N_{i,t}$ is the number of times arm x_i is selected in $\mathbf{x}_t$, and the corresponding design matrix is $A_{\mathbf{x}_t} = tA_\lambda$.[3] Then, for a random stopping time τ, the allocation of pulls to the arms becomes:

$$\lambda = \left[\frac{\mathbb{E}[N_{1,\tau}]}{\mathbb{E}[\tau]} \quad \dots \quad \frac{\mathbb{E}[N_{K,\tau}]}{\mathbb{E}[\tau]}\right]^\top, \tag{3.12}$$

where $\mathbb{E}[N_{i,\tau}]$ denotes the expected number of pulls to arm x_i up to round τ and $\sum_{i=1}^{K} N_{i,\tau} = \tau$.

Using the result in Eq.3.11 for the soft allocation λ, it follows that:

$$\mathbb{E}[\tau] = \frac{1}{2}\mathbb{E}\left[\tilde{\varepsilon}^\top\tau A_\lambda\tilde{\varepsilon}\right] = \frac{1}{2}\tilde{\varepsilon}^\top\mathbb{E}\left[\tau A_\lambda\right]\tilde{\varepsilon} = \frac{1}{2}\mathbb{E}[\tau](\tilde{\varepsilon}^\top A_\lambda\tilde{\varepsilon}). \tag{3.13}$$

Then, from Eq. 3.10 we obtain:

$$\frac{1}{2}\mathbb{E}[\tau](\tilde{\varepsilon}^\top A_\lambda\tilde{\varepsilon}) \geq \log(1/2\delta) \Leftrightarrow \mathbb{E}[\tau] \geq 2\log(1/2\delta)\frac{1}{\tilde{\varepsilon}^\top A_\lambda\tilde{\varepsilon}}\ . \tag{3.14}$$

Lower bound. To get the lower bound on τ, let us now consider the ε which allows to maximize the lower-bound in Eq. 3.14. Denoting $\mathcal{Y}_\nu^* = \{y = x_\nu^* - x, \forall x \in \mathcal{X}\}$, we have that he condition that the best arms in the two environments differ implies that there exists at least one $y \in Y_\nu^*$ such that:

$$y^\top\theta' < 0 \Leftrightarrow y^\top\theta - y^\top\theta' > y^\top\theta \Leftrightarrow y^\top\varepsilon > y^\top\theta \Leftrightarrow y^\top\varepsilon > \Delta_\theta(y). \tag{3.15}$$

The smallest ε satisfying this condition is given by the solution of the following (relaxed) minimization problem, where $\alpha > 0$:

$$\begin{aligned} &\min\ \varepsilon^\top A_\lambda\ \varepsilon \\ &\text{s.t.}\ \ \exists y \in Y_\nu^*, y^\top\varepsilon \geq \Delta_\theta(y) + \alpha. \end{aligned} \tag{3.16}$$

[3] A more precise notation would be $A_{\lambda_{\mathbf{x}_t}}$, but to improve readability we drop the dependency on the sequence $\mathbf{x}_t$ in the notation of the design matrix. Likewise, we will use λ instead of $\lambda_{\mathbf{x}_t}$ to refer to the soft allocation corresponding to the sequence $\mathbf{x}_t$, whenever this does not create ambiguity.

The Lagrangian of the problem becomes:

$$L(\varepsilon, \gamma) = \frac{1}{2}\varepsilon^\top A_\lambda \, \varepsilon + \gamma(-y^\top \varepsilon + \Delta_\theta(y) + \alpha).$$

$$\frac{\partial L}{\partial \varepsilon} = A_\lambda \, \varepsilon - \gamma y = 0 \Leftrightarrow A_\lambda \, \varepsilon = \gamma y \Leftrightarrow A_\lambda^{1/2} \, \varepsilon = \gamma A_\lambda^{-1/2} y.$$
$$\frac{\partial L}{\partial \gamma} = -y^\top \varepsilon + \Delta_\theta(y) + \alpha = 0 \Leftrightarrow y^\top \varepsilon = \Delta_\theta(y) + \alpha.$$

Thus, we have that $y^\top \varepsilon = y^\top A_\lambda^{-1/2} A_\lambda^{1/2} \varepsilon = ||y||_{A_\lambda^{-1}} \cdot ||\varepsilon||_{A_\lambda}$, since y and ε are proportional vectors and for all $\alpha > 0$ we obtain:

$$||\varepsilon||_{A_\lambda} \geq \frac{y^\top \varepsilon}{||y||_{A_\lambda^{-1}}} = \frac{\Delta_\theta(y)}{||y||_{A_\lambda^{-1}}} + \frac{\alpha}{||y||_{A_\lambda^{-1}}} \, . \tag{3.17}$$

For every direction $y_i \in Y_\nu^*$ there exists a corresponding smallest deviation $\varepsilon_i = \theta - \theta'$ such that $\Pi(\theta) \neq \Pi(\theta')$. Let ε correspond to the vector which allows to maximize the lower-bound in Eq. 3.14, over all directions $y_i \in Y_\nu^*$:

$$||\varepsilon||_{A_\lambda} \geq \min_{y \in \mathcal{Y}_\nu^*} \left(\frac{\Delta_\theta(y)}{||y||_{A_\lambda^{-1}}} + \frac{\alpha}{||y||_{A_\lambda}^{-1}} \right) \, . \tag{3.18}$$

From Eq. 3.9 it follows that the expected number of samples τ needed to distinguish between two problems $\nu(\mathcal{X}, \theta)$ and $\nu'(\mathcal{X}, \theta')$, where $\Pi(\theta) \neq \Pi(\theta')$, is lower-bounded by the quantity

$$\mathbb{E}[\tau] \geq 2 \log(1/2\delta) \frac{1}{\varepsilon^\top A_\lambda \varepsilon}$$

and by Eq. 3.18 we obtain:

$$\mathbb{E}[\tau] \geq 2 \log(1/2\delta) \frac{1}{\min_{y \in \mathcal{Y}^*} \left(\frac{\Delta(y)}{||y||_{A_\lambda^{-1}}} + \frac{\alpha}{||y^*||_{A_\lambda}^{-1}} \right)^2} \, .$$

Finally, by letting $\alpha \to 0$, we obtain

$$\mathbb{E}[\tau] \geq 2 \log(1/2\delta) \max_{y \in \mathcal{Y}^*} \frac{||y||_{A_\lambda^{-1}}^2}{\Delta(y)^2}.$$

□

In the following, we provide a geometrical intuition of the lower-bound by studying the stopping and sampling rules used by a static sample allocation strategy that minimizes the lower-bound.

3.2 Geometrical Interpretation

In this section, based on an optimal static allocation strategy, we are going to offer some geometrical interpretation and description of optimal sampling and stopping rules for the BAI problem in linear bandits. Besides providing more intuition on the meaning of the lower-bound, the description presented in the following will also serve as a basis for the learning algorithms that we introduce in the next sections.

3.2.1 Stopping condition

Let $\mathcal{C}(x) = \{\theta \in \mathbb{R}^d, x \in \Pi(\theta)\}$ be the set of parameters θ which admit x as an optimal arm. As illustrated in Fig. 3.2, $\mathcal{C}(x)$ is the cone defined by the intersection of half-spaces such that $\mathcal{C}(x) = \cap_{x' \in \mathcal{X}} \{\theta \in \mathbb{R}^d, (x - x')^\top \theta \geq 0\}$ and all the cones together form a partition of the Euclidean space $\mathbb{R}^d$.

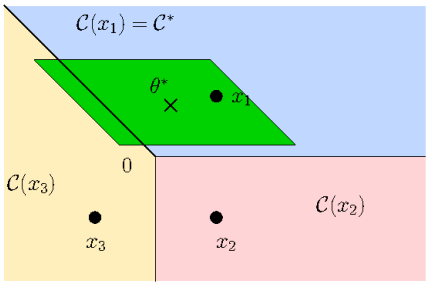


Figure 3.2: The cones corresponding to three arms (dots) in $\mathbb{R}^2$. Since $\theta^* \in \mathcal{C}(x_1)$, then $x^* = x_1$. The confidence set $\mathcal{S}^*(\mathbf{x}_n)$ (in green) is aligned with the directions $x_1 - x_2$ and $x_1 - x_3$. Given the uncertainty in $\mathcal{S}^*(\mathbf{x}_n)$, both x_1 and x_3 may be optimal.

In the following, we call a static allocation that uses the knowledge of θ^* to define the sampling and stopping rules an *oracle*. Since the oracle knows θ^*, it also knows the cone $\mathcal{C}(x^*)$ containing all the parameters for which x^* is optimal. Furthermore, we assume that for any allocation $\mathbf{x}_n$, it is possible to construct a consistent confidence set $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathbb{R}^d$ centered in θ^* such that the least-squares estimate $\hat{\theta}_n$ belongs to $\mathcal{S}^*(\mathbf{x}_n)$ with high probability:

$$\mathbb{P}\Big(\hat{\theta}_n \in \mathcal{S}^*(\mathbf{x}_n) \text{ and } \mathcal{S}^*(\mathbf{x}_n) \text{ is centered in } \theta^*\Big) \geq 1 - \delta. \tag{3.19}$$

As a result, the oracle stopping criterion simply checks whether the confidence set $\mathcal{S}^*(\mathbf{x}_n)$ is contained in $\mathcal{C}(x^*)$ or not. In fact, for every allocation strategy $\mathbf{x}_n$, whenever the confidence overlaps the cones of different arms $x \in \mathcal{X}$, there is ambiguity in the identity of the best arm. On the other hand, when the entire confidence set belongs to only one cone, then it means that all plausible values of the parameter would recommend the same best arm and there no longer need to sample. For the case of the oracle that knows $\mathcal{C}(x^*)$, when $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ all possible values of θ^* are included with high probability in the "right" cone $\mathcal{C}(x^*)$, then the optimal arm is returned.

Lemma 3.1. *Let $\mathbf{x}_n = (x_1, \ldots, x_n)$ be an allocation such that $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$. Then $\mathbb{P}\Big(\Pi(\hat{\theta}_n) \neq x^*\Big) \leq \delta$.*

Proof. The proof follows from the fact that if $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ and $\hat{\theta}_n \in \mathcal{S}^*(\mathbf{x}_n)$ with high probability, then $\hat{\theta}_n \in \mathcal{C}(x^*)$ which implies that $\Pi(\hat{\theta}_n) = x^*$ by definition of the cone $\mathcal{C}(x^*)$. □

3.2.2 Arm selection strategy

From the previous lemma it follows that the objective of an arm selection strategy is to define an allocation $\mathbf{x}_n$ which leads to $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ as quickly as possible.[4] Since this condition only depends on deterministic objects ($\mathcal{S}^*(\mathbf{x}_n)$ and $\mathcal{C}(x^*)$), it can be computed independently from the actual reward realizations. From a geometrical point of view, this corresponds to choosing arms so that the confidence set $\mathcal{S}^*(\mathbf{x}_n)$ shrinks into the optimal cone $\mathcal{C}(x^*)$ within the smallest number of pulls. To characterize this strategy we need to make explicit the form of $\mathcal{S}^*(\mathbf{x}_n)$. Intuitively speaking, the more $\mathcal{S}^*(\mathbf{x}_n)$ is "aligned" with the boundaries of the cone, the easier it is to shrink it into the cone. More formally, the condition $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ is equivalent to

$$\forall x \in \mathcal{X}, \forall \theta \in \mathcal{S}^*(\mathbf{x}_n), (x^* - x)^\top \theta \geq 0 \quad \Leftrightarrow \quad \forall y \in \mathcal{Y}^*, \forall \theta \in \mathcal{S}^*(\mathbf{x}_n), y^\top(\theta^* - \theta) \leq \Delta(y).$$

Then, we can simply use Prop. 2.2, which bounds the prediction error of a fixed allocation strategy. In fact, using this concentration inequality, we can directly control the term $y^\top(\theta^* - \theta)$ and define the oracle confidence set

$$\mathcal{S}^*(\mathbf{x}_n) = \left\{ \theta \in \mathbb{R}^d, \forall y \in \mathcal{Y}^*, y^\top(\theta^* - \theta) \leq c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_{c'}(K^2/\delta)} \right\}, \tag{3.20}$$

where the notation $\log_{c'}(K^2/\delta)$ is shorthand for $\log(c' n^2 K^2/\delta)$. The change in the logarithmic term of Prop. 2.2 appears whenever we use the concentration inequality for directions $y \in \mathcal{Y}$ and is due to an additional union bound. Now, through the use of Prop. 2.2, the stopping condition $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ is equivalent to verifying that, for any $y \in \mathcal{Y}^*$,

$$c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_{c'}(K^2/\delta)} \leq \Delta(y). \tag{3.21}$$

From this condition, the oracle allocation strategy simply follows as

$$\mathbf{x}_n^* = \arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_{c'}(K^2/\delta)}}{\Delta(y)} = \arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{||y||_{A_{\mathbf{x}_n}^{-1}}}{\Delta(y)}. \tag{3.22}$$

Notice that this sample allocation does not return an uniformly accurate estimate of θ^* but it rather pulls arms that allow to reduce the uncertainty of the estimation of θ^* over the directions of interest (i.e., $\mathcal{Y}^*$) below their corresponding gaps. This implies that the objective of Eq. 3.22 is to exploit the global linear assumption by pulling any arm in $\mathcal{X}$ that could give information about θ^* over the directions in $\mathcal{Y}^*$, so that directions with small gaps are better estimated than those with bigger gaps. In the sequel, we will refer to this design rule as the "$\mathcal{XY}$-Oracle" strategy.

[4] Notice that by definition of the confidence set and since $\hat{\theta}_n \to \theta^*$ as $n \to \infty$, then any strategy repeatedly pulling all the arms would eventually meet the stopping condition.

3.2.3 Complexity

We are now ready to define the sample complexity of the oracle, which corresponds to the number of steps needed by the deterministic static allocation in Eq. 3.22 to achieve the stopping condition in Eq. 3.21. Formally, denoting N^* the sample complexity of the oracle, achieving the stopping condition corresponds to

$$N^* = \min \left\{ n \in \mathbb{N}, \min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_{c'}(K^2/\delta)}}{\Delta(y)} \leq 1 \right\}. \tag{3.23}$$

From a technical point of view, it is more convenient to express the complexity of the problem in terms of the optimal design (soft allocation) instead of the discrete allocation $\mathbf{x}_n$. Let $\rho^*(\lambda) = \max_{y \in \mathcal{Y}^*} ||y||^2_{\Lambda_\lambda^{-1}} / \Delta^2(y)$ be the square of the objective function in Eq. 3.22 for any design $\lambda \in \mathcal{D}^k$. We define the complexity of a linear best-arm identification problem as the performance achieved by the optimal design $\lambda^* = \arg\min_{\lambda \in \mathcal{D}^k} \rho^*(\lambda)$, that is

$$H_{\text{LB}} = \min_{\lambda \in \mathcal{D}^k} \max_{y \in \mathcal{Y}^*} \frac{||y||^2_{\Lambda_\lambda^{-1}}}{\Delta^2(y)} = \rho^*(\lambda^*). \tag{3.24}$$

This definition of complexity is less explicit than in the case of H_{MAB} but it contains similar elements, notably the inverse of the gaps squared. Nonetheless, instead of summing the inverses over all the arms, H_{LB} implicitly takes into consideration the correlation between the arms in the term $||y||^2_{\Lambda_\lambda^{-1}}$, which represents the uncertainty in the estimation of the gap between x^* and x (when $y = x^* - x$). As a result, from Eq. 3.21 the sample complexity becomes

$$N^* = c^2 H_{\text{LB}} \log_{c'}(K^2/\delta), \tag{3.25}$$

where we use the fact that, if implemented over n steps, λ^* induces a design matrix $A_{\lambda^*} = n\Lambda_{\lambda^*}$ and $\max_y ||y||^2_{A_{\lambda^*}^{-1}} / \Delta^2(y) = \rho^*(\lambda^*)/n$. N^* can be interpreted as the number of steps needed by an optimal deterministic static allocation, before reaching the stopping condition that requires the confidence set to be included into the right cone. N^* matches up to a numerical multiplicative factor the lower-bound for any fixed allocation strategy, previously defined in Eq. 3.6.

We conclude the characterization of the difficulty of the BAI task by providing a range on the complexity and a comparison to H_{MAB}.

3.2.4 Range for H_{LB}

The following lemma introduces bound on the complexity of the BAI problem in the linear bandit setting.

Lemma 3.2. *Given an arm set $\mathcal{X} \subseteq \mathbb{R}^d$ and a parameter θ^*, the complexity H_{LB} (Eq. 3.24) is such that*

$$\frac{\max_{y \in \mathcal{Y}^*} ||y||^2}{(L\Delta^2_{\min})} \leq H_{LB} \leq \frac{4d}{\Delta^2_{\min}}. \tag{3.26}$$

Furthermore, whenever $\mathcal{X}$ is the canonical basis, the problem reduces to a MAB and $H_{MAB} \leq H_{LB} \leq 2H_{MAB}$.

Proof. **Upper-bound.** We have the following sequence of inequalities

$$\max_{y\in\mathcal{Y}_*} \frac{||y||^2_{\Lambda_\lambda}}{\Delta^2(y)} \leq \frac{1}{\Delta^2_{\min}} \max_{y\in\mathcal{Y}_*} ||y||^2_{\Lambda_\lambda} \leq \frac{4}{\Delta^2_{\min}} \max_{x\in\mathcal{X}} ||x||^2_{\Lambda_\lambda},$$

where the second inequality comes from a triangle inequality on $||y||^2_{\Lambda_\lambda}$. Thus we obtain

$$\rho^*(\lambda^*) = \min_{\lambda\in\mathcal{D}^k} \max_{y\in\mathcal{Y}_*} \frac{||y||^2_{\Lambda_\lambda}}{\Delta^2(y)} \leq \frac{4}{\Delta^2_{\min}} \min_{\lambda\in\mathcal{D}^k} \max_{x\in\mathcal{X}} ||x||^2_{\Lambda_\lambda} = \frac{4d}{\Delta^2_{\min}},$$

where the last equality follows from the Kiefer-Wolfowitz equivalence theorem [Kiefer and Wolfowitz, 1960].

Lower-bound. We focus on the numerator $y^\top \Lambda_\lambda^{-1} y$. Since Λ_λ is a positive definite matrix, we define its decomposition $\Lambda_\lambda = Q\Gamma Q^\top$, where Q is an orthogonal matrix and Γ is the diagonal matrix containing the eigenvalues. As a result the numerator can be written as

$$y^\top \Lambda_\lambda^{-1} y = y^\top Q \Gamma^{-1} Q^\top y = w^\top \Gamma^{-1} w,$$

where we renamed $Q^\top y = w$. If we denote by $\gamma_{\max}$ the largest eigenvalue of Λ_λ (i.e., the largest value in Γ), then

$$w^\top \Gamma^{-1} w \geq 1/\gamma_{\max} w^\top w = 1/\gamma_{\max} ||y||^2.$$

The largest eigenvalue $\gamma_{\max}$ is upper-bounded by the sum of the largest eigenvalues of the matrices $\lambda(x)xx^\top$ which is $\lambda(x)||x||_2$. As a result, we obtain the bound $\gamma_{\max} \leq \sum_x \lambda(x)||x||_2 \leq L$, since $||x||_2 \leq L$ and λ is in the simplex. Thus we have

$$\min_{\lambda\in\mathcal{D}^k} \max_{y\in\mathcal{Y}_*} \frac{||y||^2_{\Lambda_\lambda}}{\Delta^2(y)} \geq \frac{1}{L} \max_{y\in\mathcal{Y}^*} \frac{||y||^2}{\Delta(y)^2} \geq \frac{\max_{y\in\mathcal{Y}^*} ||y||^2}{L\Delta^2_{\min}}.$$

Comparison with K-armed bandit. Finally, we show how the sample complexity reduces to the known quantity in the MAB case. If the arms in $\mathcal{X}$ coincide with the canonical basis of $\mathbb{R}^d$, then for any allocation λ the design matrix Λ_λ becomes a diagonal matrix of the form $\mathrm{diag}(\lambda(x_1), \ldots, \lambda(x_K))$. As a result, we obtain

$$H_{\mathrm{LB}} = \min_{\lambda\in\mathcal{D}^k} \max_{y\in\mathcal{Y}^*} \frac{||y||^2_{A_\Lambda^{-1}}}{\Delta^2(y)} = \min_{\lambda\in\mathcal{D}^k} \max_{x\in\mathcal{X}-\{x^*\}} \frac{1/\lambda(x) + 1/\lambda(x^*)}{\Delta(x)^2}.$$

If we use the allocation $\lambda(x) = 1/(\nu\Delta^2(x))$ and $\lambda(x^*) = 1/(\nu\Delta_{\min})$, with $\nu = 1/\Delta^2_{\min} + \sum_{x\neq x^*} 1/\Delta^2(x)$, we obtain

$$H_{\mathrm{LB}} \leq \max_{x\in\mathcal{X}-\{x^*\}} \frac{\nu\Delta^2(x) + \nu\Delta^2_{\min}}{\Delta^2(x)} = \max_{x\in\mathcal{X}-\{x^*\}} \nu + \nu\frac{\Delta^2_{\min}}{\Delta^2(x)}$$

$$= 2\nu = 2\Big(\frac{1}{\Delta_{\min}^2} + \sum_{x\neq x^*} \frac{1}{\Delta^2(x)}\Big) = 2H_{\mathrm{MAB}}.$$

On the other hand, letting $\tilde{x}$ be the second best arm and $\Delta(x^*) = \Delta_{\min}$, we have that

$$\begin{aligned}
H_{\mathrm{LB}} &= \min_{\lambda\in\mathcal{D}^k} \max_{x\neq x*} \frac{1/\lambda(x) + 1/\lambda(x^*)}{\Delta^2(x)} \\
&= \min_{\lambda\in\mathcal{D}^k} \max \left\{ \max_{x\neq x*} \frac{1/\lambda(x) + 1/\lambda(x^*)}{\Delta^2(x)}; \frac{1/\lambda(\tilde{x}) + 1/\lambda(x^*)}{\Delta^2(x^*)} \right\} \\
&\geq \min_{\lambda\in\mathcal{D}^k} \max \left\{ \max_{x\neq x*} \frac{1/\lambda(x)}{\Delta^2(x)}; \frac{1/\lambda(x^*)}{\Delta^2(x^*)} \right\} \\
&= \min_{\lambda\in\mathcal{D}^k} \max_{x\in\mathcal{X}} \frac{1/\lambda(x)}{\Delta^2(x)}.
\end{aligned}$$

We set $\frac{1/\lambda(x)}{\Delta^2(x)}$ equal to a constant c and thus we get $\lambda(x) = \frac{1}{c\Delta^2(x)}$. Since $\frac{1}{c}\sum_{x\in\mathcal{X}} \frac{1}{\Delta^2(x)} = 1$, it follows that:

$$c = \sum_{x\in\mathcal{X}} \frac{1}{\Delta^2(x)} = \sum_{x\neq x^*} \frac{1}{\Delta^2(x)} + \frac{1}{\Delta_{\min}^2} = H_{\mathrm{MAB}}.$$

Thus, we get that $H_{\mathrm{MAB}} \leq H_{\mathrm{LB}} \leq 2H_{\mathrm{MAB}}$. This shows that H_{LB} is a well defined notion of complexity for the linear best-arm identification problem and the corresponding sample complexity N^* is coherent with existing results in the MAB case.

□

The previous bounds show that $\Delta_{\min}$ plays a significant role in defining the complexity of the problem, while the specific shape of $\mathcal{X}$ impacts the numerator in different ways: In the worst case the full dimensionality d appears (upper-bound), and more arm-set specific quantities, such as the norm of the arms L and of the directions $\mathcal{Y}^*$, appear in the lower-bound.

4 Static Allocation Strategies

The oracle stopping condition (Eq. 3.21) and allocation strategy (Eq. 3.22) cannot be implemented in practice since $\mathcal{C}(x^*)$, the gaps $\Delta(y)$, and the directions $\mathcal{Y}^*$ are unknown. In this section we investigate how to define algorithms that only rely on the information available from $\mathcal{X}$ and the samples collected over time. We introduce an empirical stopping criterion and two static allocations.

Empirical stopping criterion. The stopping condition $\mathcal{S}^*(\mathbf{x}_n) \subseteq \mathcal{C}(x^*)$ cannot be tested since $\mathcal{C}(x^*)$ is now unknown. Nonetheless, we notice that given $\mathcal{X}$, we can construct the cones $\mathcal{C}(x)$ for each $x \in \mathcal{X}$ beforehand. Let $\hat{\mathcal{S}}(\mathbf{x}_n)$ be a high-probability confidence set such that for any $\mathbf{x}_n$, $\hat{\theta}_n$ is the center of $\hat{\mathcal{S}}(\mathbf{x}_n)$ and $\mathbb{P}(\theta^* \in \hat{\mathcal{S}}(\mathbf{x}_n)) \geq 1-\delta$. Notice that $\hat{\mathcal{S}}(\mathbf{x}_n)$ can be directly computed from samples and we can stop whenever there exists an x such that $\hat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$.

Lemma 3.3. *Let $\mathbf{x}_n = (x_1, \dots, x_n)$ be an arbitrary allocation sequence. If after n steps there exists an arm $x \in \mathcal{X}$ such that $\hat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$ then $\mathbb{P}\Big(\Pi(\hat{\theta}_n) \neq x^*\Big) \leq \delta$.*

Proof. The proof follows from the fact that if $\hat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$ and $\theta^* \in \hat{\mathcal{S}}(\mathbf{x}_n)$ with high probability, then $\theta^* \in \mathcal{C}(x)$ which implies that $\Pi(\hat{\theta}_n) = x = x^*$. □

Arm selection strategy. Similarly to the oracle algorithm, we should design an allocation strategy which guarantees that the (random) confidence set $\hat{\mathcal{S}}(\mathbf{x}_n)$ shrinks in one of the cones $\mathcal{C}(x)$ within the fewest number of steps. Let $\widehat{\Delta}_n(x, x') = (x - x')^\top \hat{\theta}_n$ be the empirical gap between arms x, x'. Then the stopping condition $\hat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$ can be written as

$$\begin{aligned} &\exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, \forall \theta \in \hat{\mathcal{S}}(\mathbf{x}_n), (x - x')^\top \theta \geq 0 \\ &\qquad \Leftrightarrow \quad \exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, \forall \theta \in \hat{\mathcal{S}}(\mathbf{x}_n), (x - x')^\top (\hat{\theta} - \theta) \leq \widehat{\Delta}_n(x, x'). \end{aligned} \tag{3.27}$$

This suggests that the empirical confidence set can be defined as

$$\hat{\mathcal{S}}(\mathbf{x}_n) = \left\{ \theta \in \mathbb{R}^d, \forall y \in \mathcal{Y}, y^\top (\hat{\theta}_n - \theta) \leq c ||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\frac{\log(K^2/\delta)}{c'}} \right\}. \tag{3.28}$$

Unlike $\mathcal{S}^*(\mathbf{x}_n)$, $\hat{\mathcal{S}}(\mathbf{x}_n)$ considers all directions $y \in \mathcal{Y}$. As a result, the stopping condition $\hat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$, using Eq. 3.27 and Eq. 3.28, can be reformulated as

$$\exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, c ||x - x'||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\frac{\log(K^2/\delta)}{c'}} \leq \widehat{\Delta}_n(x, x'). \tag{3.29}$$

Although similar to Eq. 3.21, unfortunately this condition cannot be directly used to derive an allocation strategy. In fact, it is considerably more difficult to define a suitable allocation strategy to fit a random confidence set $\hat{\mathcal{S}}(\mathbf{x}_n)$ into a cone $\mathcal{C}(x)$ for an x which is not known in advance. As a result, we do not know on which pairs (x, x') the allocation should focus. Also, if we try to make the accuracy $c||x - x'||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_{c'}(K^2/\delta)}$ smaller than the random empirical gaps $\widehat{\Delta}_n(x, x')$ for any $(x, x') \in \mathcal{X}^2$ (to meet the stopping condition in eq. 3.29), we would obtain a very poor allocation. This might happen for instance in the case when for some pairs of suboptimal arms (i.e., (x, x') with $x, x' \neq x^*$) the gap $\Delta(x, x')$ may be zero, thus forcing the allocation to waste resources in the attempt to discriminate between arms that are equivalent.

In the rest of the section, we propose two allocations that try to achieve the condition in Eq. 3.29 as fast as possible by implementing a static arm selection strategy, while avoiding the aforementioned problem. Then, we present a more sophisticated adaptive strategy in Sect. 5. The general structure of the static allocations strategies in summarized in Alg. 3.

Algorithm 3 Static allocation algorithms

Input: decision space $\mathcal{X} \in \mathbb{R}^d$, confidence $\delta > 0$
Set: $t = 0$; $Y = \{y = (x - x'); x \neq x' \in \mathcal{X}\}$;
while Eq. 3.29 is not true **do**
  if G-allocation **then**
    Select arm x_t (Eq. 3.32 or relaxation in Eq. 3.31):
      $x_t = \arg\min_{x \in X} \max_{x' \in X} x'^\top (A + xx^\top)^{-1} x'$
  else if $\mathcal{XY}$-allocation **then**
    Select arm x_t (Eq. 3.36 or relaxation in Eq. 3.35):
      $x_t = \arg\min_{x \in X} \max_{y \in Y} y^\top (A + xx^\top)^{-1} y$
  end if
  Update $\hat{\theta}_t = A_t^{-1} b_t$, $t = t + 1$
end while
Return arm $\Pi(\hat{\theta}_t)$

4.1 G-Allocation Strategy

The definition of the G-allocation strategy directly follows from the observation that for any pair $(x, x') \in \mathcal{X}^2$ we have that

$$||x - x'||_{A_{\mathbf{x}_n}^{-1}} \leq 2 \max_{x'' \in \mathcal{X}} ||x''||_{A_{\mathbf{x}_n}^{-1}}.$$

This suggests that an allocation minimizing $\max_{x \in \mathcal{X}} ||x||_{A_{\mathbf{x}_n}^{-1}}$ does reduce an upper bound on the quantity tested in the stopping condition in Eq. 3.29. If such upper bound is not too loose, then we expect the stopping condition to be triggered in relatively few steps. Building on this observation, for any fixed n, we define the G-allocation as

$$\mathbf{x}_n^G = \arg\min_{\mathbf{x}_n} \max_{x \in \mathcal{X}} ||x||_{A_{\mathbf{x}_n}^{-1}}. \tag{3.30}$$

We notice that this formulation coincides with the standard G-optimal design (hence the name of the allocation) defined in experimental design theory (see for instance, [Pukelsheim, 2006, Sect. 9.2]) to minimize the maximal mean-squared prediction error in linear regression. The G-allocation can be interpreted as the design that allows to estimate θ^* *uniformly well* over all the arms in $\mathcal{X}$. Hence, in the MAB setting where the arms are independent, G-allocation smoothly reduces to a uniform allocation over arms.

Notice that the G-allocation in Eq. 3.30 is well defined only for a fixed number of steps n and it cannot be directly implemented in our case, since n is unknown in advance. Therefore we have to resort to a more "incremental" implementation of the G-allocation. In the experimental design literature, a wide number of approximate solutions have been proposed to solve the *NP*-hard discrete optimization problem in Eq. 3.30 (see [Bouhtou et al., 2010, Burger and Osher, 2005] for some recent results). Here, we rely on a general structure of the efficient rounding procedure defined

in [Pukelsheim, 2006, Chapter 12] to implement a design $\lambda \in \mathcal{D}^K$ into an allocation $\mathbf{x}_n$ for any n. Briefly restated, the rounding procedure goes as follows:

Let $p = \text{supp}(\lambda)$ the support of λ, then we want to compute the number of pulls n_i (with $i = 1, \dots, p$) for all the arms in the support of λ, such that $\sum_{i=1}^K n_i = n$. Basically, the fast implementation of the design is obtained in two phases, as follows:

- In the first phase, given the sample size n and the number of support points p, we calculate their corresponding frequencies $n_i = \lceil (n - \frac{1}{2}p)\lambda_i \rceil$, where $n_1, n_2, \dots, n_p$ are positive integers with $\sum_{i \le p} n_i \ge n$.
- The second phase loops until the discrepancy $\left(\sum_{i \le p} n_i\right) - n$ is 0, either:
  - increasing a frequency n_j which attains $n_j/\lambda_j = \min_{i \le p}(n-1)/\lambda_i$ to $n_j + 1$, or
  - decreasing some n_k with $(n_k - 1)/\lambda_k = \max_{i \le p}(n_i - 1)/\lambda_i$ to $n - 1$.

4.1.1 Implementation of the G-allocation

A first option is to optimize a continuous relaxation of the problem and compute the optimal design. Let $\rho^G(\lambda) = \max_x x^\top \Lambda_\lambda^{-1} x$, the optimal design is

$$\lambda^G = \arg\min_{\lambda \in \mathcal{D}_k} \max_{x \in \mathcal{X}} ||x||^2_{\Lambda_\lambda^{-1}} = \arg\min_{\lambda \in \mathcal{D}_k} \rho^G(\lambda). \tag{3.31}$$

This is a convex optimization problem and it can be solved using the projected gradient algorithm, interior point techniques, or multiplicative algorithms (see e.g., [Yu, 2010]). To move from the design λ^G to a discrete allocation we use the efficient rounding technique presented above and we obtain that the resulting allocation $\mathbf{x}_t^{\tilde{G}}$ is guaranteed to be monotonic as the number of times an arm x is pulled is non-decreasing with t. Thus from $\mathbf{x}_t^{\tilde{G}}$ we obtain a simple incremental rule, where the arm x_t is the arm for which $\mathbf{x}_t^{\tilde{G}}$ recommends one pull more than in $\mathbf{x}_{t-1}^{\tilde{G}}$. An alternative is to directly implement an incremental version of Eq. 3.30 by selecting at each step t the greedy arm

$$\begin{aligned} x_t &= \arg\min_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left(A_{\mathbf{x}_{t-1}} + xx^\top\right)^{-1} x' \\ &= \arg\min_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left[A^{-1} - \frac{A^{-1}xx^\top A^{-1}}{1 + x^\top A^{-1} x}\right] x', \end{aligned} \tag{3.32}$$

where the second formulation follows from the matrix inversion lemma. This allocation is somehow simpler and more direct than using the continuous relaxation, but it comes with an efficiency loss which is constant over time as illustrated in Lemma 3.5.

We report the performance guarantees for the two implementations proposed above. The allocation $\mathbf{x}_t^{\tilde{G}}$ obtained applying the rounding procedure has the following performance guarantee.

Lemma 3.4. *For any $t \geq d$, the rounding procedure defined in [Pukelsheim, 2006, Chapter 12] returns an allocation $\mathbf{x}_t^{\tilde{G}}$, whose corresponding design $\lambda^{\tilde{G}} = \lambda_{\mathbf{x}_t^{\tilde{G}}}$ is such that*[5]

$$\rho^G(\lambda^{\tilde{G}}) \leq \Big(1 + \frac{d + d^2 + 2}{2t}\Big) d.$$

Proof of Lemma 3.4. The proof follows directly from Lemma 3.9, whose statement and proof are given in the appendix of this chapter (Sect. 8.1). □

On the other hand, the greedy incremental allocation achieves the following performance.

