Stochastic Graphical Bilinear Bandits

Geovani Rizk

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Université Paris-Dauphine

Stochastic Graphical Bilinear Bandits

Soutenue par
Geovani RIZK
Le 26/10/2022

Ecole doctorale n° ED 543
Ecole doctorale SDOSE

Spécialité
Informatique

Composition du jury :

Gilles, STOLTZ Directeur de recherche, CNRS / Université Paris-Saclay	*Président du jury*
Emilie, KAUFMANN Chargée de recherche, CNRS/Université de Lille	*Rapportrice*
Panayotis, MERTIKOPOULOS Chargé de recherche, CNRS/INRIA-LIG	*Rapporteur*
Vianney, PERCHET Professeur, ENSAE	*Examinateur*
Marta, SOARE Maître de conférences, Université d'Orléans	*Examinatrice*
Yann, CHEVALEYRE Professeur, Université Paris Dauphine PSL	*Co-directeur de thèse*
Rida, LARAKI Directeur de recherche, CNRS / Université Paris Dauphine PSL	*Co-directeur de thèse*

Dauphine | PSL
UNIVERSITÉ PARIS

Stochastic Graphical Bilinear Bandits

Geovani RIZK

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Université Paris-Dauphine – PSL Research University

in collaboration with

Huawei, Noah's Ark Lab, Paris

under the joint supervision of

Pr. Yann CHEVALEYRE,
Dr. Rida LARAKI,
Dr. Igor COLIN,
Dr. Albert THOMAS.

À mes parents

Remerciements

Je souhaite tout d'abord remercier Emilie Kaufmann et Panayotis Mertikopoulos d'avoir accepté d'être rapporteurs de ma thèse ainsi que Vianney Perchet, Gilles Stoltz et Marta Soare d'avoir accepté de faire partie du jury de ma soutenance de thèse. Vos retours et les discussions que nous avons pu avoir durant ma soutenance de thèse ont été très enrichissants.

Durant ces quatre dernières années, j'ai eu la chance de pouvoir être encadré par Yann Chevaleyre, Rida Laraki, Igor Colin et Albert Thomas qui ont su me guider tout au long de cette thèse et m'inculquer le métier de chercheur. Pour cela, je souhaiterai leur témoigner ma profonde gratitude et les remercier de toute l'aide qu'ils ont pu m'apporter, tant sur le plan professionnel que personnel.

Cette thèse effectuée d'une part au sein de l'équipe MILES du laboratoire LAMSADE de l'Université Paris Dauphine et d'autre part au sein de l'équipe Noah's Ark Paris de Huawei a représenté pour moi un cadre de travail propice à l'épanouissement scientifique et personnel, et cela a été possible grâce aux doctorants et membres permanents de ces deux équipes. C'est pourquoi je souhaite chaleureusement les remercier et leur témoigner que tous les moments (scientifiques comme moins scientifiques) partagés avec eux resteront un très bon souvenir.

Enfin, j'ai eu la chance d'être entouré par ma famille et mes amis qui ont su me supporter et m'épauler aux moments où j'en avais le plus besoin. Pour cela, je les remercie infiniment. Finalement, une partie de ce doctorat vous revient aussi.

Abstract

We introduce a new model called *Graphical Bilinear Bandits* where a learner (or a central entity) allocates arms to nodes of a graph and observes for each edge a noisy bilinear reward representing the interaction between the two end nodes. In this thesis, we study the best arm identification problem and the maximization of cumulative rewards. For the first problem, a learner wants to find the graph allocation maximizing the sum of the bilinear rewards obtained through the graph. For the second problem, during the learning process, the learner has to make a trade-off between exploring the arms to gain accurate knowledge of the environment and exploiting the arms that appear to be the bests to obtain the highest reward. Regardless of the learner's goal, the graphical bilinear bandit model reveals an underlying NP-Hard combinatorial problem that precludes the use of any existing best arm identification (BAI) or regret-based algorithms. For this reason, we first propose an α-approximation algorithm for the underlying NP-hard problem, and then tackle the two problems mentioned above. By efficiently exploiting the geometry of the bandit problem, we propose a random sampling strategy for the BAI problem with theoretical guarantees. In particular, we characterize the influence of the graph structure (e.g., star, complete or circle) on the convergence rate and propose empirical experiments that confirm this dependence. For the problem of maximizing the cumulative rewards, we present the first regret-based algorithm for graphical bilinear bandits using the principle of optimism in the face of uncertainty. Theoretical analysis of the presented method gives an upper bound of $\tilde{O}(\sqrt{T})$ on the α-regret and highlights the impact of the graph structure on the convergence rate. Finally, we demonstrate by various experiments the validity of our approaches.