Lemma 3.5. *For any t, the greedy algorithm in Eq. 3.32 returns an allocation $\mathbf{x}_t^{\tilde{G}}$ whose corresponding design $\lambda^{\tilde{G}} = \lambda_{\mathbf{x}_t^{\tilde{G}}}$ is such that*

$$\rho^G(\lambda^{\tilde{G}}) \leq (1 + e^{-1})\Big(1 + \frac{d + d^2 + 2}{2t}\Big) d.$$

Proof of Lemma 3.5. The lemma directly follows from the remark that G-optimal design is equivalent to the D-optimal design (see Prop. 3.2), which can be defined as the maximization of a submodular function. As a result, from Theorem 2.6 in [Sagnol, 2013] we obtain that the greedy algorithm achieves a constant fraction loss of $(1 + e^{-1})$. □

Here, we use the rounding procedure defined in [Pukelsheim, 2006, Chapter 12], with a performance guarantee bounded by a factor $\beta = \Big(1 + \frac{d+d^2+2}{2t}\Big) d, \forall t \geq d$.

4.1.2 Sample complexity

For any approximate G-allocation strategy with performance no worse than a factor $(1+\beta)$ of the optimal strategy $\mathbf{x}_n^G$, we now introduce the bound on the sample complexity N^G.

Theorem 3.2. *If the G-allocation strategy is implemented with a β-approximate method (see lemmas 3.4 and 3.5 above) and the stopping condition in Eq. 3.29 is used, then*

$$\mathbb{P}\left[N^G \leq \frac{16c^2 d(1+\beta)\log(1/\delta)}{\Delta_{\min}^2} \wedge \Pi(\hat{\theta}_{N^G}) = x^*\right] \geq 1 - \delta. \tag{3.33}$$

Proof. The statement follows from Prop. 2.2 and the performance guarantees for the different implementations of the G-optimal design. By recalling the empirical stopping condition in Eq. 3.29 and the definition $\rho^G(\lambda) = \max_x x^\top \Lambda_\lambda^{-1} x$, we notice that from a simple triangle inequality applied to $||y||_{A^{-1}}$, a sufficient condition for stopping is that for any $x \in \mathcal{X}$

$$\frac{4c^2 \rho_n^{\tilde{G}} \log_{c'}(K^2/\delta)}{n} \leq \hat{\Delta}_n^2(x^*, x),$$

[5] We recall that from any allocation $\mathbf{x}_n$ the corresponding design $\lambda_{\mathbf{x}}$ is such that $\lambda_{\mathbf{x}_n}(x) = T_n(x)/n$.

where $\rho_n^{\tilde{G}} = \rho^G(\lambda_{\mathbf{x}_n^{\tilde{G}}})$ and $\mathbf{x}_n^{\tilde{G}}$ is the allocation obtained from rounding the optimal design λ^G obtained from the continuous relaxation or the greedy incremental algorithm. From Prop. 2.2 we have that the following inequalities hold with probability $1-\delta$:

$$\widehat{\Delta}_n(x^*,x) \geq \Delta(x^*,x) - c||x^*-x||_{A^{-1}_{\mathbf{x}_n^G}}\sqrt{\log(K^2/\delta)} \geq \Delta(x^*,x) - 2c\sqrt{\frac{\rho_n^{\tilde{G}}\log_{c'}(K^2/\delta)}{n}}.$$

Combining this with the previous condition and since the condition must hold for all $x \in \mathcal{X}$, we have that a sufficient condition to stop using the G-allocation is

$$\frac{16c^2\rho_n^{\tilde{G}}\log_{c'}(K^2/\delta)}{n} \leq \Delta_{\min},$$

which defines the level of accuracy that the G-allocation needs to achieve before stopping. Since $\rho_n^{\tilde{G}} \leq (1+\beta)d$ then the statement follows by inverting the previous inequality. □

Notice that this result matches (up to constants) the worst-case value of N^* given the upper bound on H_{LB}. This means that, although completely static, the G-allocation is already worst-case optimal.

4.2 $\mathcal{XY}$-Allocation Strategy

Despite being worst-case optimal, G-allocation is minimizing a rather loose upper bound on the quantity used to test the stopping criterion. Thus, we define an alternative static allocation that targets the stopping condition in Eq. 3.29 more directly by reducing its left-hand-side for any possible direction in $\mathcal{Y}$. For any fixed n, we define the $\mathcal{XY}$-allocation as

$$\mathbf{x}_n^{\mathcal{XY}} = \arg\min_{\mathbf{x}_n}\max_{y\in\mathcal{Y}}||y||_{A^{-1}_{\mathbf{x}_n}}. \tag{3.34}$$

$\mathcal{XY}$-allocation is based on the observation that the stopping condition in Eq. 3.29 requires only the empirical gaps $\widehat{\Delta}(x,x')$ to be well estimated, hence arms are pulled with the objective of increasing the accuracy of directions in $\mathcal{Y}$ instead of arms $\mathcal{X}$. This problem can be seen as a transductive variant of the G-optimal design [Yu et al., 2006], where the target vectors $\mathcal{Y}$ are different from the vectors $\mathcal{X}$ used in the design.

4.2.1 Implementation of the $\mathcal{XY}$-allocation

Notice that the complexity of the $\mathcal{XY}$-allocation trivially follows from the complexity of the G-allocation and is NP-hard. As a result, we need to propose approximate solutions to compute an allocation $\mathbf{x}_n^{\widetilde{\mathcal{XY}}}$, as was the case for the G-allocation. Let $\rho^{\mathcal{XY}}(\lambda) = \max_{y\in\mathcal{Y}} y^\top\Lambda_\lambda^{-1}y$, then the first option is the compute the optimal solution to the continuous relaxed problem

$$\lambda^{\mathcal{XY}} = \arg\min_{\lambda\in\mathcal{D}_k}\max_{y\in\mathcal{Y}}||y||^2_{\Lambda_\lambda^{-1}} = \arg\min_{\lambda\in\mathcal{D}_k}\rho^{\mathcal{XY}}(\lambda). \tag{3.35}$$

And then compute the corresponding discrete allocation $\mathbf{x}_n^{\widetilde{\mathcal{XY}}}$ using the efficient rounding procedure. Alternatively, we can use an incremental greedy algorithm which at each step t returns the arm

$$x_t = \arg\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} y^\top \left(A_{\mathbf{x}_{t-1}} + xx^\top\right)^{-1} y. \tag{3.36}$$

Lemma 3.6. *For any $t \geq d$, the rounding procedure defined in [Pukelsheim, 2006, Chapter 12] returns an allocation $\mathbf{x}_t^{\widetilde{\mathcal{XY}}}$, whose corresponding design $\lambda^{\widetilde{\mathcal{XY}}} = \lambda_{\mathbf{x}_t^{\widetilde{\mathcal{XY}}}}$ is such that*

$$\rho^{\mathcal{XY}}(\lambda^{\tilde{\mathcal{XY}}}) \leq 2\left(1 + \frac{d + d^2 + 2}{2t}\right)d.$$

Proof of Lemma 3.6. The proof follows from the fact that for any pair (x, x')

$$||x - x'||_{A_{\mathbf{x}_n}^{-1}} \leq 2 \max_{x''\in\mathcal{X}} ||x''||_{A_{\mathbf{x}_n}^{-1}}. \tag{3.37}$$

Then the proof proceeds as in Lemma 3.4. □

On the other hand, the greedy incremental allocation achieves the following performance.

Lemma 3.7. *For any t, the greedy algorithm in Eq. 3.32 returns an allocation $\mathbf{x}_t^{\tilde{G}}$ whose corresponding design $\lambda^{\tilde{G}} = \lambda_{\mathbf{x}_t^{\tilde{G}}}$ is such that*

$$\rho^{G}(\lambda^{\tilde{G}}) \leq (1 + e^{-1})\left(1 + \frac{d + d^2 + 2}{2t}\right)d.$$

Proof of Lemma 3.7. As it was the case for the proof of Lemma 3.6, we are going to rely on the corresponding results shown for the G-allocation. Thus, after applying the inequality in Eq. 3.37, we proceed as in Lemma 3.5. □

4.2.2 Sample complexity

The sample complexity of the $\mathcal{XY}$-allocation is as follows.

Theorem 3.3. *If the $\mathcal{XY}$-allocation strategy is implemented with a β-approximate method (see lemmas 3.4 and 3.5 above) and the stopping condition in Eq. 3.29 is used, then*

$$\mathbb{P}\left[N^{\mathcal{XY}} \leq \frac{32c^2 d(1+\beta)\log(1/\delta)}{\Delta_{\min}^2} \wedge \Pi(\hat{\theta}_{N^{\mathcal{XY}}}) = x^*\right] \geq 1 - \delta. \tag{3.38}$$

Proof. We follow the same steps as in the proof of Theorem 3.2. □

Although the previous bound suggests that $\mathcal{XY}$ achieves a performance comparable to the G-allocation, we now provide some simple examples when $\mathcal{XY}$ may be arbitrarily better than G-allocation.

4.3 Comparison between G-allocation and $\mathcal{XY}$-allocation

To better illustrate the difference between the G and the $\mathcal{XY}$ allocation strategies, let us consider the following two examples presenting the behavior of the strategies.

Example 1. Consider a simple problem with $\mathcal{X} \subset \mathbb{R}^2$ and arms $x_1 = [1 \;\; \epsilon/2]^\top$ and $x_2 = [1 \;\; -\epsilon/2]^\top$, where $\epsilon \in (0, 1)$. In this case, both static allocations pull the two arms the same number of times, thus inducing an optimal design $\lambda(x_1) = \lambda(x_2) = 1/2$. We want to study the (asymptotic) performance of the allocation according to the different definition of error $\max_{x \in \mathcal{X}} x^\top \Lambda_\lambda^{-1} x$ and $\max_{y \in \mathcal{Y}} y^\top \Lambda_\lambda^{-1} y$ used by G and $\mathcal{XY}$-allocation respectively. We first notice that

$$\Lambda_\lambda = \frac{1}{2}\begin{bmatrix} 1 & \epsilon/2 \\ \epsilon/2 & \epsilon^2/4 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 1 & -\epsilon/2 \\ -\epsilon/2 & \epsilon^2/4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon^2/4 \end{bmatrix}.$$

As a result, for both x_1 and x_2 we have

$$[1 \;\; \epsilon/2]\Lambda_\lambda^{-1}\begin{bmatrix} 1 \\ \epsilon/2 \end{bmatrix} = [1 \;\; \epsilon/2]\begin{bmatrix} 1 & 0 \\ 0 & 4/\epsilon^2 \end{bmatrix}\begin{bmatrix} 1 \\ \epsilon/2 \end{bmatrix} = 2.$$

On the other hand, if we consider the direction $y = x_1 - x_2 = [0 \;\; \epsilon]$, we have

$$[0 \;\; \epsilon]\Lambda_\lambda^{-1}\begin{bmatrix} 0 \\ \epsilon \end{bmatrix} = [0 \;\; \epsilon]\begin{bmatrix} 1 & 0 \\ 0 & 4/\epsilon^2 \end{bmatrix}\begin{bmatrix} 0 \\ \epsilon \end{bmatrix} = 4.$$

This example shows a scenario where the performance achieved by $\mathcal{XY}$ may indeed be similar to the performance of G-optimal.

Example 2. Now let us consider a different setting where the two arms are aligned on the same axis and have features: $x_1 = [1 \;\; 0]$ and $x_2 = [1 - \epsilon \;\; 0]$. In this case, the problem reduces to a 1-dimensional problem and both strategies would concentrate their allocation to $x_1 = [1 \;\; 0]$ since it is the arm with larger norm and it may provide a better estimate of θ^*. As a result, the G-allocation would have a performance of 1, while the $\mathcal{XY}$-allocation over the direction $[\epsilon \;\; 0]$ would have a performance ϵ^2, which can be arbitrarily smaller than 1.

These examples suggest that the $\mathcal{XY}$ allocation, by aiming to minimize the error on the distances y rather than separately on each arm, manages to exploit better the structure in the environment, compared to G-optimal. While the G allocation always considers each arm separately, when the input set allows it, we can expect $\mathcal{XY}$ to consume less samples on the estimation of dominated arms, and thus to need a smaller sample complexity before identifying the best arm.

5 Adaptive Allocation

Although both G- and $\mathcal{XY}$-allocation are sound since they minimize upper-bounds on the quantities used by the stopping condition (Eq. 3.29), they may be very suboptimal

with respect to the ideal performance of the oracle introduced in Sec. 3.2. Typically, an improvement can be obtained by moving to strategies adapting on the rewards observed over time. Nonetheless, as reported in Prop. 2.3, whenever $\mathbf{x}_n$ is not a fixed sequence, the bound in Eq. 2.11 should be used. As a result, a factor $\sqrt{d}$ would appear in the definition of the confidence sets and in the stopping condition. This directly implies that the sample complexity of a fully adaptive strategy would scale linearly with the dimensionality d of the problem, thus removing any advantage with respect to static allocations. In fact, the sample complexity of G- and $\mathcal{XY}$-allocation already scales linearly with d and from Lemma 3.2 we cannot expect to improve the dependency on $\Delta_{\min}$. Thus, to design an effective allocation strategy, on the one hand we need to use the tighter bounds in Eq. 2.10 and, on the other hand, we need to be adaptive with respect to samples. In the following section we propose a phased algorithm which successfully meets both requirements. This algorithm, called $\mathcal{XY}$-Adaptive, uses a static allocation within each phase, but *learns* in-between the phases which are the important directions to sample, based on the samples observed at the previous phases.

5.1 $\mathcal{XY}$-Adaptive Strategy

The ideal case would be to define an empirical version of the oracle allocation in Eq. 3.22 so as to adjust the accuracy of the prediction only on the directions of interest $\mathcal{Y}^*$ and according to their gaps $\Delta(y)$. As discussed in Sect. 4, this cannot be obtained by a direct adaptation of Eq. 3.29. We now describe a safe alternative to adjust the allocation strategy to the gaps.

Lemma 3.8. *Let $\mathbf{x}_n$ be a fixed allocation sequence and $\hat{\theta}_n$ its corresponding estimate for θ^*. If an arm $x \in \mathcal{X}$ is such that*

$$\exists x' \in \mathcal{X} \text{ s.t. } c||x' - x||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\frac{\log(K^2/\delta)}{c'}} < \widehat{\Delta}_n(x', x), \tag{3.39}$$

*then arm x is **sub-optimal**.*

Proof. Let $y = x' - x$. Using the definition of $\hat{\mathcal{S}}(\mathbf{x}_n)$ in Eq. 3.28, and the fact that $\theta^* \in \hat{\mathcal{S}}(\mathbf{x}_n)$, then with high probability, we have

$$(x' - x)^\top (\hat{\theta}_n - \theta^*) \leq c||x' - x||_{A_{\mathbf{x}}^{-1}} \sqrt{\frac{\log(K^2/\delta)}{c'}}.$$

Whenever the condition in Eq. 3.39 is fulfilled, it holds true that

$$(x' - x)^\top (\hat{\theta}_n - \theta^*) \leq c||x' - x||_{A_{\mathbf{x}}^{-1}} \sqrt{\frac{\log(K^2/\delta)}{c'}} < \widehat{\Delta}_n(x', x).$$

Then, by subtracting $\widehat{\Delta}_n(x', x) = (x' - x)^\top \hat{\theta}_n$ from both sides, we obtain

$$-(x' - x)^\top \theta^* < 0 \Leftrightarrow x^\top \theta^* < x'^\top \theta^*.$$

Since with high probability the (true) value of x is smaller that to the value of x', it follows that x' dominated x, and thus x cannot be the optimal arm. □

Whenever Eq. 3.39 is true for a pair (x', x), we say that x' **dominates** x. Lemma 3.8 allows to easily construct the set of potentially optimal arms, denoted $\hat{\mathcal{X}}(\mathbf{x}_n)$, by removing from $\mathcal{X}$ all the dominated arms. As a result, we can replace the stopping condition in Eq. 3.29, by just testing whether the number of non-dominated arms is equal to 1 (that is, $|\hat{\mathcal{X}}(\mathbf{x}_n)| = 1$), which corresponds to the case where the confidence set is fully contained into a single cone. Moreover, relying on $\hat{\mathcal{X}}(\mathbf{x}_n)$, we construct $\hat{\mathcal{Y}}(\mathbf{x}_n) = \{y = x - x'; x, x' \in \hat{\mathcal{X}}(\mathbf{x}_n)\}$, the set of directions along which the estimation of θ^* needs to be improved to further shrink $\hat{\mathcal{S}}(\mathbf{x}_n)$ into a single cone and trigger the stopping condition.

Note that if $\mathbf{x}_n$ was selected by an adaptive strategy, then we could not use Lemma 3.8 to discard arms. Instead, we should rely on the bound in Prop. 2.3, which also contains the dimensionality term. To avoid this problem, an effective solution is to run the algorithm through **phases**. Let $j \in \mathbb{N}$ be the index of a phase and n_j its corresponding length. Also, we denote by $\hat{\mathcal{X}}_j$ the set of non-dominated arms constructed on the basis of the samples collected in the phase $j-1$. This set is used to identify the directions $\hat{\mathcal{Y}}_j$ and to define a *static* allocation which focuses on reducing the uncertainty of θ^* along the directions in $\hat{\mathcal{Y}}_j$. Formally, in phase j we implement the allocation

$$\mathbf{x}_{n_j}^j = \arg\min_{\mathbf{x}_{n_j}} \max_{y \in \hat{\mathcal{Y}}_j} ||y||_{A_{\mathbf{x}_{n_j}}^{-1}}, \tag{3.40}$$

which coincides with a $\mathcal{XY}$-allocation (see Eq. 3.34) but restricted on $\hat{\mathcal{Y}}_j$. Notice that $\mathbf{x}_{n_j}^j$ may still use any arm in $\mathcal{X}$ which could be useful in reducing the confidence set along any of the directions in $\hat{\mathcal{Y}}_j$. Once phase j is over, the OLS estimate $\hat{\theta}^j$ is computed using the rewards observed within phase j and then is used to test the stopping condition in Eq. 3.29. Whenever the stopping condition does not hold, a new set $\hat{\mathcal{X}}_{j+1}$ is constructed using the discarding condition in Lemma 3.8 and a new phase is started.

Notice that through this process, since only the samples from phase j are used in computing the OLS estimate, then at each phase j, the allocation $\mathbf{x}_{n_j}^j$ is static conditioned on the previous allocations. Therefore, the use of the bound from Prop. 2.2 is correct.

5.2 Length of the phases

A crucial aspect of this algorithm is the length of the phases n_j. On the one hand, short phases allow a high rate of adaptivity, since $\hat{\mathcal{X}}_j$ is recomputed very often. On the other hand, if a phase is too short, it is very unlikely that the estimate $\hat{\theta}^j$ may be accurate enough to actually discard any arm. We notice that ideally a phase should terminate as soon as an arm is discarded from $\hat{\mathcal{X}}_j$, that is, as soon as the uncertainty in estimating the value of an arm falls below its corresponding gaps (see Eq. 3.39). Nonetheless, this would again make the length of the allocation *adaptive* with respect to the samples.

An effective way to define the length of a phase in a deterministic way is to relate it to the actual uncertainty of the allocation in estimating the value of all the active

directions in $\hat{\mathcal{Y}}_j$. In phase j, for any allocation $\mathbf{x}_n$, let $\rho^j(\lambda) = \max_{y\in\hat{\mathcal{Y}}_j} ||y||^2_{\Lambda_\lambda^{-1}}$. Then, given a parameter $\alpha \in (0,1)$, we define

$$n_j = \min\Big\{n \in \mathbb{N} : \rho^j(\lambda_{\mathbf{x}_n^j})/n \leq \alpha\rho^{j-1}(\lambda^{j-1})/n_{j-1}\Big\}, \tag{3.41}$$

where $\mathbf{x}_n^j$ is the allocation defined in Eq. 3.40 and λ^{j-1} is the design corresponding to $\mathbf{x}_{n_{j-1}}^{j-1}$, the allocation performed at phase $j-1$. In words, we define n_j as the minimum number of steps needed by the $\mathcal{XY}$-Adaptive allocation to achieve an uncertainty over all the directions of interest which is a fraction α of the performance obtained in the previous iteration. Notice that given $\hat{\mathcal{Y}}_j$ and ρ^{j-1} this quantity can be computed before the actual beginning of phase j. The resulting algorithm using the $\mathcal{XY}$-Adaptive allocation strategy is summarized in Alg. 4.

Algorithm 4 $\mathcal{XY}$-Adaptive Allocation

Input: decision space $\mathcal{X}\in\mathbb{R}^d$; parameter α; confidence δ
Set $j=1$; $\hat{\mathcal{X}}_j=\mathcal{X}$; $\hat{\mathcal{Y}}_1=\mathcal{Y}$; $\rho_0=1$; $n_0=d(d+1)+1$
while $|\hat{\mathcal{X}}_j| > 1$ **do**
  $\rho^j = \rho^{j-1}$
  $t = 1; A_0 = I$
  while $\rho^j/t \geq \alpha\rho^{j-1}(\mathbf{x}_{n_{j-1}}^{j-1})/n_{j-1}$ **do**
    Select arm x_t (Eq. 3.32 or relaxation in Eq. 3.31)
      $x_t = \arg\min_{x\in X}\max_{y\in Y_j} y^\top(A+xx^\top)^{-1}y$
    Update $A_t = A_{t-1} + x_t x_t^\top$, $t = t+1$
    $\rho^j = \max_{y\in\hat{\mathcal{Y}}_j} y^\top A_t^{-1} y$
  end while
  Compute $b = \sum_{s=1}^t x_s r_s$; $\hat{\theta}_j = A_t^{-1}b$
  $\hat{\mathcal{X}}_{j+1} = \mathcal{X}$
  for $x \in \mathcal{X}$ **do**
    if $\exists x' : ||x-x'||_{A_t^{-1}}\sqrt{\log_{c'}(K^2/\delta)} \leq \widehat{\Delta}_j(x',x)$ **then**
      $\hat{\mathcal{X}}_{j+1} = \hat{\mathcal{X}}_{j+1} - \{x\}$
    end if
  end for
  $\hat{\mathcal{Y}}_{j+1} = \{y = (x-x'); x, x' \in \hat{\mathcal{X}}_{j+1}\}$
end while
Return $\Pi(\hat{\theta}_j)$

5.3 Discarding uninteresting directions: M*

Although the $\mathcal{XY}$-Adaptive allocation strategy is designed to approach the oracle sample complexity N^*, in early phases it basically implements a $\mathcal{XY}$-allocation and no significant improvement can be expected until some directions are discarded from $\hat{\mathcal{Y}}$. At that point, $\mathcal{XY}$-Adaptive starts focusing on directions which only contain near-optimal arms and it starts approaching the behavior of the oracle. As a result, in

studying the sample complexity of $\mathcal{XY}$-Adaptive we have to take into consideration the unavoidable price to pay to discard "suboptimal" directions. This cost is directly related to the geometry of the arm space that influences the number of samples needed before arms can be discarded from $\hat{\mathcal{X}}$. To take into account this problem-dependent quantity, we introduce a slightly relaxed definition of complexity which will serve as basis for the theoretical analysis of the $\mathcal{XY}$-Adaptive allocation. More precisely, we define the number of steps needed to discard all the directions which do not contain x^*, i.e. $\mathcal{Y} - \mathcal{Y}^*$. From a geometrical point of view, this corresponds to the case when for any pair of suboptimal arms (x, x'), the confidence set $\mathcal{S}^*(\mathbf{x}_n)$ does not intersect the hyperplane separating the cones $\mathcal{C}(x)$ and $\mathcal{C}(x')$. Fig. 3.2 offers a simple illustration for such a situation: $\mathcal{S}^*$ no longer intercepts the border line between $\mathcal{C}(x_2)$ and $\mathcal{C}(x_3)$, which implies that direction $x_2 - x_3$ can be discarded.

More formally, the hyperplane containing parameters θ for which x and x' are equivalent is simply $\mathcal{C}(x) \cap \mathcal{C}(x')$ and the quantity

$$M^* = \inf\{n \in \mathbb{N}, \forall x \neq x^*, \forall x' \neq x^*, \mathcal{S}^*(\mathbf{x}_n^{\mathcal{XY}}) \cap (\mathcal{C}(x) \cap \mathcal{C}(x')) = \emptyset\}, \tag{3.42}$$

corresponds to the minimum number of steps needed by the static $\mathcal{XY}$-allocation strategy to discard all the *suboptimal* directions. This term together with the oracle complexity N^* characterizes the sample complexity of the phases of the $\mathcal{XY}$-Adaptive allocation. In fact, the length of the phases is such that either they correspond to the complexity of the oracle or they can never last more than the steps needed to discard all the sub-optimal directions.

Based on the previously introduces quantities, we can now introduce the overall sample complexity of the $\mathcal{XY}$-Adaptive allocation.

5.4 Sample Complexity

The sample complexity of the $\mathcal{XY}$-Adaptive allocation is bounded as shown in the following theorem (the proof is given in the appendix in Sect. 8.2).

Theorem 3.4. *If the $\mathcal{XY}$-Adaptive allocation strategy is implemented with a β-approximate method and the stopping condition in Eq. 3.29 is used, then*

$$\mathbb{P}\left[N \leq \frac{(1+\beta)\max\{M^*, \frac{16}{\alpha}N^*\}}{\log(1/\alpha)} \log\left(\frac{c\sqrt{\log_{c'}(K^2/\delta)}}{\Delta_{\min}}\right) \wedge \Pi(\hat{\theta}_N) = x^*\right] \geq 1 - \delta. \tag{3.43}$$

We first remark that, unlike G and $\mathcal{XY}$, the sample complexity of $\mathcal{XY}$-Adaptive does not have any direct dependency on d and $\Delta_{\min}$ (except in the logarithmic term) but it rather scales with the oracle complexity N^* and the cost of discarding suboptimal directions M^*. Although this additional cost is probably unavoidable, one may have expected that $\mathcal{XY}$-Adaptive may need to discard all the suboptimal directions before performing as well as the oracle, thus having a sample complexity of $O(M^* + N^*)$. On the other hand, we notice that N scales with the *maximum* of M^* and N^*, thus implying that $\mathcal{XY}$-Adaptive may actually catch up with the performance of the oracle

(with only a multiplicative factor of $16/\alpha$) whenever discarding suboptimal directions is less expensive than actually identifying the best arm. In Sect. 8.3 we further discuss the importance of M^* in the sample complexity of the $\mathcal{XY}$-Adaptive allocation.

6 Numerical Simulations

We illustrate the performance of $\mathcal{XY}$-Adaptive and compare it to the $\mathcal{XY}$-Oracle strategy (Eq. 3.22), the static allocations $\mathcal{XY}$ and G, as well as with the Fully-adaptive version of $\mathcal{XY}$ where $\hat{\mathcal{X}}$ is updated at each round and the bound from Prop.2.3 is used. For a fixed confidence δ, we compare the sampling budget needed to identify the best arm with probability at least $1-\delta$. We choose a setting where the minimum gap, $\Delta_{\min}$, is much smaller than the other gaps. This allows to illustrate the advantage of the allocation strategies $\mathcal{XY}$-Oracle and $\mathcal{XY}$-Adaptive which, unlike the static allocations, adapt to the characteristics of the problem (notably the different gaps). In particular, we illustrate how the difference in the sampling budget changes with the dimensionality of the input space.

Setting. We consider a set of arms $\mathcal{X} \in \mathbb{R}^d$, with $|\mathcal{X}| = d+1$ including the canonical basis $(e_1, \dots, e_d)$ and an additional arm $x_{\mathrm{d}+1} = [\cos(\omega) \;\; \sin(\omega) \;\; 0 \;\; \dots \;\; 0]^\top$. We choose $\theta^* = [2 \;\; 0 \;\; 0 \;\; \dots \;\; 0]^\top$ and fix $\omega = 0.01$, so that $\Delta_{\min} = (x_1 - x_{\mathrm{d}+1})^\top \theta^*$. In this setting, an efficient sampling strategy should focus on reducing the uncertainty in the direction $\tilde{y} = (x_1 - x_{\mathrm{d}+1})$ by pulling the arm $x_2 = e_2$ which is almost aligned with $\tilde{y}$. In fact, from the rewards obtained from x_2 it is easier to decrease the uncertainty about the second component of θ^*, that is precisely the dimension which allows to discriminate between x_1 and $x_{\mathrm{d}+1}$. Also, we fix the confidence level $\delta = 0.05$, $\alpha = 1/10$, and the noise $\varepsilon \sim \mathcal{N}(0,1)$. Each phase begins with an initialization matrix A_0, obtained by pulling once each canonical arm. In Fig. 3.3 we report the sampling budget of the algorithms, averaged over 100 runs, for $d = 2, \dots, 10$.

Results. The numerical results show that $\mathcal{XY}$-Adaptive is effective in allocating the samples to shrink the uncertainty in the direction $\tilde{y}$. Indeed, $\mathcal{XY}$-Adaptive identifies the most important direction after few early phases and is able to perform an allocation which mimics that of the oracle. On the contrary, $\mathcal{XY}$ and G do not adjust according to the empirical gaps and consider all directions as equally important. This behavior forces $\mathcal{XY}$ and G to allocate samples until the uncertainty is smaller than $\Delta_{\min}$ in all directions. Even though the Fully-adaptive algorithm also identifies the most informative direction rapidly, the $\sqrt{d}$ term in the bound delays the discarding of the arms and prevents the algorithm from gaining any advantage compared to $\mathcal{XY}$ and G.

As shown in Fig. 3.3, the difference between the budget of $\mathcal{XY}$-Adaptive and the static strategies increases with the number of dimensions. In fact, while additional dimensions has little to no impact on $\mathcal{XY}$-Oracle and $\mathcal{XY}$-Adaptive (the only important direction remains $\tilde{y}$ independently from the number of unknown features of θ^*), for the

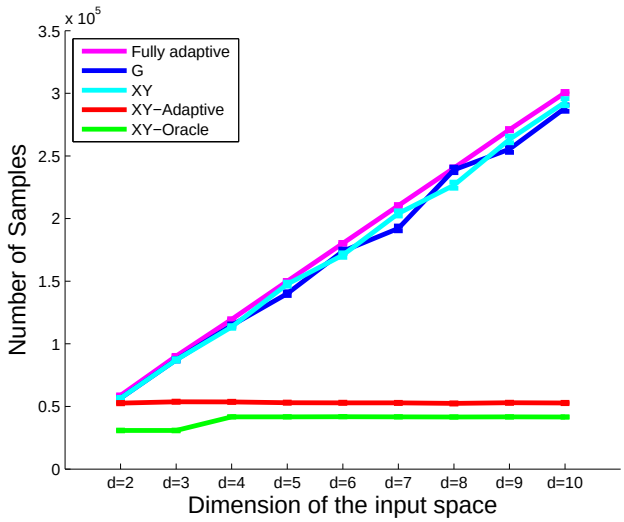


Figure 3.3: The sampling budget needed to identify the best arm, when the dimension grows from $\mathbb{R}^2$ to $\mathbb{R}^{10}$.

static allocations more dimensions implies more directions to be considered and more features of θ^* to be estimated uniformly well until the uncertainty falls below $\Delta_{\min}$.

Samples/arm	$\mathcal{XY}$-*Oracle*	$\mathcal{XY}$-*Adaptive*	$\mathcal{XY}$	G	*Fully-adaptive*
x_1	207	263	29523	28014	740
x_2	41440	52713	29524	28015	149220
x_3	2	3	29524	28015	1
x_4	2	5	29524	28015	1
x_5	1	2	29524	28015	1
x_6	0	2	1	1	1
Budget	41652	52988	147620	140075	149964

Figure 3.4: The budget needed by the allocation strategies to identify the best arm when $\mathcal{X} \subseteq \mathbb{R}^5$ and their sample allocation over arms. $\mathcal{XY}$ and G allocate samples uniformly over the canonical arms while $\mathcal{XY}$-oracle and $\mathcal{XY}$-adaptive use most of the samples for the most informative direction.

In Fig. 3.4 we can notice that even though the Fully-adaptive algorithm identifies the most informative direction and focuses the sampling on arm x_2, its sample complexity still has a growth linear in the dimension, due to the additional dimensionality term in the bound. Consequently, the advantage over the static strategies is canceled. On the other hand, $\mathcal{XY}$-Adaptive "learns" the gaps from the observations and allocates the samples very similarly to $\mathcal{XY}$-Oracle, without suffering a large loss in terms of the sampling budget. However, $\mathcal{XY}$-Adaptive's sample complexity has to account for the re-initializations made at the beginning of a new phase. Finally, we notice that in this problem the static allocations, $\mathcal{XY}$ and G, perform a uniform allocation over the canonical arms. Another interesting remark is that the number of pulls to one

canonical arm is smaller than the samples that $\mathcal{XY}$-Oracle allocated to x_2. This is explained by the "mutual information" coming from the multiple observations on all directions, which helps in reducing the overall uncertainty of the confidence set.

Remark (comparison to static algorithms). According to the theoretical analysis, the performance of $\mathcal{XY}$-Adaptive is better than static strategies in a much wider range of cases than the one considered in our experiment. Intuitively, as soon as $\mathcal{XY}$-Adaptive discards arms/directions, its sample complexity becomes smaller than for a static allocation that sticks to a fixed set of directions. In fact, $\mathcal{XY}$-Adaptive does actually perform the $\mathcal{XY}$-static strategy at the beginning but then it progressively discards arms/directions and focuses on improving the estimates of the gaps only for the remaining arms. It is intuitive that the fewer the "active" arms/directions, the fewer the samples needed to find the optimal arm. Thus, apart from an additional factor $1/\log(1/\alpha)$ due to the re-initializations in each phase, $\mathcal{XY}$-Adaptive cannot be worse than static strategies and is superior whenever at least one direction can be discarded. On the other hand, as we have seen in the example above, the improvement of $\mathcal{XY}$-Adaptive with respect to static strategies can be up to a factor of d.

Remark (comparison to previous BAI algorithms). In our experiment, although the setting is almost a MAB, there is already a huge gap in the performance of $\mathcal{XY}$-Adaptive and that of algorithms for best-arm identification in MAB (for instance, Successive-Elimination(SE)). In fact, similar to $\mathcal{XY}$-Adaptive, SE quickly discards all arms except x_1 and x_{d+1}. Then, while $\mathcal{XY}$-Adaptive exploits the structure of the problem and pulls x_2 (thus improving the estimate of θ^* along the dimension that better discriminates between x_1 and x_{d+1}), SE keeps pulling the remaining (near optimal) arms until it can confidently identify the best arm. This results in a much higher sample complexity for SE (order of 10^8 for $d = 2$), even worse than static strategies.

Remark (reaching the complexity bounds). The upper bound is achieved in a d-arm MAB problem (i.e., orthogonal arms) where all arms have the same gap. In this case H_{LB} is $O(d/\Delta_{\min}^2)$. As for the lower bound, an example is provided in the experiment where $\Delta_{\min}$ is very small, all other arms have large gaps and can be easily discarded, and one arm (i.e., x_2) is almost aligned with the direction connecting optimal and second-optimal arms. In this case H_{LB} is $O(1/\Delta_{\min}^2)$.

7 Conclusions

In this chapter we studied the problem of best-arm identification in the linear bandit setting. First, we characterized the problem-dependent complexity of the best arm identification task and showed its connection with the complexity in the MAB setting. Then, we designed and analyzed efficient sampling strategies for this problem. The G-allocation strategy allowed us to point out a close connection with optimal experimental design techniques, and in particular to the G-optimality criterion. Through the second proposed strategy, $\mathcal{XY}$-allocation, we introduced a novel optimal design problem where the testing arms do not coincide with the arms chosen in the design. Lastly, we pointed out the limits that a fully-adaptive allocation strategy might have in the linear bandit

setting and proposed a phased-algorithm, $\mathcal{XY}$-Adaptive, that learns from previous observations, without suffering from the dimensionality of the problem.

This work opens the way for an important number of similar problems of pure-exploration problems in the linear-bandit setting, already analyzed in the MAB setting. For instance, we can investigate strategies to identify the best-linear arm when having a limited budget, study the best-arm identification when the set of arms is very large (or infinite), and the multi-bandit BAI identification. In addition, notice that this problem can also be seen as the online optimization of a linear stochastic function. The particularity pointed out by the strategies that we propose will also come in this case from the fact that an efficient strategy might need to allocate samples outside the vicinity of the optimal point in order to get the needed information about the near-optimal arms.

Lastly, some interesting extensions also emerge from the optimal experimental design (OED) literature, such as the study of sampling strategies for meeting the G-optimality criterion when the noise is heteroscedastic, or the design of efficient strategies for satisfying other related optimality criteria, such as V-optimality. We dedicate the next chapter to a more detailed study of these connections with the OED criteria.

8 Appendix

8.1 Tools

First we state without proof the Equivalence Theorem introduced in [Kiefer and Wolfowitz, 1960].

Proposition 3.2 ([Kiefer and Wolfowitz, 1960]). *Define $f(x;\xi) = x^\top M(\xi)^{-1}x$, where $M(\xi)$ is a $d \times d$ non-singular matrix and x is a column vector in $\mathbb{R}^d$. We consider two extremum problems.*

The first is to choose ξ so that

$$(1) \quad \xi \quad \textit{maximizes} \quad \det M(\xi) \qquad (\textit{D-optimal design})$$

The second one is to choose ξ so that

$$(2) \quad \xi \quad \textit{minimizes} \quad \max f(x;\xi) \qquad (\textit{G-optimal design})$$

We note that the integral with respect to ξ of $f(x;\xi)$ is d; hence, $\max f(x;\xi) \geq d$, and thus a sufficient condition for ξ to satisfy (2) is

$$(3) \quad \max f(x;\xi) = d.$$

Statements (1), (2) and (3) are equivalent.

We now introduce a technical lemma useful for the study of the performance guarantees for the two implementations we proposed. Although the lemma is presented for a specific definition of uncertainty ρ, any other notion including design matrices of the kind Λ_λ will satisfy the same guarantee.

Lemma 3.9. *Let $\rho(\lambda) = \max_{x\in\mathcal{X}} x^\top \Lambda_\lambda^{-1} x$ be a measure of uncertainty of interest for any design $\lambda \in \mathcal{D}^K$. We denote by $\lambda^* = \arg\min_{\lambda\in\mathcal{D}^K} \rho(\lambda)$ the optimal design and for any $n > d$ we introduce the optimal discrete allocation as*

$$\mathbf{x}_n^* = \arg \min_{\mathbf{x}_n \in \mathcal{X}^n} \max_{x\in\mathcal{X}} \frac{x^\top \Lambda_{\lambda_{\mathbf{x}_n}}^{-1} x}{n},$$

where $\lambda_{\mathbf{x}_n}$ is the (fractional) design corresponding to $\mathbf{x}_n$. Then we have

$$\rho(\lambda^*) \leq \rho(\mathbf{x}_n^*) \leq \left(1 + \frac{p}{n}\right)\rho(\lambda^*), \tag{3.44}$$

where $p = supp(\lambda^*)$ is the number of points in the support of λ^*. If d linearly independent arms are available in $\mathcal{X}$, then we can upper bound the size of the support of λ^* and obtain

$$\rho(\lambda^*) \leq \rho(\mathbf{x}_n^*) \leq \left(1 + \frac{d(d+1)}{n}\right)\rho(\lambda^*). \tag{3.45}$$

Proof. The first part of the statement follows by the definition of λ^* as the minimizer of ρ. Let $\tilde{\mathbf{x}}_n$ be an efficient rounding technique applied on λ^* such as the one described in Lemma 12.8 in [Pukelsheim, 2006]. Then $\tilde{\mathbf{x}}_n$ has the same support as λ^* and an efficiency loss bounded by p/n. As a result, we have

$$\rho(\mathbf{x}_n^*) \leq \rho(\tilde{\mathbf{x}}_n) \leq \left(1 + \frac{p}{n}\right)\rho(\lambda^*),$$

where the first inequality comes from the fact that $\mathbf{x}_n^*$ is the minimizer of ρ among allocations of length n. Then, from Caratheodory's theorem (see e.g., [Fedorov, 1972]) the number of support points used in λ^* is upper bounded by $p \leq d(d+1)/2+1$ (under the assumption that there are d linearly independent arms in $\mathcal{X}$). The final result follows by a rough maximization of $d(d+1)/2n + 1/n \leq d(d+1)/n$.

□

Remark (number of support points). Note that the same upper-bound for the number of support points holds for any design, due to the properties of the design matrices. In fact, any design matrix is symmetric by construction, which implies that it is completely described by $D = d(d+1)/2$ elements and can thus be seen as a point in $\mathbb{R}^D$. Moreover, a design matrix is a convex combination of a subset of points in $\mathbb{R}^D$ and thus it belongs to the convex hull of that subset of points. Caratheodory's theorem states that each point in the convex hull of any subset of points in $\mathbb{R}^D$ can be defined as a convex combination of at most $D+1$ points. It directly follows that any design matrix can be expressed using $(d(d+1)/2)+1$ points.

8.2 Proof of Theorem 3.4

Before proceeding to the proof, we introduce an additional helping lemma on the length of the phases.

Lemma 3.10. *For any phase j, the length is such that $n_j \leq \max\{M^*, \frac{16}{\alpha}N^*\}$ with probability $1-\delta$.*

Proof. We first summarize the different quantities measuring the performance of an allocation strategy in different settings. For any design $\lambda \in \mathcal{D}^K$, we define

$$\rho^*(\lambda) = \max_{y \in \mathcal{Y}^*} \frac{||y||^2_{\Lambda_\lambda^{-1}}}{\Delta^2(y)}; \quad \rho^{\mathcal{XY}}(\lambda) = \max_{y \in \mathcal{Y}} ||y||^2_{\Lambda_\lambda^{-1}}; \quad \rho^j(\lambda) = \max_{y \in \hat{\mathcal{Y}}_j} ||y||^2_{\Lambda_\lambda^{-1}}. \tag{3.46}$$

For any n, we also introduce the value of each of the previous quantities when the corresponding optimal (discrete) allocation is used

$$\rho_n^* = \rho^*(\lambda_{\mathbf{x}_n^*}); \quad \rho_n^{\mathcal{XY}} = \rho^{\mathcal{XY}}(\lambda_{\mathbf{x}_n^{\mathcal{XY}}}); \quad \rho_n^j = \rho^j(\lambda_{\mathbf{x}_n^j}). \tag{3.47}$$

Finally, we introduce the optimal designs

$$\lambda^* = \arg\min_{\lambda \in \mathcal{D}^K} \rho^*(\lambda); \quad \lambda^{\mathcal{XY}} = \arg\min_{\lambda \in \mathcal{D}^K} \rho^{\mathcal{XY}}(\lambda); \quad \lambda^j = \arg\min_{\lambda \in \mathcal{D}^K} \rho^j(\lambda). \tag{3.48}$$

Let ϵ^* be the smallest ϵ such that there exists a pair (x, x'), with $x \neq x^*$ and $x' \neq x^*$, such that the confidence set $\mathcal{S} = \{\theta : \forall y \in \mathcal{Y}, |y^\top(\theta - \theta^*)| \leq \epsilon\}$ overlaps with the hyperplane $\mathcal{C}(x) \cap \mathcal{C}(x')$. Since M^* is defined as the smallest number of steps needed by the $\mathcal{XY}$ strategy to avoid any overlap between $\mathcal{S}^*$ and the hyperplanes $\mathcal{C}(x) \cap \mathcal{C}(x')$, then we have that after M^* steps

$$c\sqrt{\frac{\rho^{\mathcal{XY}}_{M^*} \log_{c'}(K^2/\delta)}{M^*}} < \epsilon^*. \tag{3.49}$$

We consider two cases to study the length of a phase j.

Case 1: $\sqrt{\frac{\rho^j_{n_j}}{n_j}} \geq \frac{\epsilon^*}{c\sqrt{\log_{c'}(K^2/\delta)}}$. From Eq. 3.49 it immediately follows that

$$\frac{\rho^j_{n_j}}{n_j} \geq \frac{\rho^{\mathcal{XY}}_{M^*}}{M^*}. \tag{3.50}$$

From definitions in Eqs. 3.46 and 3.47, since $\hat{\mathcal{Y}}_j \subseteq \mathcal{Y}$ we have for any n, $\rho^j_n \leq \rho^{\mathcal{XY}}_n$. As a result, if $n_j \geq M^*$, since ρ^j_n/n is a non-increasing function, then we would have the sequence of inequalities

$$\frac{\rho^j_{n_j}}{n_j} \leq \frac{\rho^j_{M^*}}{M^*} \leq \frac{\rho^{\mathcal{XY}}_{M^*}}{M^*},$$

which contradicts Eq. 3.50. Thus $n_j \leq M^*$.

Case 2: $\sqrt{\frac{\rho^j_{n_j}}{n_j}} \leq \frac{\epsilon^*}{c\sqrt{\log_{c'}(K^2/\delta)}}$. We first relate the performance at phase j with the performance of the oracle. For any n

$$\rho^j_n = \rho^j(\lambda_{\mathbf{x}^j_n}) \leq \rho^j(\lambda_{\mathbf{x}^*_n}) = \max_{y \in \hat{\mathcal{Y}}_j} y^\top \Lambda^{-1}_{\lambda_{\mathbf{x}^*_n}} y = \max_{y \in \hat{\mathcal{Y}}_j} \frac{y^\top \Lambda^{-1}_{\lambda_{\mathbf{x}^*_n}} y}{\Delta^2(y)} \Delta(y) \leq \max_{y \in \hat{\mathcal{Y}}_j} \frac{y^\top \Lambda^{-1}_{\lambda_{\mathbf{x}^*_n}} y}{\Delta^2(y)} \max_{y \in \hat{\mathcal{Y}}_j} \Delta^2(y).$$

If now we consider $n = n_j$, then the definition case 2 implies that the estimation error $\sqrt{\rho^j_{n_j}/n_j}$ is small enough so that all the directions in $\mathcal{Y} - \mathcal{Y}^*$ have already been discarded from $\hat{\mathcal{Y}}_j$ and $\hat{\mathcal{Y}}_j \subseteq \mathcal{Y}^*$. Thus

$$\rho^j_{n_j} \leq \max_{y \in \mathcal{Y}^*} \frac{y^\top \Lambda^{-1}_{\lambda_{\mathbf{x}^*_{n_j}}} y}{\Delta^2(y)} \max_{y \in \hat{\mathcal{Y}}_j} \Delta^2(y) = \rho^*_{n_j} \max_{y \in \hat{\mathcal{Y}}_j} \Delta^2(y). \tag{3.51}$$

This relationship does not provide a bound on n_j yet. We first need to recall from Prop. 2.2 that for any $y \in \mathcal{Y}$ (and notably for the directions in $\hat{\mathcal{Y}}_j$) we have

$$|y^\top(\hat{\theta}_{j-1} - \theta^*)| \leq c\sqrt{y^\top A^{-1}_{j-1} y \log_{c'}(K^2/\delta)},$$

where $A_{j-1} = A_{\mathbf{x}^{j-1}_{n_{j-1}}}$ is the matrix constructed from the pulls within phase $j-1$. Since $\mathbf{x}^{j-1}_n$ is obtained from a $\mathcal{XY}$-allocation applied on directions in $\hat{\mathcal{Y}}_{j-1}$, we obtain that for any $y \in \hat{\mathcal{Y}}_j$

$$|y^\top(\hat{\theta}_{j-1} - \theta^*)| \leq c\sqrt{\log_{c'}(K^2/\delta)} \max_{y \in \hat{\mathcal{Y}}_{j-1}} \sqrt{y^\top A^{-1}_{j-1} y} = c\sqrt{\frac{\log_{c'}(K^2/\delta)\rho^{j-1}_{n_{j-1}}}{n_{j-1}}},$$

Reordering the terms in the previous expression we have that for any $y \in \hat{\mathcal{Y}}_j$

$$\Delta(y) \leq \widehat{\Delta}_{j-1}(y) + c\sqrt{\frac{\log_{c'}(K^2/\delta)\rho^{j-1}_{n_{j-1}}}{n_{j-1}}}.$$

Since the direction y is included in $\hat{\mathcal{Y}}_j$ then the discard condition in Eq. 3.39 failed for y, implying that $\widehat{\Delta}_{j-1}(y) \leq c\sqrt{\frac{\log_{c'}(K^2/\delta)\rho^{j-1}_{n_{j-1}}}{n_{j-1}}}$. Thus we finally obtain

$$\max_{y\in\hat{\mathcal{Y}}_j} \Delta(y) \leq 2c\sqrt{\frac{\log_{c'}(K^2/\delta)\rho^{j-1}_{n_{j-1}}}{n_{j-1}}}.$$

Combining this with Eq. 3.51 we have

$$\rho^j_{n_j} \leq \rho^*_{n_j} 4c^2 \frac{\log_{c'}(K^2/\delta)\rho^{j-1}_{n_{j-1}}}{n_{j-1}}.$$

Using the stopping condition of phase j and the relationship between the performance ρ^j, we obtain that at time $\bar{n} = n_j - 1$

$$\frac{\rho^j_{\bar{n}}}{\bar{n}} \geq \alpha\frac{\rho^{j-1}_{n_{j-1}}}{n_{j-1}} \geq \frac{\alpha}{4c^2\log_{c'}(K^2/\delta)}\frac{\rho^j_{n_j}}{\rho^*_{n_j}}$$

We can further refine the previous inequality as

$$\frac{\rho^j_{\bar{n}}}{\bar{n}} \geq \frac{\alpha\rho^*_{N^*}}{4N^*}\frac{N^*}{c^2\log_{c'}(K^2/\delta)\rho^*_{N^*}}\frac{\rho^j_{n_j}}{\rho^*_{n_j}} \geq \frac{\alpha\rho^*_{N^*}}{4N^*}\frac{\rho^j_{n_j}}{\rho^*_{n_j}},$$

where we use the definition of N^* in Eq. 3.25, which implies $c\sqrt{\log_{c'}(K^2/\delta)\rho^*_{N^*}/N^*} \leq 1$. Reordering the terms and using $\bar{n} = n_j - 1$, we obtain

$$n_j \leq 1 + \frac{4N^*}{\alpha}\frac{\rho^j_{n_j-1}}{\rho^j_{n_j}}\frac{\rho^*_{n_j}}{\rho^*_{N^*}}.$$

From Lemma 3.9 and the optimal designs defined in Eq. 3.48 we have

$$n_j \leq 1 + \frac{4N^*}{\alpha}\frac{(1+d(d+1)/(n_j-1))\rho^j(\lambda^j)}{\rho^j(\lambda^j)}\frac{(1+d(d+1)/(n_j-1))\rho^*(\lambda^*)}{\rho^*(\lambda^*)}.$$

Using the fact that the algorithm forces $n_j \geq d(d+1)+1$, the statement follows. □

Proof of Theorem 3.4.

Proof. Let J be the index of any phase for which $|\hat{\mathcal{X}}_J| > 1$. Then there exist at least one arm $x \in \mathcal{X}$ (beside x^*) for which the discarding condition in Lemma 3.8 is not triggered, which corresponds to the fact that for all arms $x' \in \mathcal{X}$

$$c||x - x'||_{A^{-1}_{\mathbf{x}^J_{n_J}}}\sqrt{\frac{\log(K^2/\delta)}{c'}} \geq \widehat{\Delta}_J(x, x').$$

By developing the right hand side, we have

$$\widehat{\Delta}_J(x,x') \geq \Delta(x,x') - c||x-x'||_{A^{-1}_{\mathbf{x}^J_{n_J}}} \sqrt{\frac{\log(K^2/\delta)}{c'}} \geq \Delta_{\min} - c\sqrt{\frac{\rho^J_{n_J}\log_{c'}(K^2/\delta)}{n_J}}$$

which leads to the condition

$$2c\sqrt{\frac{\rho^J_{n_J}\log_{c'}(K^2/\delta)}{n_J}} \geq \Delta_{\min}. \tag{3.52}$$

Using the phase stopping condition and the initial value of ρ^0 we have

$$\frac{\rho^J_{n_J}}{n_J} \leq \alpha\frac{\rho^{J-1}_{n_{J-1}}}{n_{J-1}} \leq \alpha^J\frac{\rho^0}{n_0} = \alpha^J.$$

By joining this inequality with Eq. 3.52 we obtain

$$\alpha^J \geq \frac{\Delta^2_{\min}}{4c^2\log_{c'}(K^2/\delta)},$$

and it follows that $J \leq \log(4c^2\log_{c'}(K^2/\delta)/\Delta^2_{\min})/\log(1/\alpha)$, which together with Lemma 3.10 leads to the final statement. □

8.3 Discussion on M*

Let us introduce the simplified notation $\mathcal{S}^*_n$ for the set $\mathcal{S}^*(\mathbf{x}^{\mathcal{XY}}_n) = \mathcal{S}^*_n$. Then, as introduced in Eq. 3.42, the definition of M^* is

$$M^* = \inf\{n \in \mathbb{N} : \forall x, x' \neq x^*, \mathcal{S}^*_n \cap (\mathcal{C}(x) \cap \mathcal{C}(x')) = \emptyset\}.$$

For any pair of suboptimal arms (x, x'), we introduce the following event

$$\mathcal{E}_n(x,x') = \{\mathcal{S}^*_n \cap (\mathcal{C}(x) \cap \mathcal{C}(x')) = \emptyset\},$$

where we have that whenever $\mathcal{E}_n(x,x')$ is true, there is no longer need to sample in the direction $x - x'$. Then, it follows that

$$\begin{aligned}
\mathcal{E}_n(x,x') &= \Big\{\forall\theta\in\mathcal{S}^*_n, \big(\Delta_\theta(x,x')\neq 0\big) \vee \big(\exists x^+\neq(x,x') : \Delta_\theta(x^+,x)\geq 0 \wedge \Delta_\theta(x^+,x')\geq 0\big)\Big\} \\
&\overset{(1)}{\Leftarrow} \Big\{\big(\forall\theta\in\mathcal{S}^*_n, \Delta_\theta(x,x')\neq 0\big) \vee \big(\forall\theta\in\mathcal{S}^*_n, (\exists x^+\neq x : \Delta_\theta(x^+,x)\geq 0) \\
&\qquad \vee (\exists x^+\neq x' : \Delta_\theta(x^+,x')\geq 0)\big)\Big\} \\
&\overset{(2)}{\Leftarrow} \Big\{\underbrace{\big(\forall\theta\in\mathcal{S}^*_n, \Delta_\theta(x,x')\neq 0\big)}_{\textbf{(a)}} \vee \underbrace{\big(\forall\theta\in\mathcal{S}^*_n, \exists x^+\neq x : \Delta_\theta(x^+,x)\geq 0\big)}_{\textbf{(b)}} \\
&\qquad \vee \underbrace{(\forall\theta\in\mathcal{S}^*_n, \exists x^+\neq x' : \Delta_\theta(x^+,x')\geq 0\big)}_{\textbf{(c)}}\Big\}.
\end{aligned}$$

These three possible cases[6] are as follows:

(a) is the case where the confidence set does not intersect the hyperplane of equivalence between x and x' (that is, the "common border" of the two cones)

(b) is the case where for any θ there is always a better arm than x

(c) is the case where for any θ there is always a better arm than x'.

While step (1) should not lead to a much larger event, step (2) rules out conditions which potentially make M^* smaller. In fact, in the case of three arms, while $\mathcal{E}$ still allows the three arms to be optimal, after step (2) this condition is no longer possible. In the following we re-write the three cases, noting that mostly because of step (2), all the following conditions may be loose. These conditions can be written as

$$
\begin{aligned}
(a) &\Leftarrow \frac{1}{\sqrt{n}}||x-x'||_{\lambda_{\mathcal{XY}}}\sqrt{\frac{\log(K^2/\delta)}{c'}} < |\Delta_{\theta^*}(x,x')|,\\
(b) &\Leftarrow \exists x^+ \neq x : \frac{1}{\sqrt{n}}||x^+-x||_{\lambda_{\mathcal{XY}}}\sqrt{\frac{\log(K^2/\delta)}{c'}} \leq \Delta_{\theta^*}(x^+,x),\\
(c) &\Leftarrow \exists x^+ \neq x' : \frac{1}{\sqrt{n}}||x^+-x'||_{\lambda_{\mathcal{XY}}}\sqrt{\frac{\log(K^2/\delta)}{c'}} \leq \Delta_{\theta^*}(x^+,x'),
\end{aligned}
$$

where we use the notation $||x^+-x||_{\lambda_{\mathcal{XY}}} = ||x^+-x||_{\Lambda^{-1}_{\lambda_{\mathcal{XY}}}}$. Reordering the terms we obtain:

$$
\frac{M^*}{\log_{c'}(K^2/\delta)} \leq \max_{x,x'\neq x^*}\left\{\min\left[\frac{||x-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x,x')}; \min_{x^+\neq(x,x')}\min\left(\frac{||x^+-x||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x)}; \frac{||x^+-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x')}\right)\right]\right\}.
$$

Upper-bound. Let us now consider an upper-bound on M^*. We can derive the following sequence of inequalities:

$$
\begin{aligned}
\frac{M^*}{\log_{c'}(K^2/\delta)} &\leq \max_{x,x'\neq x^*}\left\{\min\left[\frac{||x-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x,x')}; \min_{x^+\neq(x,x')}\max\left(\frac{||x^+-x||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x)}; \frac{||x^+-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x')}\right)\right]\right\}\\
&\leq \max_{x,x'\neq x^*}\left\{\min_{x^+\neq(x,x')}\max\left(\frac{||x^+-x||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x)}; \frac{||x^+-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^+,x')}\right)\right]\right\}\\
&\leq \max_{x,x'\neq x^*}\left\{\max\left(\frac{||x^*-x||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^*,x)}; \frac{||x^*-x'||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(x^*,x')}\right)\right]\right\}
\end{aligned}
$$

[6] There might also be a case **(d)** where for two suboptimal arms $\mathcal{C}(x)\cap\mathcal{C}(x') = 0^d$. This happens when the cones of the two suboptimal arms x, x' are not "neighbors" (while there is a separation hyperplane between any two arms, this hyperplane does not necessarily define a boarder for $\mathcal{C}(x)$ or $\mathcal{C}(x')$). In this cases, the event $\mathcal{E}_n(x,x') = \{\mathcal{S}^*_n \cap (\mathcal{C}(x)\cap\mathcal{C}(x')) = \emptyset\}$ is true as soon as we are sure that $0^d \notin \mathcal{S}^*_n$. But since $\theta = 0^d$ would imply that all arms are equivalent, we exclude this case and thus consider $n = 0$ the sample complexity needed such that event $\mathcal{E}_n(x,x') = \{\mathcal{S}^*_n \cap (\mathcal{C}(x)\cap\mathcal{C}(x')) = \emptyset\}$ is true whenever $\mathcal{C}(x)\cap\mathcal{C}(x') = 0^d$.

$$\leq \max_{y \in \mathcal{Y}^*} \frac{||y||^2_{\lambda_{\mathcal{XY}}}}{\Delta^2(y)} \leq \frac{d}{\Delta^2_{\min}},$$

where in the last step we used the equivalence theorem for the $\mathcal{XY}$-design and the maximization over the gaps. This means that M^* matches the sample complexity of the static allocations, which is equal to the worse case value[7] of N^*.

MAB setting. It is also interesting to look more carefully at the value that M^* would have in a simple multi-armed bandit problem. In fact, in the MAB case, $\mathcal{XY}$ reduces to a uniform allocation over arms and $||x^* - x'||^2_{\lambda_{\mathcal{XY}}} = 2K$. Then we obtain

$$\frac{M^*_{\text{MAB}}}{\log_{c'}(K^2/\delta)} \leq 2K \max_{x, x' \neq x^*} \left\{ \min \left[\frac{1}{\Delta^2(x, x')}; \min_{x^+ \neq (x, x')} \min \left(\frac{1}{\Delta^2(x^+, x)}; \frac{1}{\Delta^2(x^+, x')} \right) \right] \right\}. \tag{3.53}$$

In this case it is easy to construct a problem where this value is larger than the sample complexity of the MAB problem. Consider for instance the setting where $K = 5$ and $\mu_1 = 1, \mu_2 = \mu_3 = 0.95, \mu_4 = \mu_5 = 0.9$. Then we have that

$$\begin{aligned} \frac{M^*_{\text{MAB}}}{\log_{c'}(K^2/\delta)} &\leq 2K \max_{x, x' \neq x^*} \left\{ \min_{x^+ \neq (x, x')} \min \left(\frac{1}{\Delta^2(x^+, x)}; \frac{1}{\Delta^2(x^+, x')} \right) \right\} \\ &= 2K \max \left(\frac{1}{\Delta^2(x_1, x_2)}; \frac{1}{\Delta^2(x_1, x_3)} \right) = \frac{10}{0.05^2} = 4000, \end{aligned}$$

which is larger than $H_{\text{MAB}} = \sum_{i=2}^5 \frac{1}{\Delta_i^2} = 2/0.05^2 + 2/0.1^2 = 800 + 200 = 1000$.

In this example M^* seems to have an important value compared to N^* and it seems important to verify whether the sample $N = \max(M^*, \zeta N^*)$ actually scales with M^* in a MAB setting, as the dimensionality of the of the problem grows. Fig. 3.5 gives an illustration of this, in a MAB setting similar to the one presented above.

Simulations. In this simulations we consider a MAB setting, where we assume the value of $\theta^* \in \mathbb{R}^d$ is $\theta^* = [1 \quad 0.95 \quad 0.2 \quad 0.2 \quad \dots \quad 0.2]^\top$. We remind that here d gives the dimensionality of the problem, as well as the number of arms. For this value of θ^*, it follows that in the given setting there is a unique best arm x_i, a unique $\Delta_{\min} = 0.05$, while all other gaps are $\Delta_i = 0.8$, for $i = 3, \dots, d$. We compute the value of the oracle sample complexity as $N^* = H_{\text{MAB}} \cdot \log_{c'}(K^2/\delta)$ and similarly, we compute M^*, as defined before in Eq. 3.53, multiplied by $\log_{c'}(K^2/\delta)$. We remind that Th. 3.4 suggests that N should scale as the maximum between M^* and ζN^*, where ζ is a factor depending on the length of the phases in $\mathcal{XY}$-Adaptive. Thus, since ζ affects which of the two terms determines the maximum, we plot in Fig. 3.5 the values of M^* *(black line)* and the value of ζN^* *(green line)*, where the parameter for the change of phase is $\alpha = 0.8$ and $\zeta = 10/\alpha = 12.5$. As we can also notice from the definitions of the sample

[7] This is not surprising since we basically reduced the event $\mathcal{E}$ to the case where all the suboptimal arms are removed by using the $\mathcal{XY}$-design.

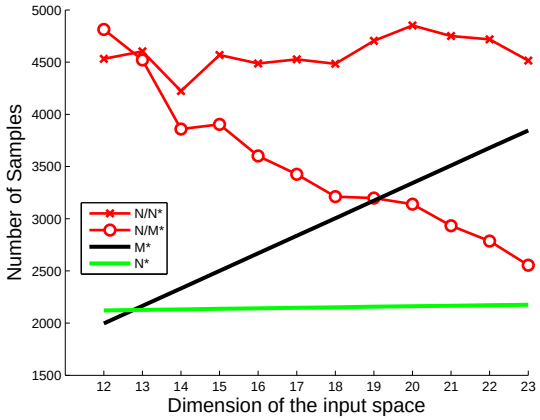


Figure 3.5: How N scales with N^* and M^* in a MAB with 12 to 23 arms.

complexities, while the change of N^* from dimension d to dimension $d+1$ is given by an additive factor equal to $1/\Delta_i^2 = 1.5625$ (imperceivable at the plot scale), M^* is characterized by a much more important growth, from $2d/\Delta^2_{\min}$ in dimension d (which also gives the number of arms since in the MAB setting $d = K$), to $2(d+1)/\Delta^2_{\min}$ for dimension $d+1$.

The two red curves are obtained by dividing the number of samples N needed by the $\mathcal{XY}$-Adaptive algorithm to identify the best arm. This sample complexity was averaged over 10 runs of the adaptive algorithm with phase length given by $\alpha = 0.8$. We plot in Fig. 3.5 the rescaled value of N divided by M^* *(red line with white circles)* and respectively, N divided by ζN^* *(red line with red crosses)*.

The goal of the simulations was to check on a simple example how the sample complexity N scales with M^* and N^*, and in particular, to verify how important is the role played by M^* in determining the sample complexity of the $\mathcal{XY}$-Adaptive algorithm. This is why we chose to plot the results for dimensions from 12 to 23, since for this setting M^* becomes increasingly bigger than ζN^* as the dimensionality grows bigger than 12. The results show that although M^* has values bigger than $\zeta N*$ for these dimensions, N seems to rather scale with ζN^*, as shown by the values of $N/\zeta N^*$, whereas N/M^* rapidly decreases as the dimensionality grows and as M^* becomes bigger than ζN^*. This simple example suggests that the presence of the term M^* in the sample complexity bound of N might be an artifact of the proof, and the performance of the $\mathcal{XY}$-Adaptive might in fact depend only on the sample complexity of $\mathcal{XY}$-Oracle.

Chapter 4

Optimal Experimental Design in Linear Bandits

In Chapter 3, where we studied the best arm identification (BAI) in linear bandits, we identified an interesting connection to the G-optimality criterion defined in the Optimal Experimental Design (OED) literature. More precisely, the G-design is a global optimization criterion, according to which the sample allocation strategy is optimal if it minimizes the estimation error of the reward function uniformly over the entire domain. From the BAI perspective, the objective of G-design is thus to estimate uniformly well the value of each arm.

The connections between the multi-armed bandit setting and OED criteria have been considered (although mostly indirectly) in several recent works, from the active learning strategies proposed in [Antos et al., 2009, Carpentier et al., 2011] with the goal of learning equally well the mean values in a K-armed bandit, to Gaussian process optimization [Srinivas et al., 2010], and dynamical system identification [Llamosi et al., 2014]. On the other hand, in the linear stochastic bandit, the design and analysis of effective and adaptive allocation strategies for OED criteria is largely unexplored, in spite of the similarity with the most used OED criteria, as we will see later on in the chapter.

In this chapter[1], we take a closer look at the properties required for an adaptive allocation strategy to reach OED criteria relevant in a linear bandit setting. For the G-optimal design (which minimizes the estimation error of each arm) and then extending to the closely related V-optimal design (which minimizes the average estimation error over all the arms), we propose a preliminary study of the properties required by online learning algorithms to obtain good performances under these criteria and using two heteroscedastic noise models.

Contents

[1] This chapter is a joint work with Alessandro Lazaric and Rémi Munos. A part of Sect. 2 was presented in [Soare et al., 2013].

1 Optimal Experimental Design

The core goal of stochastic multi-armed bandit strategies –finding optimal sample allocation strategies– has a clear, direct connection to the OED literature, where a learning agent conducts experiments and analyses the noisy outcomes with the goal of maximizing the obtained information about the environment. In other words, in OED the learner aims at making the most efficient use of the available experiments, where the measure of the efficiency (or the *information* that the agent is looking to obtain) might be in some cases similar to the information needed to minimize the regret (like in the MAB formulation), but can also be extended to more general goals and performance measures, as we will see in the following. Indeed, we present in the following section the sequential allocation problem from an OED perspective, briefly introducing the most typical optimality criteria, with focus on the practical motivation of the two criteria that we study later on in the chapter.

1.1 Assumptions and definitions

Starting from a statistical motivation, but having wide applications in several other fields (i.e. machine learning, optimization), the OED problem can be summarized as follows: Assume a function $f(x, \theta^*)$ and a limited number n of outcomes of experiments that can be observed, denoted $r_1, \dots, r_n$. The function f is a real-valued function of two arguments. The first argument, x, is a vector and represents the controllable part of the experiment (typically the learner gets to choose from a set $\mathcal{X}$ of given experiments). The second argument, θ^*, is a parameter capturing the unknown conditions of the environment. The goal of the learner is to learn as much as possible about the function f and the parameter θ^*.

One of the most simple and extensively used model functions in the OED literature is assuming that the model is linear with respect to the (unknown) system parameters. In addition, to make the problem more realistic, it is commonly assumed that the observations are noisy, which bring us back to the reward formulation of stochastic linear bandits:

$$r(x) = x^\top \theta^* + \eta,$$

where the noise in the observations is captured in $\eta_1, \dots, \eta_n$ (where η_i is the noise in experiment i), random variables with an unknown distribution function and typically independent of each other.

It is also usual to assume that the learner can choose the experiment conditions x out of a fixed, finite set of options, all of them revealed before the beginning of the experiment. Denoting by $\mathcal{X} = \{x_1, \dots, x_K\}$ the set of K options regarding the

controllable part of the experiment, we say that the **experimental design of size n** is given by a sequence of experiments $\mathbf{x}_n = (x_1, \ldots, x_n)$, where x_j is the experiment condition tested at time step j. Also, we denote by $\{N_1, \ldots, N_K\}$ the (integer) number of times each of the K experiments has been tested, such that $\sum_{i=1}^{K} N_i = n$. We say that all arms x_i that have corresponding $N_i > 0$ are **support points** of the experimental design given by the sequence $\mathbf{x}_n$.

Moreover, given an experimental design $\mathbf{x}_n$, we introduce the *design matrix* $A_{\mathbf{x}_n}$, that we have already used in the previous chapter, defined as

$$A_{\mathbf{x}_n} = \sum_{i=1}^{K} N_i x_i x_i^\top$$

where x_i is the i-th vector in $\mathcal{X}$. The importance of the design matrix comes from the crucial role that it plays in the variance/error of the estimator of θ^*, computed, as previously indicated in Eq. 2.8, since the *optimal* experimental design is given by the sequence $\mathbf{x}_n$ which minimizes (in some sense) the **error covariance matrix** $A_{\mathbf{x}_n}^{-1}$.