Résumé

Nous introduisons un nouveau modèle appelé *Bandits Bilinéaires Graphiques* où un apprenant (ou une entité centrale) alloue des bras aux noeuds d'un graphe et observe pour chaque arête une récompense bilinéaire bruitée représentant l'interaction entre les deux noeuds associés. Dans cette thèse, nous étudions le problème d'identification du meilleur bras et la maximisation des récompenses cumulées. Pour le premier, un apprenant veut trouver l'allocation du graphe maximisant la somme des récompenses bilinéaires obtenues à travers le graphe. Pour le second problème, au cours du processus d'apprentissage, l'apprenant doit faire un compromis entre l'exploration des bras pour acquérir une connaissance précise de l'environnement et l'exploitation des bras qui semblent être les meilleurs pour obtenir la récompense la plus élevée. Quel que soit l'objectif de l'apprenant, le modèle de bandits bilinéaires graphiques révèle un problème combinatoire sous-jacent qui est NP-Dur et qui empêche l'utilisation de tout algorithme existant pour l'identification du meilleur bras (BAI) ou pour la maximisation des récompenses cumulées. Pour cette raison, nous proposons tout d'abord un algorithme d'α-approximation pour le problème NP-Dur sous-jacent, puis nous nous attaquons aux deux problèmes mentionnés ci-dessus. En exploitant efficacement la géométrie du problème du bandit, nous proposons une stratégie d'échantillonnage aléatoire pour le problème BAI avec des garanties théoriques. En particulier, nous caractérisons l'influence de la structure du graphe (par exemple, étoile, complet ou cercle) sur le taux de convergence et proposons des expériences empiriques qui confirment cette dépendance. Pour le problème de la maximisation des récompenses cumulées, nous présentons le premier algorithme basé sur le regret pour les bandits bilinéaires graphiques utilisant le principe d'optimisme face à l'incertitude. L'analyse théorique de la méthode présentée borne l'α-regret par $\tilde{O}(\sqrt{T})$ et souligne l'impact de la structure du graphe sur le taux de convergence. Enfin, nous démontrons par diverses expériences la validité de nos approches.