Lastly, as we have seen in Chap. 3, when the condition that the n_i are integers is dropped, we can then obtain **relaxed** optimal experimental designs. In this case, we say that the design is characterized by the vector $\lambda \in \mathcal{D}^K$, where the i-th feature of λ, denoted λ_i, indicates the proportion of the n experiments used to observe outcomes in experimental conditions given by $x_i \in \mathcal{X}$. Therefore, we have that $\sum_{i=1}^{K} \lambda_i = 1$, and $N_i = \lambda_i \cdot n$. Similarly, we introduce the notation A_λ for the design matrix corresponding to the design characterized by λ, and A_λ^{-1} for its error covariance matrix.

We now briefly introduce some of the most common optimality criteria, with focus on the ones that we consider in the following sections as performance measures for the adaptive sampling algorithms.

1.2 Optimality criteria

Perhaps the most popular OED performance measure is given by the *determinant criterion*, or the **D-Optimal Design**, where the goal is to find the optimal proportions of allocations $n_1, \ldots, n_K$ which maximize the determinant of $A_{\mathbf{x}_n}$, or equivalently, which minimize the determinant of the error covariance matrix, since $(\det A)^{-1} = \det(A^{-1})$. The formulation of the relaxed D-Optimal Design as an optimization problem is

$$\begin{aligned} &\text{minimize} \quad -\log\det\left(\sum_{i=1}^{K} \lambda_i x_i x_i^\top\right)^{-1} \\ &\text{subject to} \quad \sum_{i=1}^{K} \lambda_i = 1; \lambda_i \geq 0, \forall i. \end{aligned} \tag{4.1}$$

Basically, this corresponds to minimizing the confidence region for the value of the parameter θ^*. In particular, as pointed out from the earlier works [Silvey, 1972, Sibson, 1972] there is a duality relationship between the D-Optimal Design and the problem of finding minimal volume covering ellipsoids, the latter having applications in a wide

range of areas (e.g., optimization, data analysis, computational geometry). To see this connection, notice that for a centre $c \in \mathbb{R}^d$ and a shape given by some positive definite matrix H, the ellipsoid $E \in \mathbb{R}^d$, is defined as follows

$$E(c, H) = \{x \in \mathbb{R}^d : (x - c)^\top H(x - c) \leq d\}.$$

Now, we can rewrite $H^{-1} = LL^\top$ and introduce $z = L^{-1}(x - c)$. At this point, we can also redefine the ellipsoid as

$$E(c, H) = \{x = c + Lz : ||z|| \leq \sqrt{d}\}.$$

Finally, we remind that $\text{vol}(E(c, H)) = |\det\ L| \cdot (\sqrt{d})^d \cdot \text{vol}(\text{hypersphere}_d)$, thus $\text{vol}(E(c, H)) = \text{const}(d)/\sqrt{\det\ H}$, and minimizing the volume of ellipsoid E is equivalent to minimizing $-\det H$ (and respectively $-\log\det H$). More details on the convex duality can be found, for instance, in [Todd, 2016].

In addition, insights from geometry can be used to identify and characterize the support points. More precisely, it is clear that only points at the margin of the convex hull of $\mathcal{X}$ are good candidates for support points. Also, if the support points are known, then, as pointed out in [Titterington, 1975], their corresponding allocation proportions in λ can be determined by using the following property.

Property 1. For any D-optimal design and for any point $x \in \mathcal{X}$, we have that

$$x^\top \left(\sum_{i=1}^{K} \lambda_i x_i x_i^\top \right)^{-1} x = d \tag{4.2}$$

if x is a support point and

$$x^\top \left(\sum_{i=1}^{K} \lambda_i x_i x_i^\top \right)^{-1} x < d \tag{4.3}$$

otherwise.

This property follows directly from the equivalence theorem (Prop. 3.2) and the fact that the support points are situated on the margin of the enclosing ellipsoid.

Another important criterion with a practical relevance for prediction problems, is the *global* optimality criterion, **G-Optimal Design**, defined as

$$\begin{aligned} &\text{minimize } \max_{x \in \mathcal{X}} x^\top \left(\sum_{i=1}^{K} \lambda_i x_i x_i^\top \right)^{-1} x. \\ &\text{subject to } \sum_{i=1}^{K} \lambda_i = 1; \lambda_i \geq 0, \forall i \end{aligned} \tag{4.4}$$

As its formulation suggests, a sample allocation strategy reaches the G-optimality criterion if it minimizes the maximal prediction error over all points in the input set. As shown in [Kiefer and Wolfowitz, 1960], there is equivalence between the determinant

criterion (D-) and the global criterion when the variance in the noise of the observation is homoscedastic. Therefore, the property in Eq. 4.2 also holds for the G-optimal design, under the conditions of the equivalence theorem.

Lastly, directly connected to the G-criterion is the scalarization that rather considers minimizing the average prediction error of the point in the input set, to which we refer to as the **V-Optimal Design**. Its formulation is as follows:

$$\begin{aligned} &\text{minimize } \frac{1}{K}\sum_{x\in\mathcal{X}} x^\top \left(\sum_{i=1}^{K}\lambda_i x_i x_i^\top\right)^{-1} x. \\ &\text{subject to } \sum_{i=1}^{K}\lambda_i = 1; \lambda_i \geq 0, \forall i \end{aligned} \tag{4.5}$$

Other scalarizations lead to optimality criteria such as minimizing the **maximum eigenvalue** of the error covariance matrix (the E-criterion) or minimizing its **trace** (the T-criterion). A more detailed presentations of the existing scalarizations can be found, for instance, in [Pukelsheim, 2006, Chap. 6], [Ahipaşaoğlu, 2009], [Boyd and Vandenberghe, 2004, Chap. 7.5].

1.3 Applications

It is also important to have in mind the potential applications where these optimality criteria are of use. Let us now consider some examples where the G- and V- optimality criteria have a direct application.

Consider for instance a recommender system problem, where given a database of several thousand movies, the goal is to be able to estimate with good accuracy the ratings that a user would give to all the movies. We suppose here that the environment fulfils the assumption of the existence of a global linear structure relating the preferences of the user to the movie characteristics and that any choice of the sequence of movies for which to collect direct feedback from the user is allowed. Ideally, using limited feedback from the user concerning only a small part of the movies, the system is able to infer with good accuracy the ratings for the rest of the movies. The success of a sample allocation strategy in providing accurate estimated ratings, will depend upon the ability to use the direct, limited, user feedback on the most informative movies.

Similarly, we can also consider the situation where a company has to make an accurate prediction of the result of a poll (e.g., an election). The company has relevant information on all the participants (such as, age, profession), but can only question a small number of them on their voting preferences. The obtained responses can then be used to infer a preference model for all the individuals in the group. To obtain an accurate estimation of the preference model, the company can use a strategy that sequentially selects the member of the group to which to send the questionnaire. The goal is thus to use the available observations to choose the most informative members of the group, whose response about their voting intentions is relevant in also estimating the voting preferences of the other members of the group.

In this type of scenarios, the G-optimal design leads to an equally good estimation of the rating/vote for all the possible options, whereas the V-optimal design is meant to minimize the average prediction error over the entire set of options. Following the same reasoning, the applicability of G- and V-criteria can be easily extended to a wide range of applications domains (e.g, in finance one wishes to be able to estimate as well as possible the value of each asset in a portfolio, or the overall value/risk of the portfolio).

In the rest of the chapter, we give the formalization of the two optimal design criteria from the bandit perspective (or rather, making connections to the bandit literature vocabulary). We propose new samples allocation strategies adapted in particular to the different variance of the noise in the rewards. We then present the limitations and advantages of the proposed strategies and provide some remarks on open problems and future directions.

2 Adaptive OED with Parameterized Heteroscedastic Noise

For the study of optimal design strategies in the linear stochastic bandits, we start by focusing on the G-optimal design, which is a worst-case strategy for the BAI problem in linear bandits. Nonetheless, the G-optimal strategy considered previously in the BAI chapter was static, and thus the strategy acted *blindly*, improving the accuracy of the estimation uniformly over all dimensions. We will now focus on designing **adaptive** G-allocation strategies, where the global optimality criterion needs to be reached in an environment where the noise in each arm will have a different, unknown variance. The behavior of an efficient sample strategy becomes in this case harder to design, since the choice of the support points, as well as the proportion of pulls allocated to each of them, will no longer depend uniquely on the geometry of the space. In fact now we have to first estimate the noise specific to each arm, then recompute the G-optimal allocation based on the current estimates. The regret of the learner will then be defined as the difference between his obtained loss and that of an optimal (oracle) allocation, that knows the arm variances beforehand.

The uniform estimation with limited budget and heteroscedastic noise has been recently studied in the bandit literature, but only for the multi-armed bandit (MAB), where the input space is the set of the orthogonal arms of the standard stochastic bandit. With the objective of estimating uniformly well the mean values of several distributions, the authors in [Antos et al., 2010] and [Carpentier et al., 2011] estimate the variance per arm and exploit heteroscedasticity to allocate more samples to the parts of the input space where the variance is larger. Our new formulation generalizes the MAB model by allowing the arms to be correlated and by taking into account the complete dimensionality of the problem at once. We study this problem in a specific reward model with heteroscedastic noise, as detailed in the following. We then propose some intuitions on how to design adaptive strategies for obtaining the corresponding

G-optimal allocations.

2.1 The model

We assume a multivariate noise characterized by an unknown covariance matrix. The particularity here comes from the assumption that the "noisy" part of the reward will be given, for each time step t, by the inner product between the chosen arm x_t and the d- dimensional noise vector sampled according the unknown covariance matrix. Let us briefly restate here the considered linear bandit model, together with the additional precisions about the noise.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a finite set of bounded arms such that $|\mathcal{X}| = K$ and for any $x \in \mathcal{X}$, $||x|| \leq L$. When an arm x is chosen, a noisy realization of an unknown function f is observed. We are again in the linear bandit setting, therefore we define a random realization r from arm x as

$$r(x) = f(x) + \varepsilon(x) = x^\top \theta^* + x^\top \eta, \tag{4.6}$$

where η is a multivariate noise with zero mean and covariance matrix Σ, and θ^* is an unknown vector. Notice that unlike standard linear regression, in this case we consider a heteroscedastic noise $\varepsilon(x)$ which strictly depends on the arm x. Also, as already pointed out, in contrast with the standard linear stochastic bandit problem where the goal is to choose the arm in $\mathcal{X}$ that yields the maximal reward, here we focus on the problem of how to allocate a budget of n pulls on different arms in order to have an estimate of θ^* that allows to predict the value off all $x \in \mathcal{X}$ with equal (and maximized) accuracy.

Let $\mathcal{A}$ be an algorithm that at each round $t = 1, \ldots, n$ chooses an arm x_t and observes the corresponding realization $r_t = x_t^\top \theta^* + \varepsilon_t = x_t^\top \theta^* + x_t^\top \eta_t$. For every sequence $\{x_t, r_t\}_{t=1}^n$ of arm-observation pairs we denote by $X_n \in \mathbb{R}^{n \times d}$ with $[X_n]_{t,i} = x_{t,i}$ the matrix of chosen arms, by $R_n \in \mathbb{R}^n$ with $[R_n]_t = r_t$ the vector of observations, and by $E_n \in \mathbb{R}^n$ with $[E_n]_t = \varepsilon_t$ the vector of noise. Once $\mathcal{A}$ used all the budget of n, we compute the least-squares estimate of θ^*:

$$\hat{\theta}_n = A_n^{-1} b_n,$$

where $A_n = \sum_{t=1}^n x_t x_t^\top = X_n^\top X_n$ and $b_n = \sum_{t=1}^n r_t x_t = X_n^\top R_n$.

For each arm $x \in \mathcal{X}$ we define the prediction error of $\hat{\theta}_n$ as the expected quadratic loss $L_n(x) = \mathbb{E}[(x^\top \hat{\theta}_n - x^\top \theta)^2]$ where the expectation refers to all possible sources of randomization in the observations r_t and in the choice of the sequence of arms x_t. Overall, we define the performance of the algorithm $\mathcal{A}$ by the loss corresponding to the worst estimated arm, i.e.

$$L_n(\mathcal{A}) = \max_{x \in \mathcal{X}} L_n(x). \tag{4.7}$$

In the sequel, we refer to the loss defined in Eq. 4.7 as the G-opt loss of algorithm $\mathcal{A}$. The objective is that given a fixed budget n, we manage to design an algorithm $\mathcal{A}$ that minimizes the G-loss $L_n(\mathcal{A})$.

2.2 The Optimal Static Allocation Algorithm

Consider a static allocation strategy $\mathcal{A}$ which selects arms $\{x_t\}$ independently from the observations. The covariance of $\hat{\theta}_n$ can be computed as

$$\begin{aligned}\mathbb{V}[\hat{\theta}_n|X_n] &= \mathbb{E}\Big[(\hat{\theta}_n - \theta^*)(\hat{\theta}_n - \theta^*)^\top | X_n\Big] \\ &= (X_n^\top X_n)^{-1} X_n^\top \mathbb{E}[E_n E_n^\top] X_n (X_n^\top X_n)^{-1} \\ &= (X_n^\top X_n)^{-1} X_n^\top \Omega_n X_n (X_n^\top X_n)^{-1},\end{aligned} \tag{4.8}$$

where $\Omega_n = \mathrm{diag}(\sigma^2(x_1), \dots, \sigma^2(x_n))$ with $\sigma^2(x_t) = \mathbb{V}[\varepsilon_t] = x_t^\top \Sigma x_t$. Since $\mathcal{A}$ is static, then the previous expectations are conditioned on the fixed set of arms chosen over n rounds, which are summarized by the matrix X_n. Thus, the loss $L_n(x)$ can now be expressed directly as

$$L_n(x; \Sigma, X_n) = \mathbb{E}[(x^\top \hat{\theta}_n - x^\top \theta^*)^2 | X_n] = x^\top \mathbb{V}[\hat{\theta}_n | X_n] x,$$

where we make explicit the dependency of the loss on the sequence of arms X_n and the covariance matrix Σ in $L_n(x; \Sigma, X_n)$. As a result, an optimal static allocation should select the sequence of arms $\mathbf{x}_n \in \arg\min_{X_n} \max_{x \in \mathcal{X}} L_n(x; \Sigma, X_n)$. Although this allocation cannot be computed in closed form, an almost equivalent allocation[2] can be obtained by pulling at each time t the arm

$$x_t = \arg\max_{x \in \mathcal{X}} L_t(x; \Sigma, X_{t-1}), \tag{4.9}$$

which corresponds to pulling the arm with the largest loss at each time step. Notice that this allocation does not require any actual observation r_t since it only relies on the set of arms $\mathcal{X}$ and on the covariance matrix Σ.

It is interesting to analyze the behavior of the optimal allocation in simple cases:

- If the arms form an orthogonal basis in $\mathbb{R}^d$, then the arms are all independent and the problem reduces to the active learning in multi-armed bandit setting studied in [Antos et al., 2010, Carpentier et al., 2011]. As a result, the optimal strategy directly allocates the budget over arms proportionally to their variance (that is, the number of times an arm x is pulled is proportional to $\sigma^2(x) / \sum_{x'} \sigma^2(x')$).

- If the noise is homoscedastic (that is, $\Sigma = \sigma^2 I$), then the optimal allocation is no longer driven by the variance of the arms, but it still needs to compensate for a possibly uneven distribution of the arms in $\mathbb{R}^d$ by allocating less samples to the arms in regions of $\mathbb{R}^d$ which are dense of many arms.

- In the general case of heteroscedastic noise and an arbitrary set of arms $\mathcal{X}$, the optimal strategy implements an allocation that balances both the different variance of the arms and their uneven distribution in $\mathbb{R}^d$.

[2] For a discussion on the performance of the greedy incremental allocation, please see Sect 4.1.

This qualitative description of the behavior of the optimal static allocation can be also illustrated by inspecting the definition of the loss L. Let $s \in \mathbb{R}^n$ be $s = x^\top (X_n^\top X_n)^{-1} X^\top$ such that for any t, $s_t = (x^\top A_n^{-1}) x_t$. Then, the loss can be written as

$$L(x; \Sigma, X_n) = \sum_{t=1}^{n} s_t^2 \sigma^2(x_t) = \sum_{x_i \in \mathcal{X}} N_{i,n} \underbrace{(x^\top A_n^{-1} x_i)^2}_{(a)} \underbrace{((x_i)^\top \Sigma\, x - i)}_{(b)}, \tag{4.10}$$

where $N_{i,n}$ denotes the number of times that arm x_i $(i = 1, \dots, K)$ was pulled up to time n. This form of the loss emphasizes the two elements that should be taken into account in designing an allocation strategy: the shape of the input space (term a) and the noise covariance matrix (term b).

2.3 GOM Algorithm

In a more realistic setting, the noise covariance matrix is unknown in advance, thus we need a learning strategy able to estimate Σ and at the same time to implement the optimal allocation suggested by Eq. 4.9. This requires to find a suitable trade-off between the *exploration* of the entire input space, with the objective of learning a good estimate of Σ (denoted $\widehat{\Sigma}$), and the *exploitation* of the current estimate to select arms according to the (estimated) optimal allocation. In order to define such a trade-off we rely on the construction of confidence bounds on the loss of each arm. The resulting algorithm for adaptive G-Optimal design with heteroscedastic Multivariate noise (GOM) is sketched in Alg. 5.

Algorithm 5 ADAPTIVE G-OPTIMAL ALLOCATION WITH MULTIVARIATE NOISE

Input: input space $\mathcal{X}$, budget n
while $t \leq n$ **do**
  Compute $B_t(x) = \max_{\tilde{\Sigma} \in \Gamma_t} L(x; \tilde{\Sigma}, X_{t-1})$
  Select $x_t = \arg\max_{x \in \mathcal{X}} B_t(x)$
  Pull x_t twice and observe r_t, r'_t
  Compute $z_t = \frac{1}{2}(r_t - r'_t)^2$
  Update the estimated variance vector $\hat{v}_t$ (Eq. 4.12)
  Update the confidence set Γ_t (Eq. 4.13)
  t = t + 2
end while
Return $\hat{\theta}_n = A_n^{-1} b_n$ (and its corresponding G-opt loss)

The most critical aspect of the algorithm is how to actually compute an estimate $\widehat{\Sigma}$ and how to build a confidence bound on it. In fact, although the idea of using upper confidence bounds has already been used in [Carpentier et al., 2011], unlike in the multi-arm bandit setting, the estimation of the variance of the noise is not trivial. In fact, we can only rely on noisy observations perturbed by a multivariate heteroscedastic noise which cannot be observed directly and which depend on the choice of the arm itself.

In order to simplify the derivation of an estimate of the covariance matrix and the construction of a confidence bound, we first introduce an assumption on the noise η.

Assumption 1. *Let the noise η be bounded in $[-1,1]^d$. Furthermore, let $v \in \mathbb{R}^d$ be the vector $v = [\sigma_1^2, \dots, \sigma_d^2]$ such that the covariance matrix is the diagonal matrix $\Sigma = diag(\sigma_1^2, \dots, \sigma_d^2)$.*

This is a common simplifying assumption in regression model depending on the estimation of a noise covariance matrix. For instance, the same assumption of an unknown diagonal covariance was also considered in [Fuller and Rao, 1978], where the authors consider a similar linear regression model, for static strategies, with a fixed sequence of noise in each arm. While the boundedness of the noise allows us to use standard concentration inequalities (which allow a possible extension to sub-Gaussian noise), a diagonal covariance matrix makes it possible to reduce the covariance estimation to a regularized regression problem. In fact, we notice that for any arm $x \in \mathcal{X}$, if we denote $y = x^2$ componentwise, then the variance of the corresponding observations can be written as:

$$\sigma^2(x) = x^\top \Sigma x = \sum_{i=1}^{d} x_i^2 \sigma_i^2 = y^\top v. \tag{4.11}$$

Equation (4.11) shows that the variance of the observations is a linear function with respect to the inputs y and the unknown variance vector v.

Although this simplifies the estimation of Σ, it is still required that at each time step t, when arm x_t is selected, two independent samples r_t and $r_{t'}$ need to be generated. Hence we can construct the sample $z_t = \frac{1}{2}(r_t - r'_t)^2$, which is an unbiased sample of the variance of the observations corresponding to x_t, since $\mathbb{E}[z_t] = y_t^\top v = \sigma^2(x_t)$. Thus, we can set up the following regularized least squares problem[3]

$$\hat{v}_t = \arg\min_{v \in \mathbb{R}^d} \left[\frac{2}{t} \sum_{t'=1}^{t/2} \left(y_{t'}^\top v - z_{t'} \right)^2 + \lambda ||v||_2 \right] = C_t^{-1} d_t, \tag{4.12}$$

where $C_t = \sum_{1=1}^{t} y_i y_i^\top + \lambda I = Y_t^\top Y_t + \lambda I$, $d_t = \sum_{i=1}^{t} z_i y_i = Y_t^\top Z_t$.

Note that the requirement to sample each arm twice does not necessarily correspond to a worsening of the performance. In fact, the same arm can be the one maximizing the loss several times before consuming the budget. Also, according to the performance measure of the algorithm, its efficiency can only be measured after the final sampling round, therefore the order in which the arms are selected is not important.

For the estimated variance vector $\hat{v}$ we can now rely on the self-normalized martingale techniques previously developed in the linear bandit setting [Abbasi-Yadkori et al., 2011]. From it, we derive the following lemma.

[3]Notice that because of the double sampling, the total amount of samples available after t steps is only $t/2$. This is why in Eq. 4.12 we use the index $t' = \{1, \dots, t/2\}$.

Lemma 4.1. *Let $\hat{v}_n$ be the regularized least-squares estimate of the variance vector v. Let $||v||_2^2 = \sum_{i=1}^d \sigma_i^4 \leq V^2$ and η satisfy Assumption 1. Then the confidence set (recall that $||y_t||_2 \leq L^2$)*

$$\Gamma_t = \left\{ s \in \mathbb{R}^d, \|\hat{v}_t - s\|_{C_t} \leq \sqrt{d \ \log\left(\frac{1 + tL^4/\lambda}{\delta}\right)} + \lambda^{1/2} V \right\}, \tag{4.13}$$

is such that $v \in \Gamma_t$ with probability at least $1 - \delta$, for all $\delta > 0$, and $t \geq 0$.

The construction of the confidence set allows to choose at each time step the arm which maximizes the loss, for all possible values of the estimate of Σ. Now, it is crucial to be able to progressively tighten the confidence sets and to select at each step the worst-case arm, that is, x_t which maximizes the loss for all possible estimate of Σ in the current confidence set Γ_t.

As showed in the pseudo code in Alg. 5, the adaptive algorithm proceeds at every iteration t according to the following steps: First, based on the estimate $\hat{v}_t$ and the confidence set given by Eq. 4.13, GOM computes for each arm the index $B_t(x)$ as the maximal statistically plausible loss in $x \in \mathcal{X}$. Then, GOM selects the arm with the largest index $B(x)$, $x_t = \arg\max_{x \in \mathcal{X}} B_t$, since this is the arm whose observed rewards would reduce the most the loss. The adaptive algorithm then pulls x_t twice and uses the observed rewards to refine the estimates of the variance vector $\hat{v}$, and respectively the confidence set Γ_t. These updated estimates will then be used for the $t+1$ iteration, where the same steps are repeated. Once the budget of available n pulls is consumed, the algorithm computes the least-squares estimator of the parameter θ^* and outputs the corresponding loss defined in Eq. 4.7.

2.4 Numerical Simulations

In Fig. 4.2 we illustrate the performance of the adaptive learning strategy GOM *(red line)* and compare it with the optimal static strategy that knowns the true variance given by Σ *(green line)*, and with a uniform strategy *(blue line)* which focuses the sampling on the d arms in $\mathcal{X}$ selected by the optimal allocation (that is, the arms that are the closest to forming an orthogonal basis). We consider an input set consisting of five vectors in $\mathbb{R}^2$, as depicted in Fig. 4.1, and we define a noise η having as diagonal of the covariance matrix the variance vector $v = [0.1, 0.4]$. In Fig. 4.2 we report the loss $L_n(\mathcal{A})$ (Eq. 4.7) multiplied by n. In fact, any static allocation strategy is expected to have a decreasing loss of the order of $O(1/n)$, thus in order to remove this trend and to emphasize the behavior of the different algorithms we plot $nL_n(\mathcal{A})$.

As illustrated in Fig. 4.2, the rescaled loss of the learning algorithm actually decreases from a performance similar to that of a uniform allocation on the d support points[4], down to the performance of the optimal allocation. This behavior suggests that as the budget grows, the loss of the learning algorithm tends to decrease as fast as for the optimal static allocation. These numerical results in this proof of concept ex-

[4]The comparison with the strategy that allocates the samples uniformly over all arms was not depicted in Fig. 4.2, since this strategy leads to a much larger loss.

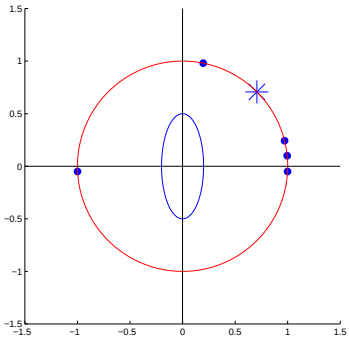


Figure 4.1: The input set $\mathcal{X}$, the covariance matrix (blue), and the θ^* vector (star).

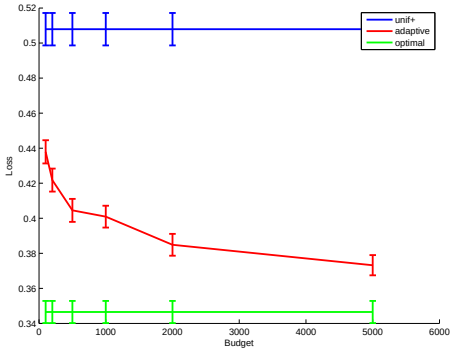


Figure 4.2: Rescaled loss $nL_n(\mathcal{A})$.

periment show that the learning algorithm GOM is effective in allocating the available budget n over arms to first estimate their variance and then to perform a nearly-optimal allocation.

2.5 Discussion

The design of an allocation strategy which approaches the optimal allocation in the particular case when the rewards are affected by multivariate heteroscedastic noise is useful for providing some first evidence on the behavior of adaptive sample strategies for OED in linear bandits. Nonetheless, even if the noise model enjoys useful properties (in particular in the case of multivariate Gaussian noise), the design of an appropriate sample allocation strategy for it remains a difficult problem. On the one hand, one needs to "invest" samples with the goal of learning the covariance matrix (a well-known difficult problem, as proven for instance, in [Bickel and Levina, 2008, Bien and Tibshirani, 2011]). On the other hand, one needs to mimic the behavior of the optimal allocation to minimize the regret of the allocation strategy.

It is also important to stress that since we are in a setting where all arms are dependent, then given the choice of the variance structure, the transformation of the noise according to the (known) arm features leads to a particular formulation of how useful/informative each arm is for the overall loss minimization. Before going any further with this problem formulation where an important part of the difficulty of creating arm indices comes from the estimation of the noise covariance matrix, we believe that it is interesting to investigate the design of heteroscedastic OED strategies in a simpler, but also more general noise structure. This would allow to understand more directly the challenges in constructing arm indices based on estimated noise variances, while also taking into account the global linear structure. Therefore, in the remaining part of the chapter, we consider a more general noise model, where the variance of the noise associated to each arm can be arbitrary, and is no longer parametrized by the features of the arm itself.

3 Adaptive OED with Heteroscedastic Noise

In this section, under a more general noise assumption, we will expose some observations, based mostly on empirical evidence, on the behavior of sample allocation strategies that need to rely on the estimation of the arm variances to achieve a certain optimality criterion fixed beforehand. As presented in Chap. 2, the design of efficient bandit allocation strategies is based on the construction of arm indices which need to be constructed in such a way to allow to identify at each time step the most "useful" arm, where the usefulness can come either from the information that the pull to the arm would reveal about the environment, or from the impact of the (expected) reward on the performance measure.

When the construction of arm indices depends on estimated arm variances and on top of this, there is a global structure making all arms dependent, then identifying the most useful arm is far for being trivial. In fact, as soon as arms are no longer independent (as in the MAB case), then there is no direct connection between the expected loss in one arm and the number of pulls that should be allocated to it in order to minimize the overall G-loss. Thus, for minimizing the G-loss in the heteroscedastic case, one can notice that the arms that are sampled more by the optimal strategy do not coincide with the arms that have the biggest losses (for a numerical example, see Sect. 5.1). On the other hand, in the case of the V-loss, the arm indices can be easily constructed. Therefore, to be able to recover reliable adaptive strategies based on arm indices, we now focus on the V-optimality design, where the effect of pulling an arm can be quantified by an arm score even in the heteroscedastic case.

3.1 Sequential V-Optimal Design

We focus here on strategies designed for reaching the V-optimality criterion, that is, sampling allocations that minimize the average prediction error over the points in the input space. We remind the formulation as an optimization problem

$$\begin{aligned} &\text{minimize } \frac{1}{K}\sum_{x\in\mathcal{X}} x^\top \left(\sum_{i=1}^{K}\lambda_i x_i x_i^\top\right)^{-1} x \\ &\text{subject to } \sum_{i=1}^{K}\lambda_i = 1; \lambda_i \geq 0, \forall i \end{aligned}$$

where λ is the vector of proportions of samples allocated over the K arms in $\mathcal{X}$. One can easily see that the optimal strategy for the V-opt criterion also leads to minimizing the sum of the prediction errors over all arms. Closely related as a goal with the G-optimal design, the V-opt allocation is also pertinent for similar applications, as pointed out in Sect. 1.3 of this chapter. In addition, for this heteroscedastic setting, we propose a sequential strategy which is based on the provably efficient arm indices introduced in [Wiens and Li, 2014]. Let us first present in detail the model.

3.1.1 Weighted Least Squares Estimation

Like in the previous sections, let $\mathcal{X} \subseteq \mathbb{R}^d$ be a finite set of bounded arms such that $|\mathcal{X}| = K$ and for any arm $x \in \mathcal{X}$, $||x|| \leq L$. When an arm x is chosen, a noisy realization of an unknown function f is observed. Now we consider the linear case where a random realization r from arm x is defined as

$$r(x) = f(x) + \varepsilon(x) = x^\top \theta^* + \eta, \tag{4.14}$$

where the noise term η has unknown variance $\sigma^2(x)$ and θ^* is an unknown vector characterizing the linear function f.

Let $\mathcal{A}$ be a sample allocation strategy which at each round $t = 1, \ldots, n$ chooses an arm x_t and observes the corresponding realization $r_t = x_t^\top \theta^* + \eta_t$. Let $\bar{x}_t = \frac{x_t}{\sigma(x_t)}$, $\bar{r}_t = \frac{r_t}{\sigma(x_t)}$ and $\bar{\eta}_t = \frac{\eta_t}{\sigma(x_t)}$ be the **weighted** versions of the chosen arm x_t, the observed reward r_t and the corresponding noise η_t. For any sequence of n arm-observation pairs let $\bar{A}_n = \sum_{t=1}^n \bar{x}_t \bar{x}_t^\top = \bar{X}_n^\top \bar{X}_n$ be the matrix of chosen arms weighted by their variance[5]. Also, let $\bar{b}_n = \sum_{t=1}^n \bar{r}_t \bar{x}_t = \bar{X}_n^\top \bar{R}_n$ be the vector of observed weighted samples and let $\bar{E}_n \in \mathbb{R}^n$ be the vector of weighted observed noise. Once $\mathcal{A}$ used a budget of n pulls, the resulting weighted least-squares (WLS) estimate of θ^* is

$$\bar{\theta}_n = \bar{A}_n^{-1} \bar{b}_n. \tag{4.15}$$

Like previously, for each arm $x \in \mathcal{X}$ we define the prediction error of $\bar{\theta}_n$ as the expected quadratic loss

$$L_n(x; \bar{X}_n) = \mathbb{E}[(x^\top \bar{\theta}_n - x^\top \theta^*)^2] \tag{4.16}$$

where the expectation refers to all possible sources of randomization in the observations r_t and in the choice of the sequence of arms $\mathbf{x_n} = (x_1, \ldots, x_n)$. However, we now define the overall performance of the algorithm $\mathcal{A}$ by the sum of the arm losses[6]:

$$L_n(\mathcal{A}) = \sum_{x \in \mathcal{X}} L_n(x). \tag{4.17}$$

In the sequel, we refer to the loss defined in Eq. 4.17 as the V-opt loss. Given a fixed budget n, the objective is to design a sample allocation strategy $\mathcal{A}$ that **minimizes** $L_n(\mathcal{A})$. An optimal sampling strategy, denoted $\mathcal{A}^*$, is defined as follows:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}} L_n(\mathcal{A}) = \arg\min_{\mathcal{A}} \sum_{x \in \mathcal{X}} L_n(x). \tag{4.18}$$

3.1.2 The Oracle Static Allocation

Consider a static[7] allocation strategy $\mathcal{A}^*$ which has access to the variance for each arm and can therefore perform the sample allocation strategy in the optimal way. Let

[5] We remind that $\bar{A}_n \in \mathbb{R}^{d \times d}, \bar{X}_n \in \mathbb{R}^{n \times d}$.

[6] Equivalently, we could have used the average of the arm losses instead of the sum for the overall loss definition.

[7] We remind that a *static* allocation is an allocation where the arm selection does not depend on the observed rewards.

$\bar{A}_n = \bar{\mathcal{X}}_n^\top \bar{\mathcal{X}}_n$ be the weighted design matrix obtained when sampling n times according to $\mathcal{A}^*$ and $\bar{b}_n$ the corresponding vector of rewards. The covariance of the WLS solution (Eq. 4.15) is computed as follows:

$$\begin{aligned}\mathbb{V}[\bar{\theta}_n|\bar{X}_n] &= \mathbb{E}\Big[(\bar{\theta}_n - \theta^*)(\bar{\theta}_n - \theta^*)^\top|\bar{X}_n\Big] \\ &= (\bar{X}_n^\top \bar{X}_n)^{-1}\bar{X}_n^\top \mathbb{E}[\bar{E}_n \bar{E}_n^\top]\bar{X}_n(\bar{X}_n^\top \bar{X}_n)^{-1} \\ &= (\bar{X}_n^\top \bar{X}_n)^{-1}\bar{X}_n^\top \bar{\Omega}_n \bar{X}_n(\bar{X}_n^\top \bar{X}_n)^{-1}, \end{aligned} \tag{4.19}$$

where $\bar{\Omega}_n = \text{diag}\,(\sigma^2(x_1)/\sigma^2(x_1), \ldots, \sigma^2(x_n)/\sigma^2(x_n)) = I_n$. Thus, the variance of the WLS solution becomes:

$$\mathbb{V}[\bar{\theta}_n|\bar{X}_n] = (\bar{X}_n^\top \bar{X}_n)^{-1} = \bar{A}_n^{-1}. \tag{4.20}$$

To make more explicit the components of the covariance matrix $\bar{A}_n$, for the case where the variances are known beforehand, we introduce the following rewriting:

$$\mathbb{V}[\bar{\theta}_n|\bar{X}_n] = (\bar{X}_n^\top \bar{X}_n)^{-1} = n^{-1}X(V^\top \Lambda V)^{-1}X^\top = n^{-1}C, \tag{4.21}$$

where Λ is a diagonal matrix indicating the pulls proportions to each arm ($[\Lambda]_{ii} = \lambda_i$) and we define $V = \Sigma^{-1/2}X$, where Σ is the diagonal matrix of weights $[\Sigma]_{ii} = 1/\sigma^2(x_i)$), and C is the V-opt covariance matrix is

$$C = X(V^\top \Lambda V)^{-1}X^\top. \tag{4.22}$$

Note that for the case when the variances are unknown, the weights will be determined based on variance estimates. Therefore, the efficiency of the allocation strategies will be determined by how close the estimate $\widehat{\Omega}_n$ is to the corresponding terms in the oracle allocation in Eq. 4.19 (equivalently, it will depend on how well estimated is $\widehat{\Sigma}$, in the rewriting in Eq. 4.21).

Going back to the oracle allocation, since $\mathcal{A}^*$ is static, then the previous expectations are conditioned on the fixed set of arms chosen over n rounds, which are summarized by $\bar{X}_n$. Therefore, the loss for every arm $x \in \mathcal{X}$ can now be expressed directly as

$$\begin{aligned} L_n(x; \bar{X}_n) &= \mathbb{E}[(x^\top \bar{\theta}_n - x^\top \theta^*)^2|\bar{X}_n] \\ &= \mathbb{E}\Big[x^\top(\bar{\theta}_n - \theta^*)(\bar{\theta}_n - \theta^*)^\top x|\bar{X}_n\Big] \\ &= x^\top \mathbb{E}\Big[(\bar{\theta}_n - \theta^*)(\bar{\theta}_n - \theta^*)^\top|\bar{X}_n\Big]x \\ &= x^\top \mathbb{V}[\bar{\theta}_n|\bar{X}_n]x \\ &= x^\top \bar{A}_n^{-1}x. \end{aligned} \tag{4.23}$$

The optimal allocation is the one that selects a sequence of arms such that $\bar{X}_n \in \arg\min_{\bar{X}_n} \sum_{x\in\mathcal{X}} L_n(x; \bar{X}_n)$. More precisely, let $\Lambda_n = \text{diag}(N_{1,n}/n, \ldots, N_{1,K}/n)$ be the matrix of pulls to the arms in $\mathcal{X}$ as given by an adaptive sample allocation strategy and denote λ_n be the corresponding pulls proportions after n samples. Also, denote $\widehat{V}_n = \widehat{\Sigma}_n^{-1/2}X_n$ obtained using the estimated weights after n samples. Following the

arm definition in Eq. 4.23 and taking the loss of all arms, the V-loss $L(\lambda_n)$ for this allocation is

$$L(\lambda_n) = \frac{1}{n} \sum_{i=1}^{K} x_i^\top (V^\top \Lambda_n V)^{-1} x_i. \tag{4.24}$$

Then, we have that the optimal allocation $\mathcal{A}^*$ over n samples is such that $L(\mathcal{A}^*) = \min_{\lambda_n} L(\lambda_n)$.

In the more realistic case where the arm variances are unknown, one has to design adaptive strategies that learn the arm variances from the observed rewards. It is therefore crucial to be able to construct arm indices which are able to guide the design, by pointing out the arms that allow to improve the information on the environment, or arms that lead to minimize the loss. Such arm indices for the V-optimization problem with heteroscedastic noise are given in [Wiens and Li, 2014]. In the following, we present the arm indices construction they proposed for an adaptive allocation design, then we build on their formulation to propose a new greedy arm selection strategy.

3.2 Arm Indices for Sequential V-Optimal Design

Since we now consider the case where the arm variances are not given in advance, any allocation algorithm must start with some exploration steps where the only goal is to obtain an estimate of the arm variances. The basic preliminary requirement is to sample each arm twice to initialize the estimations of the (possibly) different noise variance for each arm. Denote $n_0 = 2K$ the number of samples needed to initialize the variance estimate for each of the K arms[8].

Then, to ensure that the designed sampling strategy is reliable, a basic condition is that all arms are pulled frequently enough to obtain *consistent* variance estimates. A simple way to meet this requirement is to have a **forced exploration** phase. This is precisely what the sequential algorithm proposed in [Wiens and Li, 2014] does: from sample n_0 to n_{init}, the algorithm sequentially selects the most under-pulled arm. Define for each arm $x_i \in \mathcal{X}$

$$u_{i,t} = \frac{c}{K\sqrt{t}} - \frac{N_{i,t}}{t} \tag{4.25}$$

where t gives the number of observed samples, K is the number of arms, and $N_{i,t}$ is the number of times that arm $x_i \in \mathcal{X}$ has been pulled up to time step t. With respect to the forced exploration phase, we say an arm x_i is under-pulled if $u_{i,t} > 0$. At each time step $t < n_{\text{init}}$, we select the most under-pulled arm, that is, $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} u_{i,t}$, or, if there is no under-pulled arm (for some given constant c), then we select according to the index policy given in Eq. 4.27.

[8]For the settings where the support arms can be identified, we only compute the terms $u_{i,t}$ for the I support points, where $0 < I \leq K$ (and later on also only I variance estimates). However, we keep here the notation with K support arms initially considered, since we treat the general case where there is no prior information useful to identify the support points before observing rewards.

As shown in [Wiens and Li, 2014, Eq. 21], in addition to a typical condition on the noise of the observations (i.i.d. noise, with $\mathbb{E}[|\eta_i|^{4+\gamma}] = \upsilon < \infty$, for some $\gamma > 0$), in order to obtain consistent variance estimates, it is necessary that n_{init} satisfies

$$\lim_{n\to\infty} \frac{(\log n)^2}{n_{\text{init}}} = 0 \quad \text{and} \quad \lim_{n\to\infty} \frac{n_{\text{init}}}{n} = 0.$$

Then, as soon as all the samples for the forcing phase are consumed, the estimates of the variances are *frozen* and the allocation strategy for the remaining samples will be based on these estimates, that we denote by $\widehat{\sigma^2}_{1,N_{1,\text{init}}}, \ldots, \widehat{\sigma^2}_{K,N_{K,\text{init}}}$, where we have that $N_{i,\text{init}}$ is the number of times arm x_i was pulled in the initialization phase, and $\sum_{i=1}^{K} N_{i,\text{init}} = n_{\text{init}}$. For more precision, we specify that the empirical variance of an arm x_i at some time step t and is computed here as

$$\widehat{\sigma^2}_{i,t} = \frac{1}{N_{i,t} - 1} \sum_{j=1}^{N_{i,t}} (r_{i,j} - \widehat{\mu}_{i,t})^2, \tag{4.26}$$

an unbiased estimator built using the empirical mean $\widehat{\mu}_{i,t}$ of the rewards $r_{i,j}, \ldots, r_{i,N_{i,t}}$ observed from arm x_i up to time step t.

For the remaining sampling budget, the algorithm in Wiens and Li [2014], to which we will refer in the sequel as V-FORCED, allocates samples as summarized in the pseudo code given in Alg. 6.

Now, to get to the arm index construction, let us first introduce the notation for the empirical quantities presented earlier in the case of the oracle allocation. Let Λ_t be the diagonal matrix with $[\Lambda]_{ii} = N_{i,t}/t$ which captures the sample allocation up to time step t and let the estimate of the covariance matrix defined in Eq. 4.22 be $\widehat{C}_t = \mathcal{X}(\widehat{V}^\top \Lambda_t \widehat{V})^{-1} \mathcal{X}^\top$, where $\widehat{V} = \widehat{\Sigma}^{-1/2}\mathcal{X}$ is computed using the estimated variances obtained at the end of the forced exploration phase (that is, $[\widehat{\Sigma}^{-1/2}]_{ii} = 1/\widehat{\sigma^2}_{i,N_{i,\text{init}}}$). The adaptive algorithm in Alg. 6, computes at each time step t and for each arm x_i, the index

$$\alpha_{i,t} = [\widehat{C}_t^2]_{ii} / \sigma^2_{i,N_{i,\text{init}}} \tag{4.27}$$

and then selects at the next time step the arm satisfying $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} \alpha_{i,t}$.

The allocation strategy constructed in this way is proven to be asymptotically optimal. Nonetheless, the need to fix n_{init} before starting to allocate samples and the fact that one no longer takes advantage on the information brought by the observed rewards to refine the variance estimates after the forced exploration phase, leaves place for more adaptive and possible more practical algorithms. Building on the results in [Wiens and Li, 2014], particularly on their arm-index construction, in the following section we propose another adaptive allocation strategy, which requires only an initialization phase for the arm variances. Then, the strategy we propose greedily allocates the remaining sampling budget based on the sequentially refined confidence intervals for the arm variances.

Algorithm 6 V-Optimal Allocation with Forcing, [Wiens and Li, 2014]

Input: arm set $\mathcal{X}$, budget n, forcing phase budget n_{init}
Initialization phase:
Pull each arm twice and compute $\widehat{\sigma^2}_0(x_1), \dots, \widehat{\sigma^2}_0(x_K)$
Forced exploration phase:
for $t = n_0 + 1$ **to** n_{init} **do**
  Compute $u_{i,t}$ for all support points $x_i \in \mathcal{X}$ (Eq. 4.25)
  if $\exists x_i \in \mathcal{X} : u_{i,t} > 0$ **then**
    Pull arm $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} u_{i,t}$
    Update the variance estimate of the pulled arm
  else
    Compute $\alpha_{i,t}$ for all support points $x_i \in \mathcal{X}$ (Eq. 4.27)
    Pull arm $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} \alpha_{i,t}$
    Update $\widehat{C}_t$
  end if
end for
Adaptive allocation phase:
for $t = n_{init} + 1$ **to** n **do**
  Compute $\alpha_{i,t}$ for all support points $x_i \in \mathcal{X}$ (Eq. 4.27)
  Pull arm $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} \alpha_{i,t}$
  Update $\widehat{C}_t$
end for
Output: V-loss of the sample allocation: $L(\lambda_n) = \frac{1}{n} \sum_{i=1}^{K} x_i^\top (V^\top \Lambda_n V)^{-1} x_i$.

3.3 V-Adaptive Algorithm

Instead of using a forced exploration phase, by introducing the V-Adaptive algorithm we propose to construct arm indices based more directly on the uncertainty in estimating the arm variances after each time step. For this, we will rely on the construction of confidence bounds, based on the Chernoff-Hoeffding's bound on the variances of the arms. More precisely, after each time step t, the confidence bounds for the variance of an arm x_i are computed as

$$\sigma_{i,t}^{2\,UB} = \widehat{\sigma^2}_{i,t} + 3\sqrt{\frac{\log(1/\delta)}{2N_{i,t}}} \qquad \sigma_{i,t}^{2\,LB} = \widehat{\sigma^2}_{i,t} - 3\sqrt{\frac{\log(1/\delta)}{2N_{i,t}}}$$

where $\widehat{\sigma^2}_{i,t}$ is the empirical variance of arm x_i at time step t and is computed as shown in Eq. 4.26. We now explain how we modify the definition of $\alpha_{i,t}$ to obtain the more adaptive arm index for an arm x_i.

Starting from the arm index definition used by V-Forced (Eq. 4.27), we replace in the numerator the $\widehat{C}_t$ matrix based on the empirical variances, with the C_t^{UB} matrix where we use the upper-confidence bounds of the estimated variances. More precisely, let $V_t^{UB} = (\Sigma_t^{UB})^{-1/2}\mathcal{X}$, where Σ_t^{UB} is the diagonal matrix with $[\Sigma_t^{UB}]_{ii} = \sigma_{i,t}^{2\,UB}$, the upper-confidence bounds on the arm variances as estimated after t samples. Then we

obtain

$$C_t^{UB} = \mathcal{X}\left((V_t^{UB})^\top \Lambda_t \, V_t^{UB}\right)\mathcal{X}^\top, \tag{4.28}$$

where Λ_t is the diagonal matrix of the current sample allocation. In addition, in the denominator, instead of the empirical variance we use the lower-bounds on the variance of x_i to obtain upper-bounds the arm scores. This leads to the index

$$\widetilde{\alpha}_{i,t} = \frac{[C_t^2]_{ii}^{UB}}{\sigma_{i,t}^{2\ LB}} \tag{4.29}$$

for an arm $x_i \in \mathcal{X}$ and for each time step $t > n_0$. Lastly, given the arm indices thus constructed, we again choose to pull next the arm satisfying $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} \widetilde{\alpha}_{i,t}$. The pseudo-code of this V-ADAPTIVE sample allocation strategy is given in Alg. 7.

The arm-index construction in Eq. 4.29 allows to avoid the forced exploration phase and the need to fix in advance an appropriate threshold n_{init} used such that a certain minimum proportion of observations is allocated to each point. In fact, notice that in the definition of $\widetilde{\alpha}_{i,t}$ we have at the numerator the part of the arm-index that is affected by the global uncertainty over all points in $\mathcal{X}$, while the term in the denominator concerns exclusively the uncertainty in arm x_i. Whether an arm $x_i \in \mathcal{X}$ has the largest index $\widetilde{\alpha}_{i,t}$ can either driven by the uncertainty in estimating the arm-specific variance σ_i^2 (lower-bound in the denominator), or by their role in the global uncertainty.

Choosing at each step the arm $x_i \in \mathcal{X}$ that maximizes this index dependent on these quantities ensures that all arms will be pulled *infinitely often*. If this sampling condition is respected, then the V-ADAPTIVE strategy can obtain consistent variance estimates for all arms, while also adapting more to the observed rewards (since the variance estimates are updated after each time step). Therefore, we can expect this strategy to perform even better than V-FORCED which depends heavily on the length of the forced exploration phase.

Algorithm 7 ADAPTIVE V-OPTIMAL ALLOCATION

Input: arm set $\mathcal{X}$, budget n
INITIALIZATION PHASE:
Pull each arm twice and compute $\widehat{\sigma^2}_0(x_1), \ldots, \widehat{\sigma^2}_0(x_K)$
ADAPTIVE ALLOCATION PHASE:
for $t = n_0 + 1$ **to** n **do**
  Compute $\widetilde{\alpha}_{i,t}$ for all support points $x_i \in \mathcal{X}$ (Eq. 4.29)
  Pull arm $x_{t+1} = \arg\max_{x_i \in \mathcal{X}} \widetilde{\alpha}_{i,t}$
  Update the UB and LB on the variance estimate
end for
Output: V-loss of the sample allocation: $L(\lambda_n) = \frac{1}{n}\sum_{i=1}^K x_i^\top (V^\top \Lambda_n V)^{-1} x_i$.

In support of this observation, we evaluate the empirical performance of the V-ADAPTIVE algorithm (*red* lines in the following plots) through direct comparison with the optimal allocation, V-ORACLE algorithm (*blue* lines), that knows the true arm

variances in advance, and to V-FORCED, depicted by the *green* lines. In the following, for the most relevant examples presented in [Wiens and Li, 2014], we compare the loss obtained by V-ADAPTIVE, to the loss of V-FORCED, having several initialization budgets ranging from $O(\text{polylog}(n))$ to $O(n)$, where n is the total budget. We also compare to the optimal results of an oracle knowing the variances in advance. Additional experiments are reported in the final section of this chapter (Sect. 5.2).

3.3.1 Example 1: MAB

In this first example we test the algorithms in the MAB case. Thus, we suppose that the regressors are the canonical basis, here denoted as the group indicators e_i – the i-th column of $\mathcal{X} = I_d$, so that $\mathcal{X}$ is square and non-singular. Then the predictions do not depend on the regressors and we have that $C = \Sigma\Lambda^{-1}$, with $L(\lambda) = \sum_{i=1}^{d}(\sigma_i^2/\lambda_i)$. Note that in this setting the optimal allocations, implemented by theV-ORACLE strategy, are computed as

$$\lambda_i^* = \frac{\sigma_i}{\sum_{i=1}^{d}\sigma_i}. \tag{4.30}$$

It follows that the respective loss of the optimal allocation is

$$\min_{\lambda} L(\lambda) = L(\lambda^*) = (\sum_{i=1}^{d}\sigma_i)^2. \tag{4.31}$$

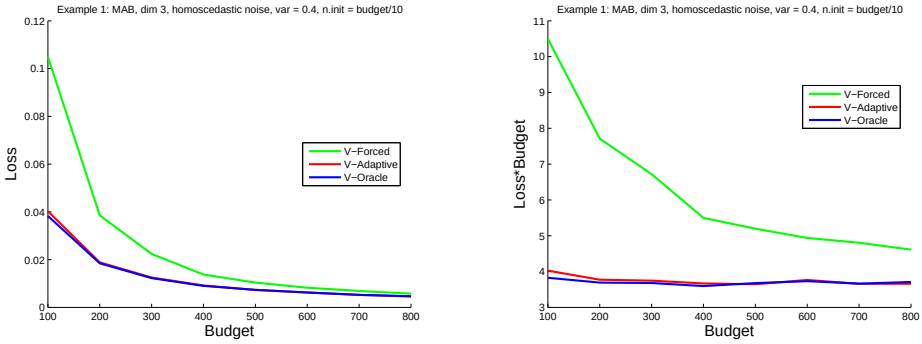
As pointed out in [Wiens and Li, 2014], this allocation is a special case of the results in [Pukelsheim and Torsney, 1991], where the construction of optimal design weights is studied in a similar V-optimal design setting with homoscedastic noise.

For the experiments in this setting, we consider both homoscedastic and heteroscedastic noise for a setting in $\mathbb{R}^3$. We compute the loss over sampling budgets going from 100 to 800 and we plot the losses averaged over 3000 runs.

Homoscedastic noise We consider first the scenario when for all three canonical arms have the same noise variance (here $\sigma^2 = 0.4$). The optimal allocation is to pull all three arms uniformly.

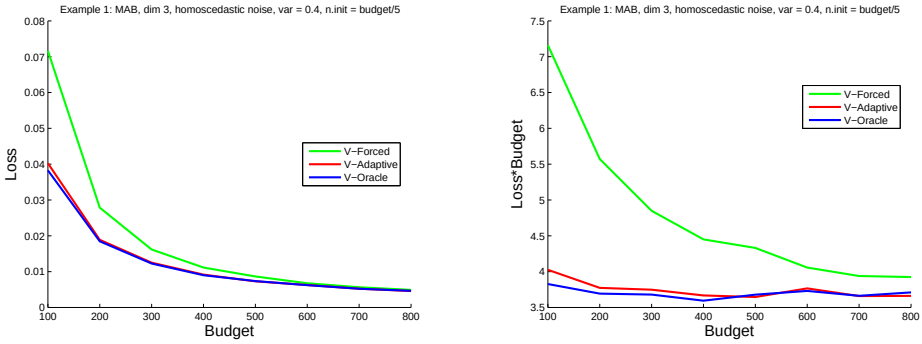
In Fig. 4.3 and Fig. 4.4 we compare the V-loss when the forced exploration phase of V-FORCED are $n/10$ and respectively $n/5$. The larger exploration rate in Fig. 4.4 allows V-FORCED to obtain a smaller loss, since the variance estimates are more refined compared to the *frozen* variances estimated used to compute the allocation whose loss is depicted in Fig. 4.3. Nonetheless, in both cases the performance of V-ADAPTIVE is better than V-FORCED, the former having an equal performance to that of the oracle as the total available budget increases.

Heteroscedastic noise The same behavior can be observed also in the case of MAB scenarios with heteroscedastic noise, depicted in Fig. 4.5. We consider here the case where the arm variances are: [0.16 0.04 0.04]. Here the optimal allocation is proportional to the standard deviation, thus the vector of optimal proportions of pulls to the arms is $\lambda^* = [0.5 \;\; 0.25 \;\; 0.25]$.



(a) V-loss (b) Loss*Budget

Figure 4.3: Example 1, MAB with homoscedastic noise, $n_{\text{init}} = n/10$



(a) V-loss (b) Loss*Budget

Figure 4.4: Example 1, MAB with homoscedastic noise, $n_{\text{init}} = n/5$

3.3.2 Example 2: 1-dimension problem

In the second example considered by the authors of [Wiens and Li, 2014], the assumption is that $d = 1$. In this case the V-loss will be computed as

$$L(\lambda) = ||x||^2 / \sum_{i=1}^{K} (\lambda_i x_i / \sigma_i^2).$$

Then, letting $\rho_i = \frac{x_i^2}{\sigma_i^2}$, it follows that in this case the V-loss is minimized by the choice of allocation:

$$\lambda_i^* = 1 \quad \text{if} \quad \rho_i = \max_{j=\{1,\dots,K\}} \rho_j$$
$$\lambda_i^* = 0 \quad \text{otherwise.}$$

We report in the following the loss for the strategy when there is an unique $x \in \mathcal{X}$ reaching $\max_k \frac{x_k^2}{\sigma_k^2}$ and the case when there are several arms reaching the maximum.

1 arm with $\max_\rho$ In the first setting for example 2, we choose $K = 3$ and the arms have values $\{1, 2, 3\}$, with respective arm variances $\{0.4, 0.4, 0.2\}$. In this case, arm x_3

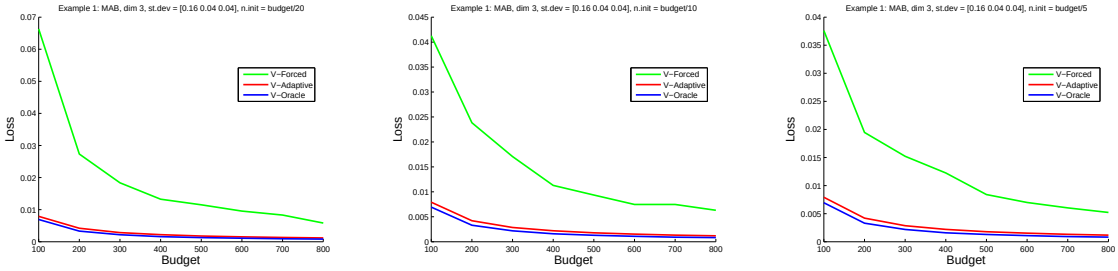
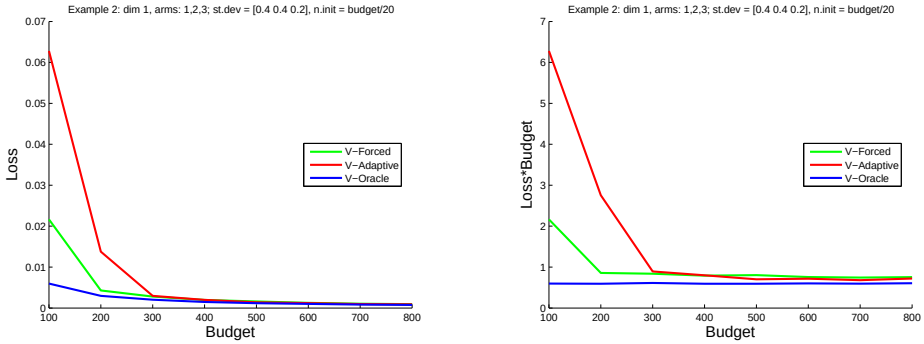


Figure 4.5: Example 1, MAB with heteroscedastic noise, V-loss for $n_{\text{init}} = \{n/20, n/10, n/5\}$

the one with the biggest α score. Therefore, this is the only arm that is going to be pulled by V-ORACLE and by the adaptive algorithms, once the exploration phase is over.



(a) V-loss (b) Loss*Budget

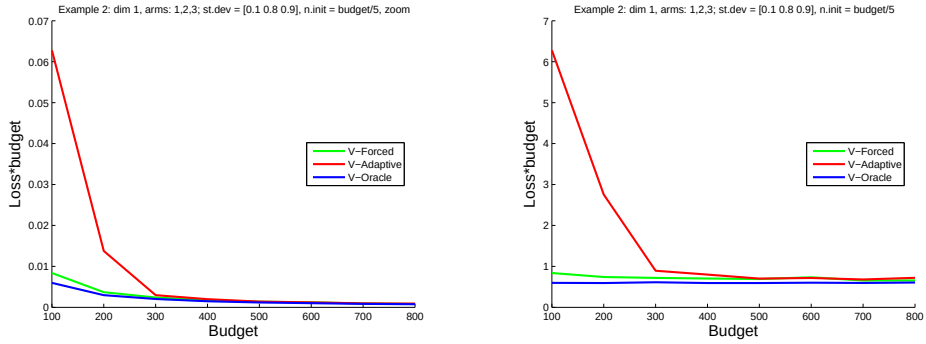
Figure 4.6: V-loss for Example 2 with 1 arm to be pulled and with $n_{\text{init}} = n/20$.

We illustrate the V-loss obtained by the three strategies. We can notice that V-FORCED performs in this setting very well, the initial forced exploration phase ($n_{\text{init}} = n/20$ in Fig. 4.6) allowing to learn which is the only arm that needs to be pulled, therefore the only loss comes from the exploratory phase. On the other hand, V-ADAPTIVE obtains a bigger loss for the cases with small budget, but the arm index construction allows to gather enough information to perform better than V-FORCED once the available budget is larger than 300 pulls.

More than d arms reaching $\max_\rho$ In this case the optimal allocation still focuses on only d arms: the ones that reach the $\max_\alpha$ and which have the smallest variance. This was verified on several settings.

Here we plot the results for the case when the arms are $\{1, 2, 3\}$ with respective variances $\{0.1, 0.8, 0.9\}$, which gives the scores $\alpha = \{10, 5, 10\}$. Even if two arms (or in general, more than d arms) have the same α, here only the first arm will be pulled by V-ORACLE, due to its smaller variance. We can see in Fig. 4.7 that the same ordering in the performance remains valid also for a larger initial exploration phase of length

$n/5$, with V-FORCED having a better performance for smaller budgets, but then V-ADAPTIVE rapidly recovers and for this simple problem, the allocation strategies reach the optimal V-loss.



(a) V-loss (b) Loss*Budget

Figure 4.7: Example 2, V-loss for $\alpha = \{10, 5, 10\}$ and $n_{\text{init}} = n/5$.

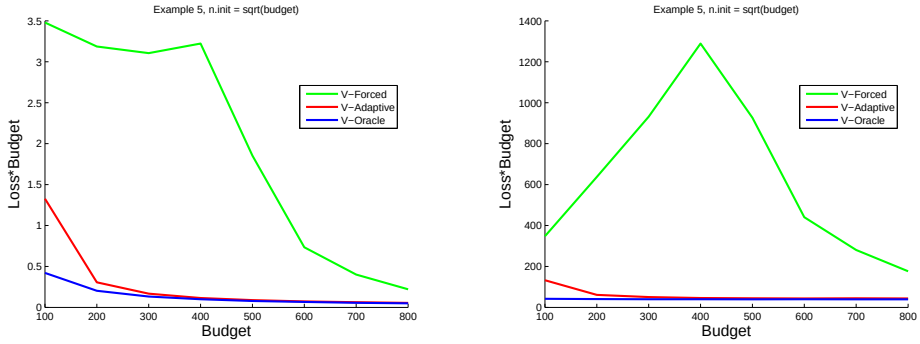
Examples 3 and 4 are meant to show how the optimal design for a given input set of 8 arms and dimension 4 can have from 4 up to 8 support points (depending on the features of the arms and their given variances). These examples are chosen by the authors of [Wiens and Li, 2014] to show that the optimal design can have from d up to K support points. In addition, only the loss of the oracle is reported for these settings, which are then slightly transformed to obtain Examples 5 and 6, on which we focus in the next subsection.

3.3.3 Examples 5 and 6

In both examples we consider a 4-dimensions problem, where the input space has 11 arms with features $(1, x, x^2, x^3)$, where $x \in \{-1, -0.8, -0.6, ..., 0.6, 0.8, 1\}$. In Example 5 the noise is homoscedastic ($\sigma^2 = 1$). Then, in Example 6, we keep the same setting, but with different variances for each arm. More precisely, the considered standard deviations for the arms in the input space are (preserving the ordering): $\sigma = \{0.7, 1.3, 0.1, 0.4, 0.4, 0.3, 0.3, 0.4, 0.2, 1.5, 1.2\}$.

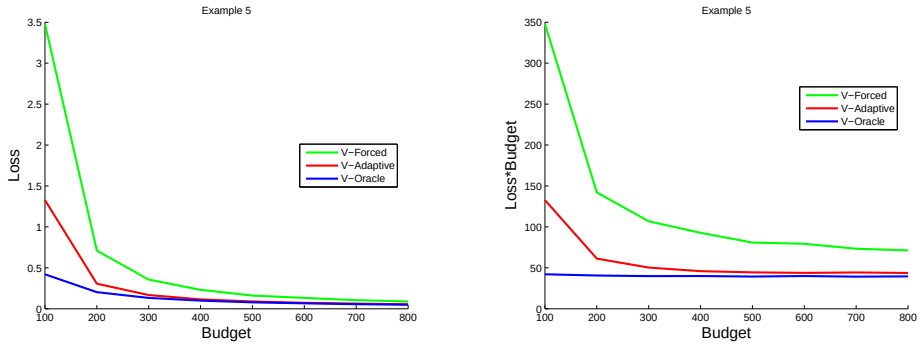
We report here the V-loss obtained by the three algorithms, with $n_{\text{init}} = \sqrt{n}$ (Fig. 4.8 for Example 5 and Fig. 4.10 for Example 6) and for $n_{\text{init}} = \log^2 n$ (Fig. 4.9 for Example 5 and Fig. 4.11 for Example 6). We again plot the V-loss averaged over 3000 runs and for total sampling budgets going from $n = 100$ to $n = 800$. Additional choices of n_{init} and the obtained losses for further settings are reported in the Appendix of this Chapter (Sect. 5).

Example 5 - Homoscedastic noise We can see that in the homoscedastic case using V-ADAPTIVE seems to lead to a lower V-loss. First because V-FORCED needs a larger sampling budget before approaching the optimal allocation, but more importantly, as we can see from the normalized loss in the *right* subfigures (b), because of



(a) V-Loss (b) Loss* Budget

Figure 4.8: V-loss for Example 5 when $n_{\text{init}} = n^{1/2}$

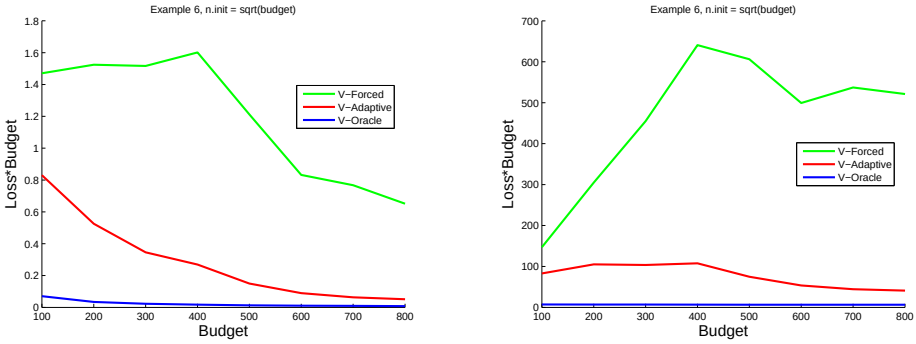


(a) V-Loss (b) Loss* Budget

Figure 4.9: V-loss for Example 5 when $n_{\text{init}} = \log^2(n)$

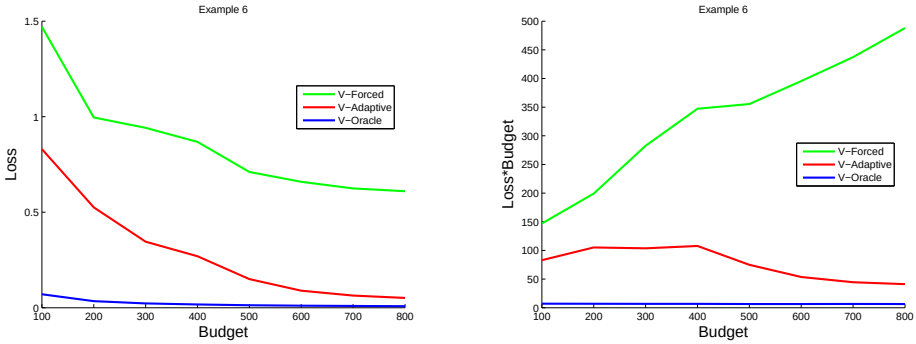
the fixed exploration phase, the gap with the optimal allocation seems to be hard to eliminate even for larger sampling budgets. On the other hand, V-ADAPTIVE is able to quickly obtain sufficiently good estimates and allocates the samples to reach the optimal loss starting from sampling budget $n = 300$.

Example 6 - Heteroscedastic noise For the heteroscedastic case, the same observations as for Example 5 remain valid and after comparing the losses of the two strategies, V-ADAPTIVE seems even more clearly better than V-FORCED. The difference is that, at least for these choices of n_{init}, the gap between the loss of V-FORCED and the loss of the optimal allocation remains significant even with the growth in the sampling budget. In this setting the performance of V-ADAPTIVE also keeps a gap with respect to the optimal loss even for the largest sampling budget $n = 800$, but this gap is reduced much more rapidly, as we can see in particular from the normalized losses illustrated in (subfigures (b)).



(a) Loss (b) Loss* Budget

Figure 4.10: V-loss for Example 6 when $n_{\text{init}} = n^{1/2}$



(a) V-Loss (b) Loss* Budget

Figure 4.11: V-loss for Example 6 when $n_{\text{init}} = \log^2(n)$

3.4 Future work on V-opt

The experimental results obtained for several settings, in particular for the examples with a heteroscedastic noise structure, suggest that the adaptive algorithm that we propose might indeed lead to an improvement in the V-loss, compared to the V-FORCED algorithm. An advantage of using V-ADAPTIVE is that one does not need to fix the length of the forced exploration phase in advance. Rather, the algorithm that we propose has a greedy selection strategy which only needs two pulls to each arm for the initialization of the variance estimates. In addition to not having to depend on the parameter n_{init}, another advantage of V-ADAPTIVE comes from the fact that it uses every observation to refine the variance estimates, therefore this sample allocation can potentially make a better use of the information from the observed rewards. To explore this information, we proposed to use the confidence intervals on the arm variances to construct arm indices that pull the arms which are more informative about the environment. Nonetheless, as future work, the empirical evidence we give here needs to be reinforced by analytical results showing the consistency and performance guarantees that can be obtained for this strategy.

4 Conclusion

In this chapter we made a few steps towards the study of adaptive sampling strategies for OED criteria applied in a stochastic linear bandit problems. We considered first the G-optimality criterion in a model with heteroscedasctic noise parametrized by the chosen arm. Then, for the V-optimality criterion, we considered a simpler and more general reward model, where the variance in the noise of each arm is arbitrary.

In both cases there are several aspects of the problem that transform the design of an adaptive allocation into a challenging task. First, the sample allocation needs to be made while also estimating the unknown, (possibly) different arm variances. This uncertainty makes it difficult to define which arms should be support points for the design and leads to an unavoidable "waste" of samples on uninformative arms. Furthermore, even if the different arm variances are known, given the global structure of the problem and the interdependency of the arms, defining an arm index that directly relates to the informativeness of the arm or to the benefit it would bring in optimizing some of the OED criteria remains non-trivial.

Consider for instance the G-optimal criterion with heteroscedastic (non-parametric) noise. It is still an open question whether one can design arm indices such that an optimistic choice leads to a proper sample allocation strategy for estimating the arm-variances and at the same time for minimizing the G-loss. Since every arm is dependent on the others, the "impact" over the global uncertainty obtained by observing a reward coming from a certain arm is difficult to define. Likewise, the use of the confidence interval for designing an allocation strategy is in this case less direct than in the case of the V-optimality criterion seen in Sect. 3. If one tries to construct arm indices similar to the ones proposed for the V-Adaptive allocation strategy, based on the lower and upper bounds on the variance estimates, one cannot easily control the sequential exploration and the improvement over the information needed to minimize the G-loss (whose goal is to minimize the largest prediction error over the arms). For instance, an apparently intuitive and conservative choice of an adaptive allocation chooses the support points based on the lower-bounds on the variances, then allocates to these identified support points based on their upper-confidence bounds and corresponding impact in maximizing the G-loss. Simple examples show that the sampling can get "stuck" on pulling only the arm that seems to be useful, but whose pull does not actually reducing the overall uncertainty or obtaining allocation proportions close to the optimal ones. The ambiguity in constructing an arm index indicating how much the arms should be pulled for a G-optimal allocation with heteroscedastic noise remains thus an open problem.

The chapter certainly leaves an important number of open problems. The immediate ones concern the analysis of the allocation strategies we proposed for the G-allocation with heteroscedastic multivariate noise in Sect. 2 and for the heteroscedastic adaptive V-optimal allocation we give in Sect. 3 are for which the empirical comparison to previously proposed algorithms seems encouraging. More in general, a question that follows for the preliminary study we make here concerns the limitations of the

allocation strategies exploring the confidence intervals of the estimates to construct arm indices useful for adaptive allocations strategies. Their applicability to other performance measures different that the traditional cumulative regret is not always direct and new tools seem necessary to make these methods more applicable also for different types of optimality criteria.

In the following chapter we move from the study of methods to be transfered from bandit literature to solve OED criteria, to the study of the potential advantages of transferring from one linear bandit problem to another linear bandit problem defined on the same environment.

5 Appendix

5.1 Optimal Allocation for Heteroscedastic G-loss

In this section we report numerical simulations showing the behaviour of the optimal static allocation for minimizing the heteroscedastic G-loss. The optimal allocation, uses the true values of the variances and finds the optimal allocation proportions over the arms. Let us first introduce the definition of the G-optimal allocation, using the notation and derivations previously introduced in Sect. 3 for the V-allocation.

Definition. The loss in each arm, as in the case of V-opt, is given by the prediction error of the weighted least-squares estimate, that is

$$L_n(x) = \mathbb{E}[(x^\top \bar{\theta}_n - x^\top \theta)^2]$$

where the expectation refers to all possible sources of randomization in the observations r_t and in the choice of the sequence of arms $\mathbf{x_n} = x_1, \dots, x_n$. Now, we define the G-opt performance of the algorithm $\mathcal{A}$ by the loss corresponding to the worst estimated arm, that is $L_n(\mathcal{A}) = \max_{x \in \mathcal{X}} L_n(x)$. Given a fixed budget n, the objective is to design a sample allocation strategy $\mathcal{A}$ that **minimizes** $L_n(\mathcal{A})$. Then, an G-optimal sampling strategy, denoted $\mathcal{A}^*$, is defined as follows:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}} L_n(\mathcal{A}) = \arg\min_{\mathcal{A}} \max_{x \in \mathcal{X}} L_n(x) = \arg\min_{\mathcal{A}} \max_{x \in \mathcal{X}} \mathbb{E}\left[(x^\top \bar{\theta}_n - x^\top \theta)^2\right].$$

As for the V-opt strategy, since the optimal allocation is static and we use the true variances, then in the case of the G-opt, we can also express directly the loss as in the case of a strategy that uses the true variances, then the G-loss for arm $x_i \in \mathcal{X}$ can be re-written as $L_n(x; \bar{X}_n) = x^\top \bar{A}_n^{-1} x$, where $\bar{A}_n = \bar{\mathcal{X}}_n^\top \bar{\mathcal{X}}_n$ (see Eq. 4.23). Lastly, note that for the allocation using the optimal proportions given by λ^*, we also have that $\bar{A}_n = \sum_{i=1}^K n\lambda_i^* x_i x_i^\top$.

Setting. For these numerical simulations, we use a 3-dimensional setting, with 10 arms that have different noise variance and ℓ_2 norms, with features as shown in Fig. 4.12. We report here the variance of each arm, as well as the optimal proportions of pulls, as allocated to the arms by a strategy that minimizes the G-loss. We computed the optimal allocation using the CVX convex optimization solver [Grant and Boyd]. For clarity, we show in the next three columns how the samples were allocated for after using a budget n of $\{700, 800, 900, 1000\}$ and a naive rounding procedure that computes $N_{i,n} = \lceil n\lambda_i^* \rceil$, where λ_i^* gives the optimal allocation proportion for arm $x_i \in \mathcal{X}$ and $N_{i,n}$ is the (integer) number of allocation used for arm x_i, given budget n. More importantly, in the last column in Fig. 4.12, we report the loss of each arm, computed as in Eq. 4.23.

Results. As we can notice, 5 of the 10 arms are support points for allocation λ^*, that is have at least 1 arm pull allocated. On the other hand, we can see that the maximal

prediction error over the arms, 1.0749, whose value also gives the G-loss, is attained in 4 of the arms in the input set. Interestingly, the arm x_9 reaches the G-loss, but is not a support point. It follows that we cannot use a simple arm index policy which supposes that the more an arm has a high associated G-loss, the more it should be pulled. It is therefore not immediate to see how to compute the optimal number of pulls allocated in a heteroscedastic G-loss and to compute an arm score that would guide an adaptive allocation strategy (with unknown arm variances) to mimic the behaviour of the G-opt allocation λ^* and to obtain similar values for the G-loss.

Arms	**Var.**	$\boldsymbol{\lambda^*}$	**n = 700**	**n =800**	**n =900**	**n= 1000**	$\mathbf{x^\top \bar{A}^{-1} x}$
$x_1 = [1 \;\; 1 \;\; 0]$	0.2	**0**	0	0	0	0	0.3363
$x_2 = [0 \;\; 2 \;\; 2]$	0.2	**0.16301**	114	132	148	164	0.67595
$x_3 = [3 \;\; 0 \;\; 3]$	0.2	**0.16585**	116	132	150	166	1.0749
$x_4 = [2 \;\; 1 \;\; 3]$	0.4	**0**	0	0	0	0	0.77303
$x_5 = [1 \;\; 1 \;\; 1]$	0.4	**0**	0	0	0	0	0.13392
$x_6 = [1.7 \;\; 1.9 \;\; 0]$	0.5	**0.3146**	220	252	284	316	1.0749
$x_7 = [0.5 \;\; 2.3 \;\; 2]$	0.3	**0**	0	0	0	0	0.58223
$x_8 = [3 \;\; 0.8 \;\; 1]$	0.3	**0.096074**	68	78	88	96	1.0038
$x_9 = [2.2 \;\; 1.8 \;\; 3.6]$	0.4	**0**	0	0	0	0	1.0749
$x_{10} = [0 \;\; 3.5 \;\; 1.5]$	0.5	**0.26047**	184	210	236	262	1.0749

Figure 4.12: Example of the optimal Heteroscedastic G-allocation proportions in a 3 dimensional setting with 10 arms. We can notice that the arms with the largest losses are not necessarily among the support points.

5.2 Additional Experiments for V-Adaptive

We report here additional experimental results for several V-optimal allocation strategies. These results are a complement to the same examples presented in Sect. 3.3, for different choices of variances or different n_{init} set for the initialization phase of V-FORCED. The experimental conditions (available budget, number of runs) remain the same as in Sect. 3.3, unless specified otherwise for some of the cases.

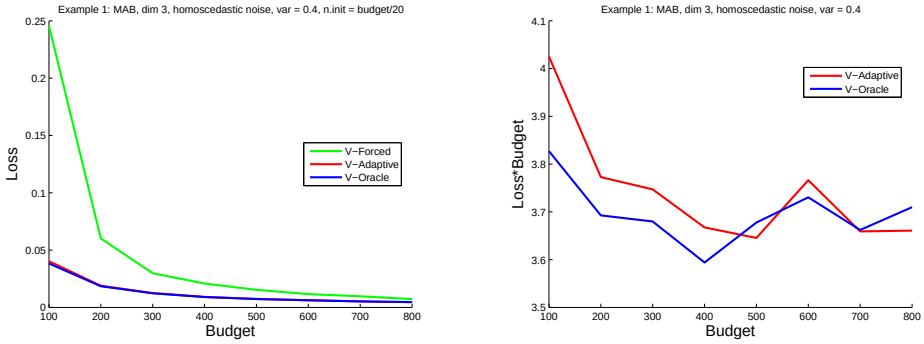
5.2.1 Example 1: MAB

Homoscedastic noise Fig. 4.13 depicts the V-loss for the three sample allocation strategies, under the same conditions as in Sect. 3.3.1, but for the case where the forced exploration phase has length $n/20$.

Heteroscedastic noise

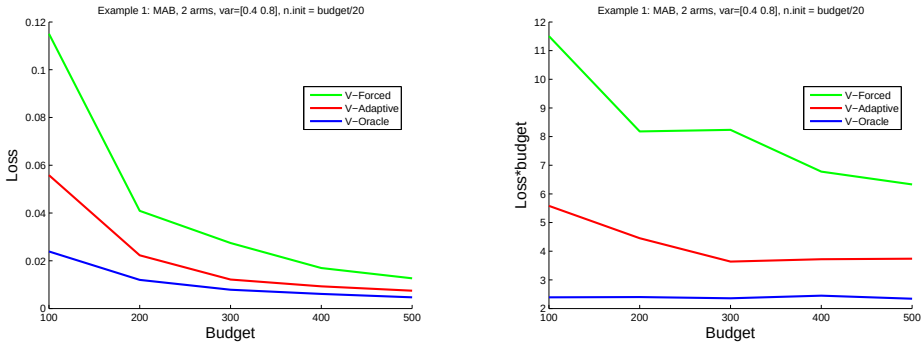
a. 2 arms with variances 0.4 and 0.8 For the MAB example with heteroscedastic noise we first present in Fig. 4.14 the case of a MAB in $\mathbb{R}^2$ with corresponding arms variance 0.4 and respectively 0.8. We can see that once again V-ADAPTIVE manages to use the observed rewards better than V-FORCED and thus obtains a smaller V-loss.

b. 3 arms with variances $[0.64 \;\; 0.16 \;\; 0.16]$



(a) V-loss (b) Loss * Budget

Figure 4.13: $n_{\text{init}} = n/20$



(a) V-loss (b) Loss*Budget

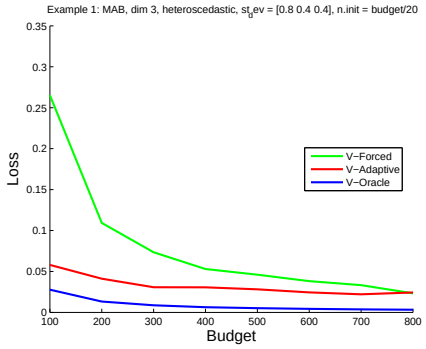
Figure 4.14: V-loss for MAB, 2 arms, $n_{\text{init}} = n/20$

Then, for the case of MAB in $\mathbb{R}^3$, we consider a case similar to the one in Fig.4.5 in the sense that they share the same optimal allocation λ^*, but in the case considered here the arms have higher variance: $[0.64 \;\; 0.16 \;\; 0.16]$. We show the V-loss using budgets from $n = 100$ to $n = 800$ and with an exploration phase for V-FORCED $n_{\text{init}} = n/20$ in Fig. 4.15(a), $n_{\text{init}} = n/5$ in Fig. 4.15(b), and $n_{\text{init}} = \sqrt{n}$ in Fig. 4.16. In this setting, for all the choices of n_{init} that we tested the V-FORCED algorithm performed worse than V-ADAPTIVE, the allocation using the *frozen* variance estimates leading to a greater loss, in particular for small budgets.

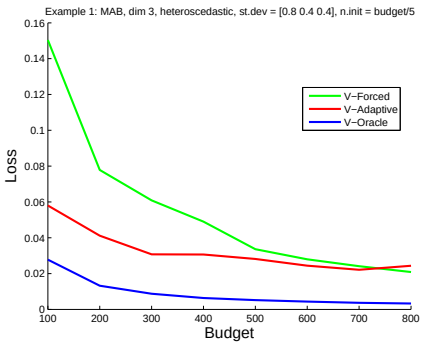
5.2.2 Example 2: 1-dimension problem

Here we present some additional plots comparing the V-loss for the same two settings considered in Sect. 3.3.2 for example 2.

1 arm with $\max_\rho$ We can see in Fig. 4.17 that the same ordering in the performance remains valid also for a larger initial exploration phase of length $n/5$, with V-FORCED having a better performance for smaller budgets, but then V-ADAPTIVE rapidly recovers and for this simple problem, the allocation strategies reach the optimal V-loss.



(a) V-loss when $n_{\text{init}} = n/5$



(b) V-loss when $n_{\text{init}} = n/5$

Figure 4.15: V-loss for Example 1 when $n_{\text{init}} = n^{1/2}$ and $\sigma^2 = [0.64, 0.16, 0.16]$

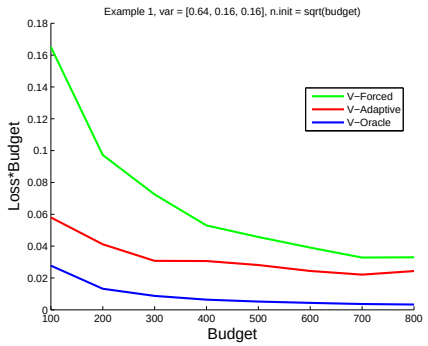


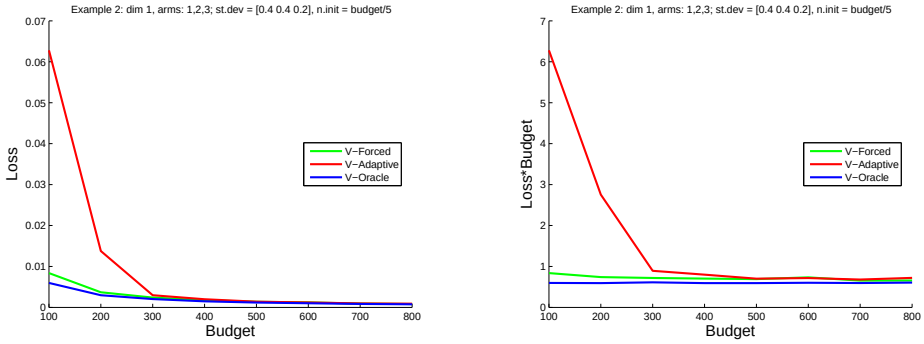
Figure 4.16: V-loss for Example 1 when $n_{\text{init}} = n^{1/2}$ and $\sigma^2 = [0.64, 0.16, 0.16]$

More than d arms reaching $\max_\rho$ We consider here the same setting as described for the plot in Fig. 4.7, but we illustrated the behaviour of V-FORCED for a lower exploration rate of $n/20$ (Fig. 4.18) and we also zoom on the respective V-losses that the strategies obtain for total sampling budgets between 500 and 800 (Fig. 4.19).

5.2.3 Examples 5 and 6

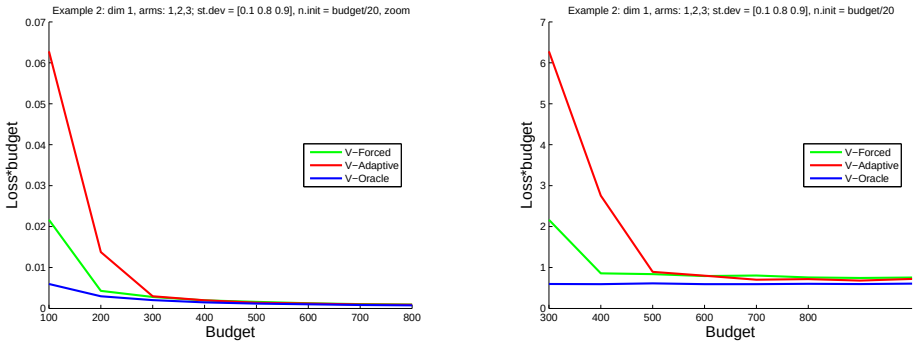
We report the V-loss obtained by the three algorithms, in several settings similar to the one already described in the Sect.3.3.3. The differences in each setting are stated directly in the names of the figures or in their description. Nonetheless, the results lead to the same general conclusion and comparison between V-ADAPTIVE and V-FORCED, showing that V-ADAPTIVE leads to a better performance for a wider range of length in the exploration phased used by the algorithm proposed in [Wiens and Li, 2014].

Example 5 - homoscedastic noise In addition to the comparison of the empirical performance provided earlier in Sect. 3.3.3, we report here the performance (the V-loss and the V-loss normalized) for additional values of n_{init}, ranging from $n/20$ in Fig. 4.20, to $n^{1/3}$ in Fig. 4.22, and $\log^3(n)$ in Fig. 4.23.



(a) V-loss (b) Loss* Budget

Figure 4.17: V-loss for Example 2 with 1 arm to be pulled and with $n_{\text{init}} = n/5$.

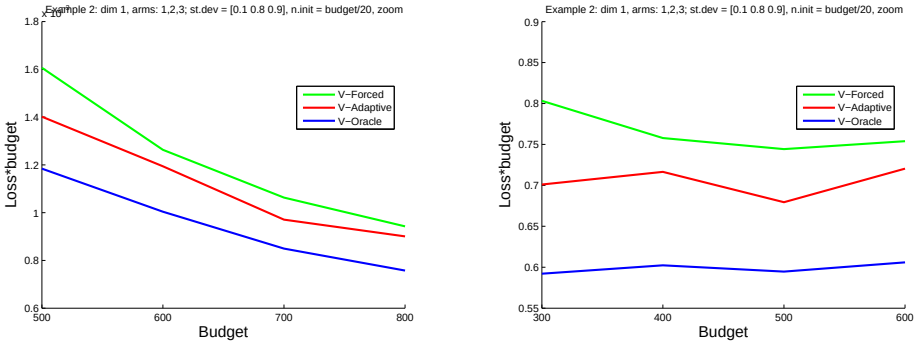


(a) V-loss (b) Loss*Budget

Figure 4.18: Example 2, V-loss for $\alpha = \{10, 5, 10\}$ and $n_{\text{init}} = n/20$.

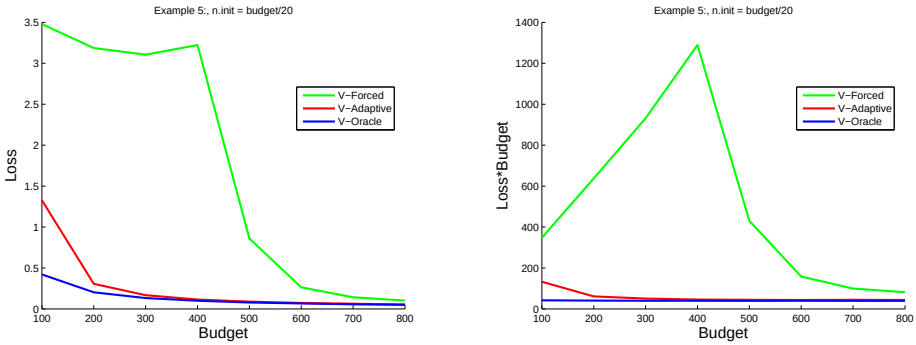
Example 6 - heteroscedastic noise Similarly to the results provided for Example 5, in complement to the comparison provided earlier in Sect. 3.3.3 with the V-FORCED algorithm, we report here the performance (the V-loss and the V-loss normalized) for additional values of n_{init}, ranging from $n/20$ in Fig. 4.24, to $n/5$ in Fig. 4.25, $n^{1/3}$ in Fig. 4.26, and $\log^3(n)$ in Fig. 4.27.

Also, we report the results when the different variances of the arms are: $\sigma^2 = \{0.7, 1.3, 0.1, 0.4, 0.4, 0.3, 0.3, 0.4, 0.2, 1.5, 1.2\}$. We illustrate the average loss over 1000 runs. All other parameters stay as previously. Lastly, for this specific setting, we also report the results when the budget grows from 10 to 100, as reported also in [Wiens and Li, 2014, Fig.1].



(a) V-loss (b) Loss*Budget

Figure 4.19: Example 2, V-loss for $\alpha = \{10, 5, 10\}$ and $n_{\text{init}} = n/5$, zoomed over $n = 500$ to $n = 800$.



(a) V-loss (b) Loss*Budget

Figure 4.20: V-loss for Example 5 when $n_{\text{init}} = n/20$.

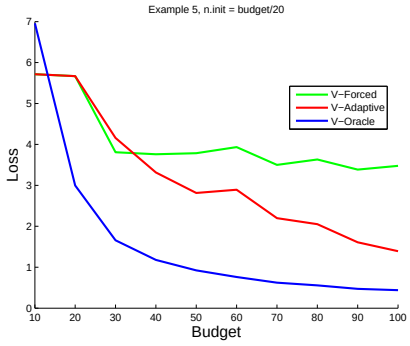
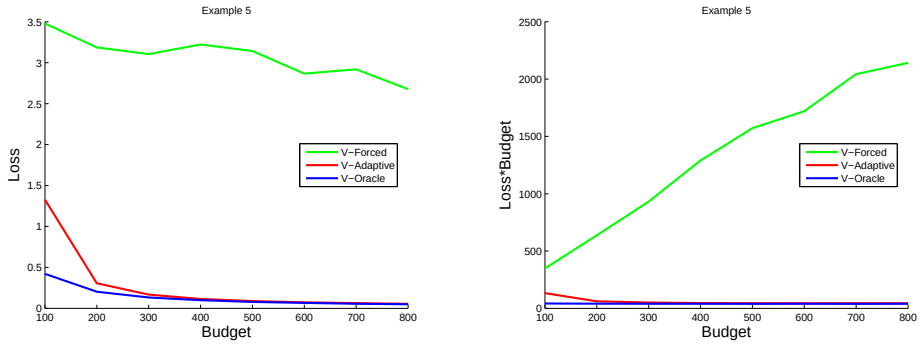
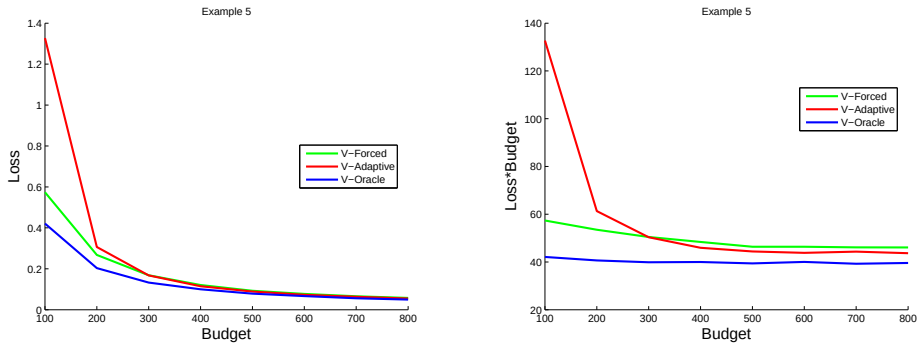


Figure 4.21: V-loss for Example 5 when $n_{\text{init}} = n/20$, and the budget ranges from 10 to 100.



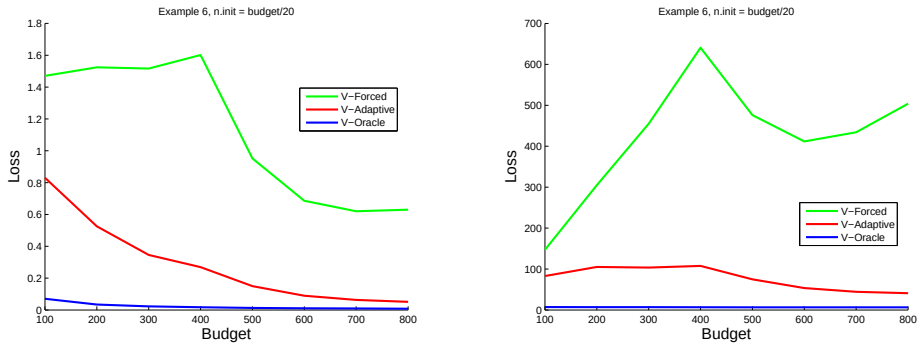
(a) V-loss (b) Loss* Budget

Figure 4.22: V-loss for Example 5 when $n_{\text{init}} = n^{1/3}$.



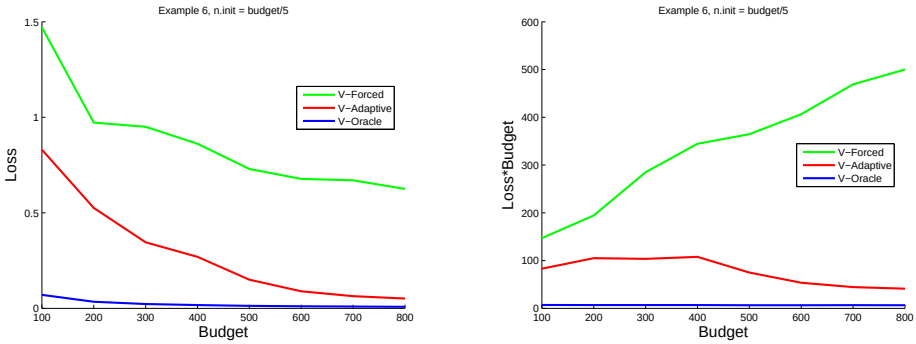
(a) V-loss (b) Loss* Budget

Figure 4.23: V-loss for Example 5 when $n_{\text{init}} = \log^3(n)$.



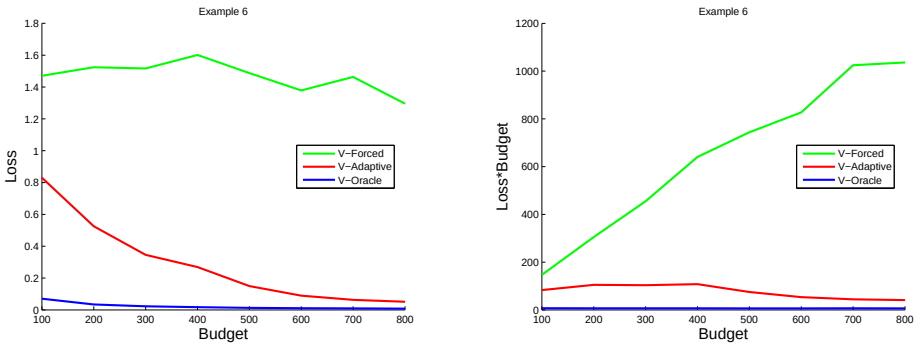
(a) V-loss (b) Loss* Budget

Figure 4.24: V-loss for Example 6 when $n_{\text{init}} = n/20$.



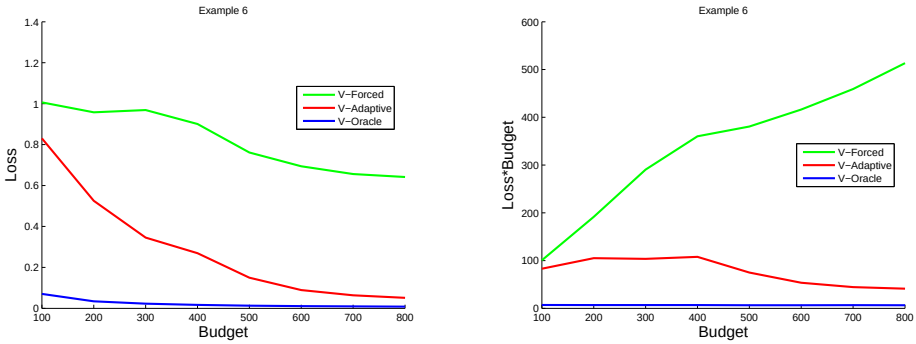
(a) V-loss (b) Loss* Budget

Figure 4.25: V-loss for Example 6 when $n_{\text{init}} = n/5$.



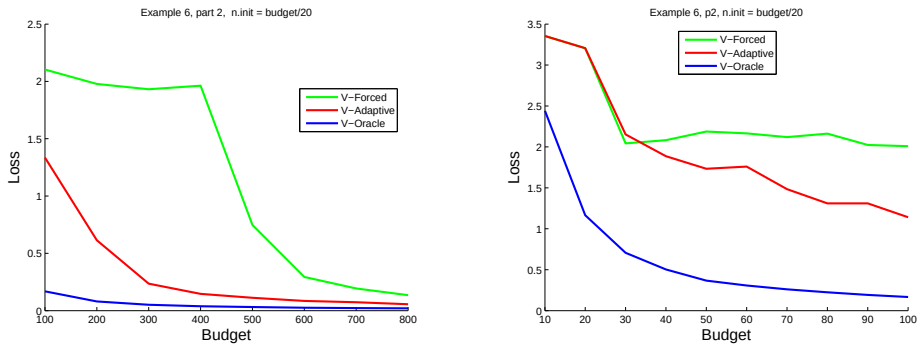
(a) V-loss (b) Loss* Budget

Figure 4.26: V-loss for Example 6 when $n_{\text{init}} = n^{1/2}$.



(a) V-loss (b) Loss* Budget

Figure 4.27: V-loss for Example 6 when $n_{\text{init}} = \log^3(n)$.



(a) V-loss (b) Loss*Budget

Figure 4.28: V-loss for different variances of the arms in Example 6 when $n_{\text{init}} = n/20$.

Chapter 5

Sequential Transfer of Samples in Linear Bandits

We have seen in the previous chapters resource allocation strategies designed with the goal of using the sampling budget both for gaining information about the unknown environment and to minimize a predefined regret function. The underlying common assumption was that the budget is sufficiently large to allow, when allocated in an efficient manner, to reach an exploration-exploitation trade-off that mimics the behavior of oracle strategy, which does not need to allocate resources for exploration.

On the other hand, there are many problems where the available amount of resources is very limited, or there is an important cost and risk associated to observing the result of an experiment. To tackle the cases where there is limited available data, in machine learning there have been developed transfer and multi-task methods able to exploit the knowledge obtained in prior tasks. While the potential gains of these approaches have been well studied in batch learning scenarios, the transfer learning setting has received far less attention for sequential decision making problems with limited feedback.

In this chapter[1], we study the sequential transfer of samples scenario, where each task is a linear stochastic bandit problem and the objective is to improve the per-task regret by transferring relevant information from previous tasks. We propose multi-task algorithms that meet this criterion, and provide a theoretical analysis of their performance, as well as preliminary empirical results.

Contents

[1]This chapter is a joint work with Ouais Alsharif, Alessandro Lazaric, and Joelle Pineau. A part of it was presented in [Soare et al., 2014a].

1 Introduction

In order to be able to learn from only a few samples, a learning algorithm needs to exploit some structural bias in the environment of the task at hand. Such bias can be introduced in the learning algorithms in many different ways, from Bayesian priors to regularization (e.g., smoothness, sparsity). One particularly successful way to define such bias is to learn it directly from multiple different, yet related, tasks. While this approach, known as multi-task or transfer learning (see [Pan and Yang, 2010] for a survey), has made significant gains in supervised learning scenarios (see e.g., [Collobert and Weston, 2008]), it has received far less attention for sequential decision making problems with limited feedback. In particular, while multi-task learning has been studied in batch reinforcement learning (RL) and in the case of transfer from one source to a target task (see [Lazaric and Restelli, 2011, Taylor and Stone, 2009] for a survey), a formal study on how *online* RL algorithms could benefit from transfer is mostly unexplored.

A notable exception is the work in [Brunskill and Li, 2013], where a two-phase multi-task RL approach is studied. At the end of the first phase, where single-task learning is run, the data observed across multiple tasks are clustered to *discover* the set of underlying MDPs. This knowledge is then exploited to reduce the sample complexity of learning in new tasks, under the assumption that tasks are drawn from the same distribution over a finite number of MDPs. A similar scenario is considered in the *multi-armed bandit* setting in [Gheshlaghi Azar et al., 2013], where bandit tasks are sequentially drawn from a finite distribution. The proposed solution is more sophisticated in that it allows to *continuously* learn and transfer, with a provable reduction in the per-task regret as the set of problems is estimated more and more accurately.

More recently, in [Gentile et al., 2014, Maillard and Mannor, 2014] it was also assumed that all tasks come from a finite set of distributions. Without initially knowing the number of distributions, the goal is to explore the graph structure of the space in [Gentile et al., 2014], or respectively, the existing latent variables in [Maillard and Mannor, 2014] to cluster the tasks according to their similarity. The common assumption is that the true number of reward distributions is much smaller that the number of tasks. Then once the clusters are identified, the transfer of information is done only between tasks belonging to the same cluster, which, provided the inferred clusters are correctly constructed, avoids the introduction of transfer bias. The method proposed in [Gentile et al., 2014] starts by assuming that all tasks belong to the same cluster, that is, they start from a complete graph structure. Then using the observed rewards, they delete the edges between points whose ℓ_2 difference is above a certain threshold. On the other hand, the method in [Maillard and Mannor, 2014] constructing and comparing UCBs on arm values to decide whether the generating distributions belong to the same class.

In this chapter we introduce an alternative transfer approach adapted to the *linear bandit* setting and study its properties. Whereas in the multi-arm bandit the goal is to learn the reward (e.g., a click through rate or a star ranking) of different arms (e.g.,

items), in the linear bandit, each arm is a feature vector and the reward function is assumed to be a linear combination of the feature vector with an unknown parameter vector θ (e.g., a vector characterizing the preference of a user). Solving this problem requires finding a suitable balance between choosing arms that can contribute to better learn the parameter vector (e.g., learn user's preference) and selecting arms that yield higher reward. The introduction of transfer learning in this setting aims to exploit the fact that if two users have similar parameter vectors, then knowledge of one's interactions can be exploited to minimize the *regret* of a learning algorithm while interacting with the other.

Similarity definition. In any multi-task/transfer scenario, it is crucial to define a suitable notion of *similarity* between tasks that can be exploited by the multi-task learner. Here we assume that two tasks are similar whenever their parameter vectors are close to each other (in a ℓ_2-norm sense). A similar notion of task relatedness was considered in batch learning settings, for the problem of transfer from multiple sources [Crammer et al., 2008, Lazaric and Restelli, 2011] and for curriculum learning of multiple tasks [Pentina et al., 2015]. This definition of similarity allows every new task to be different than previous ones. In this sense, our setting is more general than previous bandit transfer algorithms [Brunskill and Li, 2013, Gentile et al., 2014, Gheshlaghi Azar et al., 2013, Maillard and Mannor, 2014] where all tasks are sampled from a finite set.

Transfer mecanism. The primary goal of this chapter is to explore how multi-task learning can contribute to improving the performance of linear bandit algorithms. For this, we focus on the simple *transfer of samples* setting, where samples observed in past tasks are directly used in learning the optimal arm in each subsequent task.[2] Transfer of samples was shown to be effective in reducing the learning complexity and improving the empirical performance in a variety of batch settings such as multi-source supervised learing [Crammer et al., 2008], batch-RL [Lazaric and Restelli, 2011, Lazaric et al., 2008], and object detection [Lim et al., 2011], and thus it is a natural candidate for transfer in the bandit setting.

Contributions. This chapter makes several technical contributions. First, we propose a simple learning algorithm, Multi-Task Linear UCB, that extends LinUCB [Li et al., 2010] by constructing upper-confidence bounds based on both a single-task estimate and a multi-task estimate. In doing so, we avoid the problem of *negative transfer* that can plague multi-task learners. Furthermore, we derive a regret analysis showing that the per-task regret can be reduced by leveraging samples from similar tasks. Our analysis also reveals critical aspects of the transfer of samples in a sequential learning setting. Unlike in batch settings, the samples are not i.i.d. from a distribution fixed across tasks (see e.g. [Crammer et al., 2008]), but a learning algorithm keeps adapting the sampling strategy across tasks. We show how this may negatively affect the bias introduced by the transfer of samples from different tasks. Furthermore,

[2]Note that each linear bandit problem can also be seen as a new task. In this sense, since we consider a sequence of linear bandit problems, we also refer to the current setting as the *multi-task* linear bandit problem.

we also show how estimating such a bias may not even be possible in a bandit setting. Nonetheless, under the assumption that an upper-bound on the distance between tasks is available, we develop an algorithm which selectively transfers samples so as to reduce the overall transfer bias. Finally, we include empirical results of our multi-task algorithm showing the potential improvement in the regret, compared to learning tasks separately.

2 Transfer of Samples in Linear Bandits

We begin this section by briefly restating the linear bandit setting, with the particularity that we will consider here that a linear bandit problem is a task in a sequence of tasks to be used in a transfer scenario. We then introduce our formulation of the transfer version, where given a sequence of linear bandit problems, we investigate whether the transfer of samples from previous tasks can reduce the regret in the current task.

2.1 The Stochastic Linear Bandit Problem

We consider a decision set $\mathcal{X} = \{x_1, x_2, \dots, x_K\} \subset \mathbb{R}^d$ with a finite set of arms such that $||x||_2 \leq L$ for any $x \in \mathcal{X}$. A learner has to solve a sequence of tasks, where each task is a linear bandit problem. For every task $j = 1, 2, \dots$, the learner is given a limited sampling budget n_j and sequentially decides what arm $x \in \mathcal{X}$ to pull. At each step $t \leq n_j$, after an arm $x_{j,t} \in \mathcal{X}$ is pulled, the learner observes a reward $r_{j,t}$ obtained as a noisy realization of a linear combination of the selected arm and an unknown parameter $\theta_j \in \mathbb{R}^d$, i.e., $r_{j,t} = x_{j,t}^\top \theta_j + \eta_{j,t}$,where η is a zero-mean i.i.d. noise bounded in $[-\sigma; \sigma]$.

In each task j, the objective of the learner is to maximize the sum of expected rewards. Since the parameter θ_j is unknown, the learner faces the so-called *exploration-exploitation dilemma*, where the exploration of different arms may improve the estimate of θ_j, while the exploitation of the estimated best-arm would supposedly maximize the sum of rewards. Here we use the most typical regret definition and therefore the performance of the learner will be evaluated with respect to the sum of rewards obtained by pulling the optimal arm at each step. Denoting $x^*(\theta_j) = \arg\max_{x \in \mathcal{X}} x^\top \theta_j$ as the best arm for task j, the *regret* suffered by the learner at the end of task j is

$$\mathcal{R}(\theta_j) = \sum_{t=1}^{n_j} (x^*(\theta_j) - x_{j,t})^\top \theta_j.$$

Notice that this problem differs from the *contextual* bandit setting since no side information about task θ_j is provided. At each step t, the parameter θ_j is estimated by *regularized least-squares* as $\hat{\theta}_{j,t} = (A_{j,t} + \lambda I)^{-1} b_{j,t}$, where we denote by $A_{j,t} = \sum_{s=1}^{t} x_{j,s} x_{j,s}^\top$ is the design matrix, $b_{j,t} = \sum_{s=1}^{t} x_{j,s} r_{j,s}$, and λ is a regularization parameter. The accuracy of $\hat{\theta}_{j,t}$ is evaluated as previously, by using the concentration inequality for adaptive sequences in Prop.2.3.

2.2 Multi-task Linear Bandits

We now introduce the transfer of samples framework in a sequence of linear bandit problems. Before defining an actual transfer algorithm, we first study how the estimation of the parameter vector of a task may benefit from the transfer of samples coming from previous tasks. Let us consider a sequence of past tasks $(\theta_1, \dots, \theta_{m-1})$ and the current task whose parameter is θ_m. After t steps during task m, we construct the regularized *multi-task estimate* of θ_m by transferring all samples from past tasks:

$$\widetilde{\theta}_{m,t} = \Big(\sum_{j=1}^{m-1} A_j + A_{m,t} + \lambda I \Big)^{-1} \cdot \Big(\sum_{j=1}^{m-1} b_j + b_{m,t} \Big) \overset{\text{def.}}{=} (\widetilde{A}_{m,t} + \lambda I)^{-1} \cdot \widetilde{b}_{m,t}, \tag{5.1}$$

where $A_j = A_{j,n_j}$, $b_j = b_{j,n_j}$ are past design matrices and cumulative samples respectively, while $\widetilde{A}_{m,t}$ and $\widetilde{b}_{m,t}$ are the multi-task design matrix and the multi-task cumulative samples. In the following, we also use $\widetilde{A}^{\lambda}_{m,t} = \widetilde{A}_{m,t} + \lambda I$. Notice that $\widetilde{\theta}_{m,t}$ is obtained using a total of $N_{m,t} = \sum_{j=1}^{m-1} n_j + t$ samples, instead of the t samples of the current task. On the one hand, we expect that the larger number of samples could dramatically reduce the variance of $\widetilde{\theta}_{m,t}$ with respect to the single-task estimate $\hat{\theta}_{m,t}$. On the other hand, samples generated from different distributions may bias the estimate away from θ_m. In fact, it can be shown that $\widetilde{\theta}_{m,t}$ tends to concentrate towards the *multi-task vector*

$$\overline{\theta}_{m,t} = \widetilde{A}^{-1}_{m,t} \Big(\sum_{j=1}^{m-1} A_j \theta_j + A_{m,t} \theta_m \Big). \tag{5.2}$$

This observation allows to derive[3] a high-probability bound on the prediction error of the multi-task estimate obtained by sequential transfer of samples through tasks.

Lemma 5.1. *Let $\widetilde{\theta}_{m,t}$ be the multi-task regularized least-squares estimate defined in Eq. 5.1. Then, for all $x \in \mathbb{R}^d$, for all $\delta \geq 0$ and $t \geq 1$, with probability greater than $1 - \delta$:*

$$\left| x^\top (\widetilde{\theta}_{m,t} - \theta_m) \right| \leq \widetilde{\Delta}_{m,t}(x) + \Gamma_{m,t}(x), \tag{5.3}$$

where the estimation error $\widetilde{\Delta}_{m,t}(x)$ and the bias $\Gamma_{m,t}(x)$ are defined as

$$\widetilde{\Delta}_{m,t}(x) = ||x||_{(\widetilde{A}^{\lambda}_{m,t})^{-1}} \left(\sigma \sqrt{d \log \left(\frac{1 + N_{m,t} L^2 / \lambda}{\delta} \right)} + \lambda^{1/2} \|\overline{\theta}_{m,t}\| \right) \tag{5.4}$$

$$\Gamma_{m,t}(x) = x^\top (\overline{\theta}_{m,t} - \theta_m) = x^\top \Big(\widetilde{A}^{-1}_{m,t} \sum_{j=1}^{m-1} A_j (\theta_j - \theta_m) \Big). \tag{5.5}$$

Remark (estimation error). Compared to its single-task counterpart (see Prop. 2.3), the multi-task estimation error, $\widetilde{\Delta}_{m,t}(x)$, benefits from a reduced variance as soon as samples from at least one task are transferred, and this advantage increases

[3] All proofs are reported in the appendix of this chapter (Sect. 7).

with the number of tasks. In fact, the norm of x is now weighted by the inverse of the much "larger" multi-task design matrix $\widetilde{A}^{\lambda}_{m,t}$. This comes with a slight limited increase in the logarithmic term, where now the total number of samples $N_{m,t}$ appears. All other terms remain unchanged with respect to Prop. 2.3.

Remark (bias). The second term in the bound (Eq. 5.5) accounts for the bias of transferring samples from many different sources and is due to the fact that $\widetilde{\theta}_{m,t}$ tends to $\overline{\theta}_{m,t}$ (Eq. 5.2).

Let us first study more in detail how the bias reduces in the multi-armed bandit case, where all arms $x \in \mathcal{X}$ are orthogonal with norm 1. Let $T_{j,t}(x)$ be the number of times arm x is pulled in task j until the end of step t. Then, the bias for arm x at time step t of task m is

$$\Gamma_{m,t}(x) = \sum_{j=1}^{m-1} \frac{T_{j,n_j}(x)}{\sum_{j'=1}^{m-1} T_{j',n_{j'}}(x) + T_{m,t}(x)} x^\top(\theta_j - \theta_m),$$

which corresponds to a simple weighted average of the differences of value of arm x across different tasks, where the weights are determined by the proportions of pulls (i.e., the more an arm is pulled in a task θ_j, the higher the weight on the difference between θ_j and θ_m).

When moving to the linear bandit case, things are more intricate and the bias in the prediction of the value of x is not just a simple distance between the *average* value predicted in each different task and the value in the target task (i.e., $\sum_j w_j x^\top(\theta_j - \theta_m)$ for some specific weights w_j). Here, each component of $\overline{\theta}_{m,t}$ is a complicated mixture of all the components of the other tasks. As discussed in the next section, such mixture may even lead to amplify the norms of all the vectors θ_j from the past tasks. Moreover, we notice that the bias partially changes over time, as the multi-task design matrix $\widetilde{A}_{m,t}$ changes with t. Overall, the effectiveness of the multi-task estimate $\widetilde{\theta}_{m,t}$ will strongly depend on how much the reduction in variance is contrasted by the bias term.

3 The Oracle Multi-Task Linear UCB

In this section, we outline the general structure of a multi-task extension of LinUCB. We begin by proposing an oracle algorithm. Although not practical, this algorithm serves as a basis to study the general properties of the transfer of samples in linear bandits and in the regret analysis of multi-task strategies.

3.1 The MT-LinUCB Algorithm

Similarly to LinUCB [Chu et al., 2011, Li et al., 2010], MT-LinUCB relies on the construction of upper-confidence bounds on the value of the arms, which are then used to select the best *optimistic* arm. For every task j, while LinUCB only computes upper-bounds on the single-task estimate $\widehat{\theta}_{j,t}$ (Eq. 5.6), MT-LinUCB also computes the multi-task estimate $\widetilde{\theta}_{j,t}$ and stores upper-confidence bounds on both terms using

Algorithm 8 MT-LinUCB

Input: budgets $\{n_j\}_j$, arms $\mathcal{X} \subset \mathbb{R}^d$, regularizer λ
$\widetilde{A}_j = 0, \tilde{b}_j = 0$
for $j = 1, \dots, m$ **do**
  $A_{j,1} = 0, b_{j,1} = 0$
  for $t = 1, \dots, n_j$ **do**
    Compute $B_{j,t}(x) = x^\top \hat{\theta}_{j,t} + \Delta_{j,t}(x)$
    Compute $\widetilde{B}_{j,t}(x) = x^\top \widetilde{\theta}_{j,t} + \widetilde{\Delta}_{j,t}(x) + \Gamma_{j,t}(x)$
    Pull arm $x_t = \arg\max_{x \in \mathcal{X}} \min\left\{B_{j,t}(x); \widetilde{B}_{j,t}(x)\right\}$
    Observe reward r_t
    Update $A_{j,t+1} = A_{j,t+1} + x_t x_t^\top$, $b_{j,t+1} = b_{j,t} + x_t r_t$
    Update $\widetilde{A}_{j,t+1} = \widetilde{A}_j + A_{j,t+1}$, $\tilde{b}_{j,t+1} = \tilde{b}_j + b_{j,t+1}$
  end for
  $\widetilde{A}_{j+1} = \widetilde{A}_j + A_{j,n_j}, \tilde{b}_{j+1} = \tilde{b}_j + b_{j,n_j}$
end for

Prop. 2.3 and Lemma 5.1 respectively. More precisely, for the first task $j = 1$, MT-LinUCB coincides with LinUCB, since there are no samples available from past tasks, while for $j > 1$, MT-LinUCB constructs two independent upper-confidence bounds:

$$B_{j,t}(x) = x^\top \hat{\theta}_{j,t} + \Delta_{j,t}(x), \tag{5.6}$$
$$\widetilde{B}_{j,t}(x) = x^\top \widetilde{\theta}_{j,t} + \widetilde{\Delta}_{j,t}(x) + \Gamma_{j,t}(x) \tag{5.7}$$

Since both equations give valid upper-confidence bounds on the value of x, we only retain the tightest (smallest) of them, that is, the one closest to the true value of the arm and at each time step, the selected arm is

$$x_t = \arg\max_{x \in \mathcal{X}} \min\left\{B_{j,t}(x); \widetilde{B}_{j,t}(x)\right\}.$$

The use of the minimum between the two bounds is crucial for the correct functioning of the resulting algorithm (Fig. 8).

If we use only the multi-task bound $\widetilde{B}_{j,t}$, the algorithm may suffer from *negative transfer*. In fact, $\widetilde{B}_{j,t}$ contains a bias term $\Gamma_{j,t}$ which, unlike the estimation errors $\Delta_{j,t}$ and $\widetilde{\Delta}_{j,t}$, does not shrink as the number of samples t from the current task increases. As a result, an algorithm which only relies on $\widetilde{B}_{j,t}(x)$ may get stuck in pulling the same sub-optimal arm over and over, even when the estimation of θ_j is accurate. While this is a common issue in large segments of the transfer literature, where transfer may have a negative impact on the learning performance whenever tasks differ too much, in MT-LinUCB we successfully avoid this by integrating both single-task and multi-task bounds.

On the other hand, an algorithm failing to fully take into account the bias term would result in a *lack of asymptotic correctness* (and thus linear regret). In fact, an instance of LinUCB which continuously transfers samples without resetting at each task, would end up using the term $\widetilde{B}_{j,t}(x)$ without the $\Gamma_{j,t}(x)$. Not only such bound

would not be a valid upper-confidence bound anymore, such an algorithm would rapidly get stuck on pulling only the "best on average" arm. Therefore, the update of the two bounds is essential to make MT-LinUCB robust to negative transfer and to guarantee the asymptotical correctness.

3.2 Regret analysis

We now investigate the possible advantages and drawbacks of MT-LinUCB by comparing it to LinUCB. The following theorem provides a high-probability bound on the regret of MT-LinUCB at task m, after observing t task-specific rewards, and transferring available samples from previous tasks.

Theorem 5.1. *If* MT-LinUCB *is run on task θ_m using samples from tasks $\theta_1, \dots, \theta_{m-1}$, then with probability greater than $1-\delta$ its regret is upper-bounded as follows:*

$$\mathcal{R}(\theta_m) \leq \underbrace{2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)}+\lambda^{1/2}\overline{\theta}_{m,\max}\right)}_{\boldsymbol{A}}\underbrace{\sqrt{n_m\sum_{t=1}^{n_m}\alpha_{m,t}||x_t||^2_{(A^{\lambda}_{m,t-1})^{-1}}}}_{\boldsymbol{B}}+\underbrace{2\sum_{t=1}^{n_m}\Gamma_{m,t}(x_t)}_{\boldsymbol{C}},$$

where $\overline{\theta}_{m,\max} = \max_t ||\overline{\theta}_{m,t}||$, $N_{m,n_m} = \sum_{j=1}^{m} n_j$, and we define

$$\alpha_{m,t} = \frac{1}{1+\lambda_{\min}\left((A^{\lambda}_{m,t})^{-1}\sum_{j=1}^{m-1}A_j\right)}.$$

First notice that since MT-LinUCB uses the minimum of $B_{m,t}$ and $\widetilde{B}_{m,t}$ at each step, then its regret is indeed upper-bounded by the minimum between the bound provided in the previous theorem and the regret of single-task LinUCB (Thm. 3 in [Abbasi-Yadkori et al., 2011]). As such, MT-LinUCB is guaranteed to avoid negative transfer and never perform worse than LinUCB.

While term **A** is basically the same as in the bound of LinUCB, term **B**, which can be simplified to $\widetilde{O}(\sqrt{\alpha_m n_m})$ (up to logarithmic factors), summarizes the potential of MT-LinUCB in reducing the regret. In fact, since $\alpha_m < 1$ by definition, the regret is always smaller than the single-task counterpart $O(\sqrt{n_m})$.[4] In Corollary 5.1 we discuss the role of the coefficients α_m and provide a clearer regret bound, showing that in the best case we can obtain a regret of $\widetilde{O}(\sqrt{n_m/m})$, with an improvement of $1/\sqrt{m}$ w.r.t the regret of LinUCB.

Corollary 5.1. *Let $\lambda_0 = \min_{j=1,\dots,m-1}\lambda_{\min}(A_j)$, then the coefficients $\alpha_{m,t}$ can be upper-bounded as $\alpha_{m,t} \leq \alpha_m := \frac{n_m L+\lambda}{n_m L+\lambda+(m-1)\lambda_0}$. Moreover, we can define the effective multi-task fraction as*

$$\beta_{m,t} = \max_{\beta}\left\{\beta > 0 : \exists H_{m,t} \succ 0, \sum_{j=1}^{m-1}A_j = \beta A^{\lambda}_{m,t} + H_{m,t}\right\}.$$

[4] We recall that the terms in the inner summation of **B** decrease with t and even if $\alpha_{m,t} = 1$ the overall summation is at most of order $O(\log n_m)$ as proved in Lemma 11 in [Abbasi-Yadkori et al., 2011].

Then if $\beta_m = \min_{t=1,\dots,n_m} \beta_{m,t}$ *we obtain* $\alpha_m = 1/(1+\beta_m)$ *and then the regret can be written as*

$$\mathcal{R}(\theta_m) \leq 2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)}+\lambda^{1/2}\overline{\theta}_{m,\max}\right)\sqrt{\alpha_m n_m d\log(1+N_{m,n_m}L^2/\lambda)}+2\sum_{t=1}^{n_m}\Gamma_{m,t}(x_t).$$

It is interesting to inspect more closely the definition of α_m as a function of the effective multi-task fraction β_m. The fraction $\beta_{m,t}$ measures the amount of *useful* information brought from the past tasks with respect to the arms currently sampled in task m. In particular, the usefulness of the past design matrices depends on how much they are *aligned* with the current design $A^{\lambda}_{m,t}$. It is easy to see that in the best case $\beta_m = O(m)$, which corresponds to a regret $\widetilde{O}(\sqrt{n_m/m})$ with a $1/\sqrt{m}$ reduction with respect to to LinUCB. On the other hand, when only few past tasks are available and the bandit strategy in the previous tasks produced a multi-task design matrix very different from the one generate during the current task, then α_m could be very close to 1, thus reducing the multi-task advantage. Finally, we notice that the bound in Corollary 5.1 guarantees that α_m decreases with m, thus confirming the intuition that the performance of MT-LinUCB will tend to improve over tasks.

Another important term in the previous bound is the smallest eigenvalue λ_0 of the task design matrices. While it is not always trivial to lower-bound λ_0, in general it is easy to force the algorithm to perform an initial "smart" exploration to obtain a desired minimum eigenvalue (see e.g., the use of barycentric spanners in [Dani et al., 2008]). Moreover, results in linear regression (see e.g., [Lai and Wei, 1982]) show that the condition $o(\lambda_{\min}) = \log(\lambda_{\max})$ is required to guarantee asymptotic consistency, necessary to have sublinear anytime regret. Since $\lambda_{\max} = O(n)$, this condition implies that λ_0 should be at least $\log n$.

Term **C** captures the potential drawback of MT-LinUCB. The bias introduced by transferring samples from tasks which differ from the current one is accumulated over time and it introduces a linear factor in the regret. This is the unavoidable price to pay for transferring from different tasks, although it remains small when considering tasks that are relatively close to each other and when the sampling budget per task is limited (as is usually the case in scenarios using transfer methods).

In the next section, we propose a practical algorithm for the multi-task linear bandit problem, we upper-bound its worse-case regret, and we further discuss how to deal with the bias term.

4 Multi-task Strategies

In MT-LinUCB the multi-task bound $\widetilde{B}_{m,t}(x)$ is constructed using the ℓ_2-norm of the multi-task vector $||\overline{\theta}_{m,t}||$ in Eq. 5.4 and the bias $\Gamma_{m,t}(x)$ in Eq. 5.5. In general, these quantities cannot be computed exactly, since vectors $\theta_1, \dots, \theta_m$ are unknown. As a result, they must be replaced by quantities that can actually be computed by the algorithm. We consider the case where an upper-bound on the *distance* between any two tasks is known, and we derive a bound on the terms related to $\overline{\theta}_{m,t}$.

Lemma 5.2. *Let $\epsilon > 0$, such that $\max_{i,j=1,\dots,m} ||\theta_i - \theta_j|| \leq \epsilon$. We introduce the* multi-task disalignment *as $\Upsilon_{m,t} = \sum_{j=1}^{m-1} \|\widetilde{A}_{m,t}^{-1} A_j\| + \|\widetilde{A}_{m,t}^{-1} A_{m,t}\|$. Then we have*

$$\|\overline{\theta}_{m,t}\| \leq S + \epsilon \Upsilon_{m,t} \quad \textit{and} \quad \Gamma_{m,t}(x) \leq L\epsilon\Upsilon_{m,t}. \tag{5.8}$$

4.1 MT-UB Strategy

Upper-bound on the bias (MT-UB). The term $\Upsilon_{m,t}$ can be computed exactly since all its elements are known at run time. As a result, when MT-LinUCB is provided with ϵ, it can replace $\|\overline{\theta}_{m,t}\|$ and $\Gamma_{m,t}(x)$ needed to compute $\widetilde{B}_{m,t}$ with the upper-bounds in Eq. 5.8. We will refer to such an algorithm as MT-UB. Notice that knowing ϵ does not differ much from the standard assumption in LinUCB of knowing an actual upper-bound S on the norm of the parameter vector $||\theta_j|| \leq S$ (see definition of $\Delta_{m,t}$). Here the upper-bound is not on the norm of each θ_j but on their distance. As in the case for S, where loose upper-bounds may negatively affect the performance, ϵ should be tight enough to provide an actual improvement over single-task learning. In the following corollary, we provide an upper-bound on the worst-case regret that MT-UB can suffer.

Corollary 5.2. *Let $\lambda_0 = \min_{j=1,\dots,m-1} \lambda_{\min}(A_j)$ and $n_{\max} = \max_{j=1,\dots,m} n_j$, then $\Upsilon_{m,t} \leq \Upsilon_m = \frac{n_{\max}}{\lambda_0}$. With probability at least $1-\delta$, the regret of* MT-UB *satisfies*

$$\mathcal{R}(\theta_m) \leq 2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)} + \lambda^{\frac{1}{2}}(S+\epsilon\Upsilon_m)\right)\sqrt{\alpha_m n_m d\log(1+N_{m,n_m}L^2/\lambda)} + 2n_m L\epsilon\Upsilon_m.$$

While the impact of Υ_m in the first part of the regret is negligible, since it can be easily contrasted by a suitable choice of the regularizer λ at the cost of a mild increase in the logarithmic term, it may be more relevant in the linear bias term. Nonetheless, we notice that in some cases Υ_m may be very close to 1 and thus we expect that whenever the tasks are indeed very similar (that is, ϵ is small) the advantage with respect to single-task learning discussed in the previous section will still be preserved.

Remark (multi-task disalignment). It is interesting to understand in more detail the multi-task disalignment and its potential relationship with the similarity between tasks. From the study of Lemma 5.1 in the multi-armed bandit case (i.e., all arms are orthogonal), we can see that the bounds in Lemma 5.2 are loose. More generally, whenever all the design matrices can be decomposed in the same eigenspace, we can obtain much tighter bounds. Let us assume that there exists an orthonormal matrix $U \in \mathbb{R}^d$ such that any matrix $A_j = U\Sigma U^\top$,[5] then

$$\|\widetilde{A}_{m,t}^{-1} A_j\| = U\Big((\sum_{j'=1}^{m-1} \Sigma_{j'} + \Sigma_{m,t})^{-1}\Sigma_j\Big)U^\top,$$

and $\Upsilon_{m,t} = 1$. As a result, in this case, with a more careful upper-bound on $||\overline{\theta}_{m,t}||$, we obtain $||\overline{\theta}_{m,t}|| \leq S$ and $\Gamma_{m,t}(x) \leq L\epsilon$. On the other hand, in general $\Upsilon_{m,t}$ can be

[5] All matrices A_j are positive definite and a singular value decomposition $A_j = U_j\Sigma_j U_j^\top$ always exists but not necessarily with the same eigenvectors.

larger than 1 but we can expect it to be small depending on the similarity between tasks. In fact, the more similar the design in each task (i.e., $A_j \approx A_{j'}$), the smaller $\Upsilon_{m,t}$, with the extreme case when the same allocation strategy is used across all the tasks, for which $\Upsilon_{m,t} = 1$. Since in each task bandit algorithms tend to explore more often nearly-optimal arms, we expect design matrices to be more aligned as ϵ is small (i.e., tasks are close to each other). The negative effect of the disalignment is intuitive, since the *usefulness* of the samples transferred from past tasks is strictly related to how similar the sampling strategies were. This problem is also common to related problems such as *domain adaptation* where training and testing distributions are not the same (see e.g., [Mansour et al., 2009]).

4.2 MT-TS Strategy

Lemma 5.2 shows that it is critical to keep the disalignment $\Upsilon_{m,t}$ as small as possible, and it suggests that selecting only *well-aligned* past tasks could improve the performance. Nonetheless, this would come at the cost of decreasing the number of transferred samples and the corresponding advantage in terms of variance reduction. As a result, it is important to find a suitable trade-off between the disalignment (and its negative effect on the bias) and the number of transferred samples.

Here we propose a solution which performs a Task Selection at each time step and includes only the most relevant prior tasks in the transfer scenario. More precisely, at time step t of task m, we include in the multi-task design matrix $\widetilde{A}_{m,t}$ only the prior design matrices A_j whose mixture would lead to constructing the tightest multi-task upper-bound on the arms values. Notice that a similar argument is used in [Kuzborskij and Orabona, 2014] in the context of hypothesis transfer. Let $\mathcal{S} \subseteq \{1, \dots, m-1\}$ be a subset of past tasks, and $\Upsilon_{m,t}(\mathcal{S})$ and $\widetilde{\Delta}_{m,t}(x, \mathcal{S})$ be the disalignment and estimation errors when only the tasks in $\mathcal{S}$ are used. The goal is to find the set of tasks minimizing the (non-convex) function[6]:

$$f_{m,t}(\mathcal{S}) = L\epsilon\Upsilon_{m,t}(\mathcal{S}) + \max_{x \in \mathcal{X}} \widetilde{\Delta}_{m,t}(x, \mathcal{S}).$$

Here, we use a greedy algorithm to approximate the optimal solution achieved by $\mathcal{S}^*_{m,t} = \arg\min_{\mathcal{S}} f_{m,t}(\mathcal{S})$. Starting from $\mathcal{S}^0 = \emptyset$, we evaluate all the past tasks and we add the one that achieves the smallest value of $f_{m,t}(\mathcal{S})$. Then this task is added to $\mathcal{S}^0$, thus obtaining $\mathcal{S}^1$ and the process is repeated. At each iteration, we try to add tasks not in $\mathcal{S}$ until the objective function is no longer improved and the final set $\mathcal{S}_{m,t}$ is returned. We refer to this algorithm as MT-TS.

Note that the MT-TS strategy is a refinement of the MT-UB strategy, therefore the regret of the former would be the same as given in Cor. 5.2 but with more optimized terms α_m and υ_m, whose exact expression would strongly depend on the specific sequence of tasks up to m.

[6] Notice that $f(\mathcal{S})$ could also be defined for each x separately.

5 Experiments

In this section we report numerical simulations of the multi-task algorithms introduced above, with the objective of evaluating the accuracy of the theoretical bounds, in particular in terms of the regret reduction with respect to single-task learning. Besides the comparison with the baseline LinUCB, we report the performance of the Oracle multi-task algorithm (MT-LinUCB), which has access to the parameters θ of all tasks and can compute the exact bias $\Gamma_{m,t}(x)$ defined in Eq. 5.5. While such algorithm cannot be actually implemented in practice, it serves as a lower-bound on the smallest regret achievable with the transfer scheme. Then, we measure the regret obtained by the multi-task algorithm that receives as parameter an upper-bound on the bias (MT-UB). A more sophisticated transfer of samples is done by task-selection algorithm (MT-TS) which sequentially chooses the prior tasks to be included in the transfer of samples scenario.

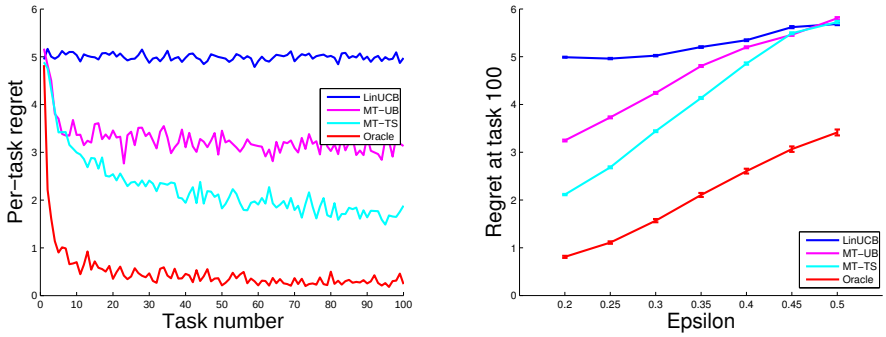


Figure 5.1: Per-task regret for multi-task algorithms *(left)* and sensitivity to parameter ϵ *(right)*.

We illustrate the performance of the algorithms in a $d = 4$ setting consisting of 100 tasks, with parameters θ generated uniformly at random, but with bounded range for the features such that $\max_\theta ||\theta||_2 = 2$ and $\epsilon = 0.2$. The decision set $\mathcal{X}$ consists of six arms with $||x||_2 < L = 1$, the regularization parameter λ is set to 0.02, the noise η is distributed uniformly on $[-0.25, 0.25]$ and $\delta = 0.05$. In Fig. 5.1-*left* we report the per-task regret obtained using a per-task sampling budget of 50 pulls equal for all tasks [7] and results are averaged over 100 independent runs.

Single- versus Multi-task Strategies. We first notice that the regret of LinUCB remains constant (on average) over tasks, since independently of how many tasks were already performed, it only uses the task-specific samples to decide on the arm sampling strategy. In the case of the multi-task algorithms, the regret decreases over tasks, whenever the reduction in variance obtained through transfer is more important than the introduced bias. In particular, MT-UB outperforms LinUCB, thus showing that the potential negative effects of the disalignment are not of primary concern in this

[7] The choice of a short horizon is motivated by the fact that transfer methods are usually designed to compensate for the lack of per-task samples available per task.

case and they do not prevent it from improving the performance across tasks.

MT-UB versus MT-TS. Nonetheless, we notice that MT-UB always takes into account the upper-bound on the bias, regardless of the similarity of prior tasks to the task at hand. This prevents its regret to decreasing down to levels similar to the ORACLE. On the other hand, for MT-TS, the decision of whether to add a specific prior task in the transfer scenario depends on the variance-bias trade-off that is generated. In this experiment, $\mathcal{S}_{m,t}$ is recomputed every 5 steps and a careful selection of tasks allows MT-TS to significantly improve the performance of MT-UB by reducing the impact of the multi-task disalignment and preserving the advantage of transfer at the same time. The regret of MT-TS is thus much closer to ORACLE, which exploits the exact knowledge of the bias.

Finally, we also study the sensitivity of our algorithms to ϵ. In Fig. 5.1-*right*, we report the per-task regret after 100 tasks for values of ϵ in $[0.2; 0.5]$, keeping all other parameters fixed as in the previous experiment. First notice that the regret of LINUCB is not affected, since the bias term does not appear in the single-task bound. Then, we recall from Cor. 5.2 that MT-UB suffers from a regret that grows linearly with n_m and ϵ. As a result, it is not surprising that the performance of MT-UB rapidly worsens as ϵ increases. While a similar situation holds also for MT-TS, we can see that task-selection makes the algorithm more robust even for larger ϵ. On the other hand, when ϵ is too big the performance of multi-task algorithms (including ORACLE) tends to approach LINUCB. However, we recall that they would never perform worse than LINUCB due to the use of the minimum between the bounds $B_{m,t}$ and $\widetilde{B}_{m,t}$ in the arm selection.

6 Discussion

Estimating the bias. Even if a relatively tight bound ϵ is available, MT-LINUCB may be sub-optimal, since ϵ represents an upper-bound on the distance across *all* tasks. For instance, at task $j < m$, the actual bias $\Gamma_{j,t}$ compares only tasks up to θ_j and it may be much smaller than ϵ. Consider for instance the case where tasks $\theta_1, \dots, \theta_{m-1}$ are very close to each other, while θ_m is very different, then ϵ should be designed so as to take into consideration the *largest* possible deviation between two tasks, thus becoming very loose for all tasks but the last. Furthermore, the bias $\Gamma_{m,t}$ measures the distance between the multi-task vector $\overline{\theta}_{m,t}$ and θ_m (see Eq. 5.2) rather than the *sum of the distance* as in the derivation of Eq. 5.8, thus potentially resulting in a loose bound. As a result, it is possible that $\overline{\theta}_{m,t}$ will be closer to θ_m than the actual average distance, thus making the use of ϵ very sub-optimal. This suggests that MT-LINUCB may greatly benefit from replacing the bound $L\epsilon\Upsilon_{m,t}$ with a direct estimate of the bias, such as

$$\widehat{\Gamma}_{m,t}(x) = x^\top \left(\widetilde{A}_{m,t}^{-1} \sum_{j=1}^{m-1} A_j (\hat{\theta}_j - \hat{\theta}_{m,t}) \right), \tag{5.9}$$

where $\hat{\theta}_j$ and $\hat{\theta}_{m,t}$ are the empirical estimates of the parameter vectors. In order to integrate $\widehat{\Gamma}_{m,t}(x)$ into $\widetilde{B}_{m,t}(x)$ we need to derive a bound on the accuracy of $\widehat{\Gamma}_{m,t}(x)$ and use it to guarantee that $\widetilde{B}_{m,t}(x)$ is still a valid upper-bound on the value of the arm. A trivial solution is to apply Prop. 2.3, thus obtaining

$$|\widehat{\Gamma}_{m,t}(x) - \Gamma_{m,t}(x)| \leq \widetilde{\Delta}_{m,t}(x) + \Delta_{m,t}(x),$$

which leads to the multi-task bound

$$\widetilde{B}_{m,t}(x) = x^\top \widetilde{\theta}_{m,t} + 2\widetilde{\Delta}_{m,t}(x) + \widehat{\Gamma}_{m,t}(x) + \Delta_{m,t}(x).$$

Unfortunately, in this case any potential advantage in reducing the variance in $\widetilde{\Delta}_{m,t}(x)$ would be removed by the presence of the (high-variance) term $\Delta_{m,t}(x)$ which already appears in the single-task bound $B_{m,t}$. More sample efficient solutions for the estimation of the transfer bias have been proposed by [Crammer et al., 2008] for binary classification and by [Lazaric and Restelli, 2011] in MDPs. Nonetheless, the proposed methods require that *exactly* the same sampling policy is used across all the tasks and the derived bounds are only provided *on average* w.r.t. to such sampling policy. As a result, these methods cannot be used here, where the sampling policy is an adaptive bandit algorithm which selects different arms in each task and the bias in each possible $x \in \mathcal{X}$ needs to be estimated. This shows that estimating the bias may indeed be very difficult in bandit problems and we leave this problem for future work.

Contributions and future work. In this chapter, we formalized the transfer learning problem in the linear bandit setting. Unlike previous work on transfer in sequential decision-making settings, we did not assume a finite set of problems is available. Rather, we considered the case where all tasks can be different, provided their parameter vectors are close in a ℓ_2-norm sense. We studied a simple transfer mechanism where samples are incrementally transferred from past tasks with the objective of reducing the estimation error in the current task. The theoretical analysis shows that our proposed oracle and practical multi-task strategies may already be effective in reducing the per-task regret. On the other hand, the analysis also revealed the critical role played by the similarity in the arm-selection strategies of different tasks. Finally, we studied the empirical performance of multi-task algorithms in a simple synthetic problem. Directions for future investigation include the relaxation of the notion of similarity, the study of the problem of bias estimation, and an investigation of more sophisticated transfer strategies (such as the transfer of solutions mechanism).

7 Appendix

7.1 Proofs of Section 2

Proof of Lemma 5.1. We bound the prediction error in estimating the expected value of any arm x when using the multi-task estimate $\widetilde{\theta}_{m,t}$ built using samples from the $m-1$ past tasks and t samples from the current task m. In particular, we decompose the error of the multi-task least-squares estimate defined in Eq. 5.1 into the estimation error coming from the randomness of the samples **(1)** and the bias introduced by the difference between tasks **(2)**:

$$\left|x^\top \widetilde{\theta}_{m,t} - x^\top \theta_m\right| \leq \underbrace{\left|x^\top \widetilde{\theta}_{m,t} - x^\top \overline{\theta}_{m,t}\right|}_{\textbf{(1)}} + \underbrace{\left|x^\top \overline{\theta}_{m,t} - x^\top \theta_m\right|}_{\textbf{(2)}}. \tag{5.10}$$

(1) Multi-task estimation error

Using the definition of the multi-task estimator $\widetilde{\theta}_{m,t}$ (Eq. 5.1) and denoting by $A_{m,t}, b_{m,t}$ the design matrix and the sample vector containing only samples observed from task m, we have that for all $0 < t \leq n_m$:

$$\begin{aligned}
\widetilde{\theta}_{m,t} &= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\widetilde{b}_{m,t} \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1} b_j + b_{m,t}\Big) \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s} r_{j,s} + \sum_{s=1}^{t} x_{m,s} r_{m,s}\Big) \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\left(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}(x_{j,s}^\top \theta_j + \eta_{j,s}) + \sum_{s=1}^{t} x_{m,s}(x_{m,s}^\top \theta_m + \eta_{m,s})\right) \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\left(\sum_{j=1}^{m-1} A_j\theta_j + \sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + A_{m,t}\theta_m + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}\right) \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}\Big) + \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1} A_j\theta_j + A_{m,t}\theta_m\Big) \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}\Big) \\
&\quad + \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\widetilde{A}_{m,t}\underbrace{\left(\widetilde{A}_{m,t}^{-1}\Big(\sum_{j=1}^{m-1} A_j\theta_j + A_{m,t}\theta_m\Big)\right)}_{\overline{\theta}_{m,t}} \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}\Big) \\
&\quad + \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\widetilde{A}_{m,t} + \lambda I\Big)\overline{\theta}_{m,t} - \lambda\Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\overline{\theta}_{m,t} \\
&= \Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\Big(\sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}\Big) + \overline{\theta}_{m,t} - \lambda\Big(\widetilde{A}_{m,t} + \lambda I\Big)^{-1}\overline{\theta}_{m,t}\ .
\end{aligned}$$

For notational convenience we introduce $\widetilde{A}^{\lambda}_{m,t} = \widetilde{A}_{m,t} + \lambda I$, $h_{m,t} = \sum_{j=1}^{m-1}\sum_{s=1}^{n_j} x_{j,s}\eta_{j,s} + \sum_{s=1}^{t} x_{m,s}\eta_{m,s}$ and $N_{m,t} = \sum_{j=1}^{m} n_j + t$. Then, we obtain that for every $x \in \mathcal{X}$, with probability greater than $1-\delta$, we have

$$\begin{aligned}
\left|x^\top \widetilde{\theta}_{m,t} - x^\top \overline{\theta}_{m,t}\right| &= \left|x^\top \left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1} h_{m,t} - \lambda x^\top \left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1} \overline{\theta}_{m,t}\right| \\
&\overset{(a)}{\leq} \left\langle x, h_{m,t}\right\rangle_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} + \left\langle \lambda x, \overline{\theta}_{m,t}\right\rangle_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} \\
&\overset{(b)}{\leq} \left\|x\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} \left(\left\|h_{m,t}\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} + \lambda \left\|\overline{\theta}_{m,t}\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}}\right) \\
&\overset{(c)}{\leq} \left\|x\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} \left(\left\|h_{m,t}\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} + \lambda^{1/2} \left\|\overline{\theta}_{m,t}\right\|_{2}\right) \\
&\overset{(d)}{\leq} \left\|x\right\|_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} \left(\sigma \sqrt{d \log\left(\frac{1 + N_{m,t}L^2/\lambda}{\delta}\right)} + \lambda^{1/2} \left\|\overline{\theta}_{m,t}\right\|_{2}\right),
\end{aligned}$$

where **(a)** follows from the triangle inequality and the definition of matrix-weighted inner product, **(b)** follows from the Cauchy-Schwarz inequality, **(c)** follows from the fact that $\left\|\overline{\theta}_{m,t}\right\|^2_{\left(\widetilde{A}^{\lambda}_{m,t}\right)^{-1}} \leq 1/\lambda_{\min}(\widetilde{A}^{\lambda}_{m,t})\left\|\overline{\theta}_{m,t}\right\|^2_2 \leq 1/\lambda \left\|\overline{\theta}_{m,t}\right\|^2_2$, **(d)** is an application of Thm. 2 in [Abbasi-Yadkori et al., 2011].

Thus, under the assumption that the noise is the same across all tasks, we recover the same type of bound as in the single-task setting (Prop. 2.3). The first difference is that the single-task matrix regularized $A^{\lambda}_{m,t}$ is replaced here with the multi-task matrix $\widetilde{A}^{\lambda}_{m,t}$. Finally, the norm of the target mixture of tasks $\overline{\theta}_{m,t}$ appears. Although $\overline{\theta}_{m,t}$ is a combination of the parameter vectors of the original tasks, each of them is transformed by the matrix $\widetilde{A}^{-1}_{m,t}A_j$ which may significantly affect the norm of the resulting vector, as studied in the following.

(2) Multi-task bias

From the definition of $\overline{\theta}_{m,t}$ we can easily rewrite the bias term as

$$x^\top(\overline{\theta}_{m,t} - \theta_m) = x^\top\left(\widetilde{A}^{-1}_{m,t}\sum_{j=1}^{m-1} A_j(\theta_j - \theta_m)\right).$$

□

7.2 Proofs of Section 3

Proof of Theorem 5.1. We focus on the regret of task m when transferring samples from tasks $j = 1, \dots, m-1$ and we study the regret $\rho_{m,t}$ at each time step t

$$\begin{aligned}
\rho_{m,t} &= (x^*(\theta_m) - x_t)^\top \theta_m \\
&\overset{(a)}{\leq} 2\left(\widetilde{\Delta}_{m,t-1}(x_t) + \Gamma_{m,t}(x_t)\right)
\end{aligned}$$

$$\overset{(b)}{\leq} 2\left|\left|x_t\right|\right|_{(\widetilde{A}^{\lambda}_{m,t-1})^{-1}}\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,t}L^2/\lambda}{\delta}\right)}+\lambda^{1/2}\|\overline{\theta}_{m,t}\|\right)+2\Gamma_{m,t}(x_t)$$
$$=\rho'_{m,t}+2\Gamma_{m,t}(x_t)$$

where we recall that $x^*(\theta_m)=\arg\max_{x\in\mathcal{X}} x^\top\theta_m$ is the best arm at task m, x_t is the arm chosen at time t of task m, **(a)** follows from Lemma 5.1 and the same steps as in Eq. 7 of [Abbasi-Yadkori et al., 2011], and **(b)** comes from the definition of the estimation error in Eq. 5.4. We now move to the full regret of task m and obtain

$$\begin{aligned}R_{n_m}(\theta_m)&=\sum_{t=1}^{n_m}\rho_{m,t}\\&\overset{(a)}{\leq}\sqrt{n_m\sum_{t=1}^{n_m}\rho'^2_{m,t}}+2\sum_{t=1}^{n_m}\Gamma_{m,t}(x_t)\\&\overset{(b)}{\leq}2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)}+\lambda^{1/2}\overline{\theta}_{m,\max}\right)\sqrt{n_m\sum_{t=1}^{n_m}\left|\left|x_t\right|\right|^2_{(\widetilde{A}^{\lambda}_{m,t-1})^{-1}}}\\&+2\sum_{t=1}^{n_m}\Gamma_{m,t}(x_t),\end{aligned}$$

where in **(a)** we use Cauchy-Schwarz inequality and in **(b)** we use the definition of $\rho'_{m,t}$ and $\overline{\theta}_{m,\max}=\max_{t=1,\dots,n_m}||\overline{\theta}_{m,t}||$. Thus we are left with the study of the sequence $\left|\left|x_t\right|\right|^2_{(\widetilde{A}^{\lambda}_{m,t-1})^{-1}}$. Let $B=\sum_{j=1}^{m-1}A_j$, then we have the following upper bound

$$\begin{aligned}\left|\left|x_t\right|\right|^2_{(\widetilde{A}^{\lambda}_{m,t})^{-1}}&=x_t^\top(\widetilde{A}^{\lambda}_{m,t-1})^{-1}x_t\\&=x_t^\top\left(A^{\lambda}_{m,t-1}+B\right)^{-1}x_t\\&=x_t^\top\left[(A^{\lambda}_{m,t-1})^{1/2}(I+(A^{\lambda}_{m,t-1})^{-1/2}B(A^{\lambda}_{m,t-1})^{-1/2})(A^{\lambda}_{m,t-1})^{1/2}\right]^{-1}x_t\\&=x_t^\top(A^{\lambda}_{m,t-1})^{-1/2}\left(I+(A^{\lambda}_{m,t-1})^{-1/2}B(A^{\lambda}_{m,t-1})^{-1/2}\right)^{-1}(A^{\lambda}_{m,t-1})^{-1/2}x_t\\&\leq\frac{1}{\lambda_{\min}(I+(A^{\lambda}_{m,t-1})^{-1/2}B(A^{\lambda}_{m,t-1})^{-1/2})}||x_t||^2_{A^{-1}_{m,t-1}}\\&\overset{(a)}{\leq}\frac{1}{1+\lambda_{\min}((A^{\lambda}_{m,t-1})^{-1}B)}||x_t||^2_{(A^{\lambda}_{m,t-1})^{-1}},\end{aligned}$$

where **(a)** follows from the fact that the spectrum of $(A^{\lambda}_{m,t-1})^{-1/2}B(A^{\lambda}_{m,t-1})^{-1/2}$ and the spectrum of $(A^{\lambda}_{m,t-1})^{-1}B$ coincide. By defining $\alpha_{m,t}<1$ the first factor in the previous expression we obtain the final statement as

$$R_{n_m}(\theta_m)\leq 2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)}+\lambda^{1/2}\overline{\theta}_{m,\max}\right)\sqrt{n_m\sum_{t=1}^{n_m}\alpha_{m,t}\left|\left|x_t\right|\right|^2_{(A^{\lambda}_{m,t-1})^{-1}}}$$

$$+ 2\sum_{t=1}^{n_m} \Gamma_{m,t}(x_t).$$

□

Proof of Corollary 5.1. In this corollary, we provide a more comprehensive study of the coefficients $\alpha_{m,t}$ in the regret bound.

A relatively loose lower-bound on the smallest eigenvalue of the matrix $(A^{\lambda}_{m,t})^{-1}B$ is

$$\begin{aligned}\lambda_{\min}((A^{\lambda}_{m,t-1})^{-1}B) &\overset{(a)}{\geq} \lambda_{\min}((A^{\lambda}_{m,t-1})^{-1})\lambda_{\min}(B) = \lambda_{\max}(A^{\lambda}_{m,t-1})^{-1}\lambda_{\min}(B) \\ &\overset{(b)}{\geq} \frac{\sum_{j=1}^{m-1}\lambda_{\min}(A_j)}{(t-1)L+\lambda} \geq \frac{(m-1)\min_{j=1,\dots,m-1}\lambda_{\min}(A_j)}{(t-1)L+\lambda} \\ &\geq \frac{(m-1)\min_{j=1,\dots,m-1}\lambda_{\min}(A_j)}{n_m L+\lambda},\end{aligned}$$

where **(a)** follows from the fact that $\lambda_{\min}(M_1M_2) \geq \lambda_{\min}(M_1)\lambda_{\min}(M_2)$ for any pair of positive definite matrices (see e.g., [Merikoski et al., 2004]), **(b)** comes from the fact that $A^{\lambda}_{m,t}$ is the sum of $(t-1)$ rank 1 matrices (with largest eigenvalue L) and the regularization λI, and from $\lambda_{\min}(M_1+M_2) \geq \lambda_{\min}(M_1)+\lambda_{\min}(M_2)$. As a result we have

$$\alpha_{m,t} \leq \frac{tL+\lambda}{tL+\lambda+(m-1)\min_{j=1,\dots,m-1}\lambda_{\min}(A_j)},$$

and

$$\alpha_{m,t} \leq \alpha_m = \frac{n_m L+\lambda}{n_m L+\lambda+(m-1)\min_{j=1,\dots,m-1}\lambda_{\min}(A_j)}.$$

Alternatively we can define the *effective multi-task fraction*

$$\beta_{m,t} = \max_{\beta}\Big\{\beta > 0 : \exists H_{m,t} \succ 0, B = \beta A^{\lambda}_{m,t} + H_{m,t}\Big\},$$

which represents *how much* of the design matrix in the current task is represented in the multi-task design matrix from past tasks. Then we have the bound

$$\lambda_{\min}((A^{\lambda}_{m,t})^{-1}B) \geq \lambda_{\min}((A^{\lambda}_{m,t})^{-1}\beta_{m,t}A^{\lambda}_{m,t} + (A^{\lambda}_{m,t})^{-1}H_{m,t}) \overset{(a)}{\geq} \beta_{m,t},$$

where in **(a)** we have that the matrix $(A^{\lambda}_{m,t})^{-1}H_{m,t}$ in general is not positive definite but, since it is obtained as the product of two positive definite matrices, it has all real eigenvalues that we (loosely) lower-bound by 0. Let $\beta_m = \min_{t=,1\dots,n_m}\beta_{m,t}$ then we obtain[8]

$$\alpha_{m,t} \leq \alpha_m = \frac{1}{1+\beta_m}.$$

[8]Notice that we can reasonably expect $\beta_{m,t}$ to decrease as t increases, since the number of arms composing $A_{m,t}$ increases. Nonetheless, even after n_m pulls, we have that A_{m,n_m} has to be compared with the multi-task matrix B which contains $m-1$ matrices of a similar form as A_{m,n_m} and thus we can expect that β_m is still large.

In both cases, given $\alpha_m < 1$ we have $\left|\left|x_t\right|\right|^2_{(\widetilde{A}^{\lambda}_{m,t})^{-1}} \leq \alpha_m ||x_t||^2_{(A^{\lambda}_{m,t})^{-1}}$. As a result we can use the same bound as in [Abbasi-Yadkori et al., 2011] and obtain that

$$\sum_{t=1}^{n_m} \left|\left|x_t\right|\right|^2_{(\widetilde{A}^{\lambda}_{m,t})^{-1}} \leq \alpha_m d \log\Big(1 + N_{m,n_m} L^2/\lambda\Big),$$

which leads to the final regret bound

$$R_{n_m}(\theta_m) \leq 2\left(\sigma\sqrt{d\log\left(\frac{1+N_{m,n_m}L^2/\lambda}{\delta}\right)} + \lambda^{1/2}||\overline{\theta}_{m,t}||\right)\sqrt{\alpha_m n_m d \log\Big(1+N_{m,n_m}L^2/\lambda\Big)} + 2\sum_{t=1}^{n_m}\Gamma_{m,t}(x_t).$$

□

7.3 Proofs of Section 4

Proof of Lemma 5.2. We first introduce an alternative *average multi-task* parameter vector as

$$\overline{\theta}'_{m,t} = \frac{1}{N_{m,t}}\Big(\sum_{j=1}^{m-1} n_j \theta_j + t\theta_m\Big).$$

Unlike $\overline{\theta}_{m,t}$, this new definition is a direct (weighted) average of the parameter vectors of all the tasks. For any task θ_j, $j = 1, \ldots, m$ we have that

$$\|\theta_j - \overline{\theta}'_{m,t}\| \leq \frac{1}{N_{m,t}}\Big(\sum_{j'=1}^{m-1} n_j \|\theta_j - \theta_{j'}\| + t\|\theta_j - \theta_m\|\Big) \leq \epsilon,$$

where ϵ is upper-bounding any pairwise distance between tasks. As a result we have that

$$\begin{aligned}
\|\overline{\theta}_{m,t}\| &= \|\overline{\theta}_{m,t} - \overline{\theta}'_{m,t} + \overline{\theta}'_{m,t}\| \\
&\leq \|\overline{\theta}'_{m,t}\| + \left\|\widetilde{A}^{-1}_{m,t}\Big(\sum_{j=1}^{m-1} A_j(\theta_j - \overline{\theta}'_{m,t}) + A_{m,t}(\theta_m - \overline{\theta}'_{m,t})\Big)\right\| \\
&\leq S + \epsilon\Big(\sum_{j=1}^{m-1} \|\widetilde{A}^{-1}_{m,t}A_j\| + \|\widetilde{A}^{-1}_{m,t}A_{m,t}\|\Big).
\end{aligned}$$

The analysis of the bias proceeds with very similar steps as

$$\begin{aligned}
\Gamma_{m,t}(x) = |x^\top(\overline{\theta}_{m,t} - \theta_m)| &\leq \|x\|_2 \|\overline{\theta}_{m,t} - \theta_m\|_2 \\
&\overset{(a)}{\leq} L\|\overline{\theta}_{m,t} - \theta_m\|_2 \\
&\leq L\epsilon\Big(\sum_{j=1}^{m-1} \|\widetilde{A}^{-1}_{m,t}A_j\| + \|\widetilde{A}^{-1}_{m,t}A_{m,t}\|\Big)
\end{aligned}$$

where **(a)** follows from the upper bound on the norm of the arms.

We can introduce a convenient rewriting of the previous expressions by introducing matrices $C_j \in \mathbb{R}^{dd}$ such that $A_j = n_j\Big(\frac{1}{n_j}\sum_{s=1}^{n_j} x_{j,s}x_{j,s}^\top\Big) = n_j C_j$, so that $\widetilde{A}_{m,t} = N_{m,t}\Big(\sum_{j=1}^{m}\frac{n_j}{N_{m,t}}C_j + \frac{t}{N_{m,t}}C_{m,t}\Big) = N_{m,t}\widetilde{C}_{m,t}$. Then we obtain the final expression

$$\|\overline{\theta}_{m,t}\| \leq S + \epsilon\Big(\sum_{j=1}^{m-1}\frac{n_j}{N_{m,t}}\big\|\widetilde{C}_{m,t}^{-1}C_j\big\| + \frac{t}{N_{m,t}}\big\|\widetilde{C}_{m,t}^{-1}C_{m,t}\big\|\Big).$$

The *average disalignment of tasks*

$$\Upsilon_{m,t} = \sum_{j=1}^{m-1}\|\widetilde{A}_{m,t}^{-1}A_j\| + \|\widetilde{A}_{m,t}^{-1}A_{m,t}\|,$$

can be upper bounded as follows. Each term $\|\widetilde{A}_{m,t}^{-1}A_j\|$ is

$$\|\widetilde{A}_{m,t}^{-1}A_j\| \leq \frac{\lambda_{\max}(A_j)}{\sum_{j'=1}^{m-1}\lambda_{\min}(A_{j'})} \leq \frac{Ln_j}{(m-1)\min\limits_{j'=1,\dots,m-1}\lambda_{\min}(A_{j'})}.$$

As a result we obtain

$$\begin{aligned}\Upsilon_{m,t} &\leq \sum_{j=1}^{m-1}\frac{Ln_j}{(m-1)\min\limits_{j'=1,\dots,m-1}\lambda_{\min}(A_{j'})} + \frac{Lt}{(m-1)\min\limits_{j'=1,\dots,m-1}\lambda_{\min}(A_{j'})},\\ &\leq \frac{L\max\limits_{j=1,\dots,m} n_j}{\min\limits_{j'=1,\dots,m-1}\lambda_{\min}(A_{j'})}.\end{aligned}$$

□

Chapter 6
Conclusion

Summary

In this thesis we focused on three sample allocations problems in the stochastic linear bandit setting where the performance measure differs from the standard cumulative regret widely studied in the bandit literature.

- In Chapter 3 we focused on of the best-arm identification (BAI), for which we introduced the first characterization of the complexity of the BAI task. Then, we introduced static and adaptive sample allocation strategies designed to identify the best arm with a fixed confidence, while minimizing the number of observed rewards needed to do so. We pointed out the importance of exploiting the global linear structure to improve the estimate of the reward of near-optimal arms and showed how pulls to sub-optimal arms might be critical to obtain the information needed to distinguish between near-optimal arms. Also, we gave sample complexity guarantees for our algorithms and provided a simple empirical evaluation of their performance.

- Then, in Chapter 4, we studied the properties required for an adaptive allocation strategy to reach OED criteria relevant in a linear bandit setting. For the G-optimal design (which minimizes the estimation error of each arm) and then extending to the closely related V-optimal design (which minimizes the average estimation error over all the arms), we proposed adaptive allocation strategies and a preliminary analysis of the properties required by online learning algorithms to obtain good performances under two heteroscedastic noise models. In both cases the joint estimation-optimization task proved to be a challenging task in the global linear setting, suggesting that index policies based on the confidence intervals might not be easily applicable when the performance measures is different that the traditional cumulative regret.

- Lastly, in Chapter 5 we formalized the transfer learning problem in the linear bandit setting. In contrast to previous work on transfer in sequential decision-making settings, we considered the case where all tasks can be different from each other. We studied a simple transfer mechanism where samples are incrementally transferred from past tasks with the objective of reducing the estimation error in the current task. We proposed multi-task algorithms that manage to avoid negative transfer and we provided a theoretical analysis of their performance, showing

that our proposed multi-task strategies may already be effective in reducing the per-task regret. Finally, we studied the empirical performance of multi-task algorithms in a simple synthetic problem.

Future work

The work on these three problems opens the way for an important number of closely-connected problems and immediate extensions. For instance, regarding the BAI problem, an immediate extension is the study of the problem in a fixed-budget setting. Likewise, other problems of pure-exploration that have been already analyzed in the MAB setting could be also treated in the linear bandit case. One can naturally think at the problem of identifying the m-best arms, at the design of algorithms for the setting where the set of arms is very large (or infinite), or at the multi-bandit BAI identification in a linear setting. The design of efficient strategies for satisfying optimality criteria leaves an important number of open problems. The immediate ones concern the analysis of the allocation strategies we proposed for the G-allocation with heteroscedastic multivariate noise and for the heteroscedastic adaptive V-optimal allocation we proposed. Globally, an interesting future direction is the study of how arm indices could be built and which are the proper tools for obtaining provable efficient adaptive allocations strategies reaching known OED criteria. As for the transfer in linear bandits, immediate extensions include the relaxation of the notion of similarity, the study of the problem of bias estimation, and an investigation of more sophisticated transfer strategies (such as the transfer of solutions mechanism).

More in general, the three problems studied here for the linear bandit setting, could also be investigated in other types of bandit problems. For instance, it is interesting to study what is the complexity and which type of samples allocations strategies are efficient in meeting the optimality criteria studied here when the setting changes to combinatorial bandits, bandits with side observations, or partial monitoring problems. Another important research direction is the design and study of Bayesian strategies adapted to the problems that we studied throughout the thesis.

Bibliography

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, 2011. (→ pages 21, 22, 23, 78, 112, 120, 121, and 123.)

R. Agrawal. Sample Mean Based Index Policies with O(log n) Regret for the Multi-Armed Bandit Problem. *Advances in Applied Probability*, 27(4):pp. 1054–1078, 1995. ISSN 00018678. (→ page 18.)

S. Ahipaşaoğlu. *Solving ellipsoidal inclusion and optimal experimental design problems: theory and algorithms*. PhD thesis, Cornell University, 2009. (→ page 73.)

A. Antos, V. Grover, and C. Szepesvári. Active learning in multi-armed bandits. *Theoretical Computer Science*, 2009. (→ page 69.)

A. Antos, V. Grover, and C. Szepesvári. Active Learning in Heteroscedastic Noise. *Theoretical Computer Science*, 411:2712–2728, June 2010. (→ pages 13, 74, and 76.)

J.-Y. Audibert, R. Munos, and C. Szepesvari. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410:1876–1902, 2009. (→ page 19.)

J.-Y. Audibert, S. Bubeck, and R. Munos. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, 2010. (→ pages 13, 25, 30, and 32.)

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002. (→ pages 10, 13, 19, and 22.)

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002. (→ pages 18 and 19.)

R. Bechhofer, J. Kiefer, and M. Sobel. Sequential identification and ranking procedures. *The University of Chicago Press*, 1968. (→ page 33.)

P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Ann. Statist.*, 36(1):199–227, 02 2008. (→ page 80.)

J. Bien and R. J. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98 (4):807–820, 2011. (→ page 80.)

M. Bouhtou, S. Gaubert, and G. Sagnol. Submodularity and randomized rounding techniques for optimal experimental design. *Electronic Notes in Discrete Mathematics*, 36:679–686, 2010. (→ page 46.)

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. (→ page 73.)

E. Brunskill and L. Li. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, Bellevue, WA, USA, August 11-15, 2013*, 2013. (→ pages 106 and 107.)

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. (→ page 19.)

S. Bubeck, R. Munos, and G. Stoltz. Pure Exploration in Multi-Armed Bandit Problems. In *Proceedings of the Twentieth International Conference on Algorithmic Learning Theory*, pages 23–37, 2009a. (→ page 25.)

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT)*, 2009b. ISBN 3-642-04413-1, 978-3-642-04413-7. (→ page 24.)

S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the International Conference in Machine Learning (ICML)*, pages 258–265, 2013. (→ pages 25 and 33.)

M. Burger and S. J. Osher. A survey on level set methods for inverse problems and optimal design. *European Journal of Applied Mathematics*, null:263–301, 4 2005. ISSN 1469-4425. (→ page 46.)

O. Cappé, A. Garivier, O. Maillard, R. Munos, and G. Stoltz. Kullback-leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013. (→ page 19.)

A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. Upper-Confidence-Bound Algorithms for Active Learning in Multi-armed Bandits. In *Proceedings of the Twenty-Second International Conference on Algorithmic Learning Theory*, pages 189–203, 2011. (→ pages 69, 74, 76, and 77.)

W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual Bandits with Linear Payoff Functions. 2011. (→ page 110.)

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning - ICML*, 2008. (→ page 106.)

K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9:1757–1774, 2008. (→ pages 107 and 118.)

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic Linear Optimization under Bandit Feedback. In *COLT 2008*, pages 355–366, 2008. (→ pages 23 and 113.)

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006. (→ pages 25, 30, and 33.)

V. Fedorov. *Theory of Optimal Experiments*. Academic Press, 1972. (→ pages 25 and 61.)

W. A. Fuller and J. N. K. Rao. Estimation for a Linear Regression Model with Unknown Diagonal Covariance Matrix. *The Annals of Statistics*, 6(5):pp. 1149–1158, 1978. ISSN 00905364. (→ page 78.)

V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-Bandit Best Arm Identification. In *Proceedings of the Advances in Neural Information Processing Systems 25*, pages 2222–2230, 2011. (→ page 25.)

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012. (→ pages 25, 31, 32, 33, and 34.)

C. Gentile, S. Li, and G. Zappella. Online clustering of bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 757–765, 2014. (→ pages 106 and 107.)

M. Gheshlaghi Azar, A. Lazaric, and B. Emma. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26 - NIPS*, 2013. (→ pages 13, 106, and 107.)

M. Grant and S. Boyd. URL `http://cvxr.com/cvx/`. (→ page 96.)

M. D. Hoffman, B. Shahriari, and N. de Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 365–374, 2014. (→ page 34.)

J. Honda and A. Takemura. An asymptotically optimal policy for finite support models in the multiarmed bandit problem. *Machine Learning*, 85(3):361–391, 2011. (→ page 18.)

K. G. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil' UCB : An optimal exploration algorithm for multi-armed bandits. In *Proceeding of the 27th Conference on Learning Theory (COLT)*, 2014. (→ pages 25 and 34.)

C. Jennison, I. M. Johnstone, and B. W. Turnbull. Asymptotically Optimal Procedures for Sequential Adaptive Selection of the Best of Sebveral Normal Means. In S. S. Gupta and J. O. Berger, editors, *Statistical Decision Theory and Related Topics {III}*, pages 55 – 86. Academic Press, 1982. ISBN 978-0-12-307502-4. (→ page 33.)

S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC Subset Selection in Stochastic Multi-armed Bandits. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2012. (→ page 34.)

Z. Karnin, T. Koren, and O. Somekh. Almost Optimal Exploration in Multi-Armed Bandits. In *Proceedings of the Thirtieth International Conference on Machine Learning*, 2013. (→ page 33.)

E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceedings of the 26th Conference on Learning Theory (COLT)*, pages 228–251, 2013. (→ pages 25 and 33.)

E. Kaufmann, O. Cappé, and A. Garivier. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 2015. http://arxiv.org/abs/1407.4443. (→ pages 25, 34, 36, and 37.)

J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960. (→ pages 43, 60, and 72.)

I. Kuzborskij and F. Orabona. Learning by transferring from auxiliary hypotheses. *CoRR*, abs/1412.1619, 2014. (→ page 115.)

T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985. ISSN 0196-8858. (→ page 17.)

T. L. Lai and C. Z. Wei. Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *Ann. Statist.*, 10(1): 154–166, 03 1982. (→ page 113.)

A. Lazaric and M. Restelli. Transfer from Multiple MDPs. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1746–1754, 2011. (→ pages 106, 107, and 118.)

A. Lazaric, M. Restelli, and A. Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th Annual International Conference in Machine Learning*, pages 544–551, 2008. (→ page 107.)

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pages 661–670, 2010. ISBN 978-1-60558-799-8. (→ pages 11, 22, 107, and 110.)

J. J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, pages 118–126, 2011. (→ page 107.)

A. Llamosi, A. Mezine, F. d'Alché Buc, V. Letort, and M. Sebag. Experimental design in dynamical system identification: A bandit-based active learning approach. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8725 of *Lecture Notes in Computer Science*, pages 306–321. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-44850-2. (→ page 69.)

O. Maillard and S. Mannor. Latent bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 136–144, 2014. (→ pages 106 and 107.)

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. (→ page 115.)

O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Proceedings of the Advances in Neural Information Processing Systems 7*, 1993. (→ page 33.)

Merikoski, Jorma, Kumar, and Ravinder. Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes [electronic only]*, 4:150–159, 2004. (→ page 122.)

R. Munos. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7 (1):1–129, 2014. ISSN 1935-8237. (→ page 18.)

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010. ISSN 1041-4347. (→ page 106.)

E. Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *Ann. Math. Statist.*, 35(1):174–180, 03 1964. (→ page 33.)

A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *CVPR*, 2015. (→ page 107.)

F. Pukelsheim. *Optimal Design of Experiments*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2006. ISBN 9780898716047. (→ pages 25, 46, 47, 48, 50, 61, and 73.)

F. Pukelsheim and B. Torsney. Optimal Weights for Experimental Designs on Linearly Independent Support Points. *The Annals of Statistics*, 19(3):pp. 1614–1625, 1991. ISSN 00905364. (→ page 88.)

H. Robbins. Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952. (→ pages 13 and 15.)

G. Sagnol. Approximation of a maximum-submodular-coverage problem involving spectral functions, with application to experimental designs. *Discrete Appl. Math.*, 161 (1-2):258–276, Jan. 2013. ISSN 0166-218X. (→ page 48.)

R. Sibson. Discussion of a paper by hp wynn. *Journal of the Royal Statistical Society. Series B*, pages 181–183, 1972. (→ page 71.)

S. Silvey. Discussion of a paper by hp wynn. *Journal of the Royal Statistical Society. Series B*, pages 174–175, 1972. (→ page 71.)

M. Soare, A. Lazaric, and R. Munos. Active learning in linear stochastic bandits. In *NIPS 2013 Workshop on Bayesian Optimization in Theory and Practice*, 2013. (→ page 69.)

M. Soare, O. Alsharif, A. Lazaric, and J. Pineau. Multi-task linear bandits. In *NIPS 2014 Workshop on Transfer and Multi-task Learning*, 2014a. (→ page 105.)

M. Soare, A. Lazaric, and R. Munos. Best-arm identification in linear bandits. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 828–836. Curran Associates, Inc., 2014b. (→ page 29.)

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, pages 1015–1022, 2010. (→ page 69.)

M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009. (→ page 106.)

W. R. Thompson. On the likelihood that on unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933. (→ pages 12 and 15.)

D. M. Titterington. Optimal Design: Some Geometrical Aspects of D-Optimality. *Biometrika*, 62(2):pp. 313–320, 1975. ISSN 00063444. (→ page 72.)

M. Todd. On minimum-volume ellipsoids, May 2016. URL `http://people.orie.cornell.edu/miketodd/ublndeta.pdf`. Cornell University. (→ page 72.)

D. P. Wiens and P. Li. V-optimal designs for heteroscedastic regression. *Journal of Statistical Planning and Inference*, 145:125–138, 2014. (→ pages 26, 81, 84, 85, 86, 88, 89, 91, 99, and 100.)

W. K. Wong and R. Cook. Heteroscedastic g-optimal designs. *Journal of the Royal Statistical Society. Series B*, 55(4):871–880, 1993. ISSN 0035-9246. (→ page 26.)

K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 1081–1088, 2006. ISBN 1-59593-383-2. (→ page 49.)

Y. Yu. Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38(3), 2010. (→ page 47.)