Contents

List of Figures and Tables

Notations

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^d$	Set of d-dimensional real-valued vectors
$\mathbb{R}^{d\times d'}$	Set of $d \times d'$-dimensional real-valued matrices
$[v]_i$	the i-th element of a vector $v \in \mathbb{R}^d$
$[\mathbf{A}]_{ij}$	The element at the i-th row and j-th column of $\mathbf{A} \in \mathbb{R}^{d\times d'}$
$\mathbf{S}_d^+$	The cone of all positive semi-definite matrices in $\mathbb{R}^{d\times d}$
$\langle u, v\rangle$	The scalar product between u and $v \in \mathbb{R}^d$: $\langle u, v\rangle \triangleq u^\top v$
$\|v\|_p$	The ℓ_p norm of a vector $v \in \mathbb{R}^d$: $\|v\|_p = \left(\sum_{i=1}^d \|[v]_i\|^p\right)^{1/p}$
$\|v\|_{\mathbf{A}}$	The Mahanalobis norm of $v \in \mathbb{R}^d$ with a matrix $\mathbf{A} \in \mathbf{S}_d^+$: $\|v\|_{\mathbf{A}} \triangleq \sqrt{v^\top \mathbf{A} v}$
$\|\mathbf{A}\|$	The spectral norm of a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$: $\|\mathbf{A}\| \triangleq \sup_{x:\|x\|_2=1} \|\mathbf{A}x\|_2$
$\|\mathbf{A}\|_F$	The Frobenius norm of a matrix $\mathbf{A}$: $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^d \|[A]_{ij}\|^2}$
$\|\mathcal{X}\|$	Cardinality of the finite set $\mathcal{X}$
$\mathcal{S}_\mathcal{X}$	$\|\mathcal{X}\|$-dimensional simplex: $\mathcal{S}_\mathcal{X} \triangleq \{\gamma \in \mathbb{R}^{\|\mathcal{X}\|} \mid \sum_{x\in\mathcal{X}} \gamma_x = 1\}$
$\mathbf{I}_d$	$d \times d$ identity matrix
$\mathcal{N}(\cdot,\cdot)$	The Gaussian distribution
$\mathbb{P}(\cdot)$	The probability of a random event
$\mathbb{E}[\cdot]$	The expectation of a random event
$\mathbb{1}[\cdot]$	The indicator function: $\mathbb{1}(\mathcal{E}) = 1$ if $\mathcal{E}$ is true, 0 otherwise
$O(\cdot)$	The Landau notation: $f(T) = O(g(T)) \Leftrightarrow \limsup_{T\to\infty} \frac{f(T)}{g(T)} < \infty$
$o(\cdot)$	The Landau notation: $f(T) = o(g(T)) \Leftrightarrow \lim_{T\to\infty} \frac{f(T)}{g(T)} = 0$
$\wedge$	The logical AND
$\vee$	The logical OR
$\otimes$	The Kronecker product
$\text{vec}\,(\mathbf{A})$	The vector in $\mathbb{R}^{d^2}$ which is the concatenation of all the columns of $\mathbf{A} \in \mathbb{R}^{d\times d}$

Abbreviations

i.e.,	*id est*
e.g.,	*exempli gratia*
cf.	*confer*
i.i.d.	identically and independently distributed
MAB	Multi-Armed Bandits
MA-MAB	Multi-Agent Multi-Armed bandits
BAI	Best Arm Identification
OFU	Optimism in the Face of Uncertainty
UCB	Upper Confidence Bound
GBB	Graphical Bilinear Bandits
ETC	Explore-Then-Commit

1 Introduction

Contents

1.1 Context & motivations

This thesis aims at solving centralized multi-agent problems that involve pairwise interactions between agents. Configuring antennas in a wireless cellular network [89] is an example of those problems: the choice of a parameter for an antenna has an impact on both its own signal quality and that of each of its neighboring antennas due to signal interference. Likewise, in a wind farm, the adjustment of a turbine blade not only impacts its own energy collection efficiency but also that of its neighbors' due to wind turbulence [13, 36]. By considering each antenna or turbine blade as an agent, these problems can be modeled as a multi-agent multi-armed bandit problem (MA-MAB) [13] with the knowledge of a coordination graph [47] where each node represents an agent and each edge represents an interaction between two agents. A multi-armed bandit problem (MAB) is a sequential decision problem where a learner must take an action (also called arm) at each iteration and gets a (possibly perturbed) associated reward that informs about the quality of the chosen action. Naturally, the learner does not know the distribution of the reward for each possible action. The learner may have very different goals, such as maximizing the rewards accumulated during the process, or in a minimum number of tries and regardless of the accumulated rewards, inferring which action is the best to choose *i.e.*, the most rewarding. Hence, a multi-agent multi-armed bandit is the setting where several agents face a multi-armed bandit problem. In the bandit literature, one can distinguish unstructured and structured bandits. While the unstructured bandit considers that playing an action and getting the associated reward does not allow to deduce anything about the distribution of rewards of other actions, the structured one includes the bandit settings where the rewards of the different actions share a common parameter [59]. For instance, a popular structured bandit setting is the linear bandit [11] where the reward associated with any action is linearly dependent on an unknown parameter vector θ. Hence at a given time, choosing an action and receiving its associated reward gives information about θ and by definition also about the rewards of all other actions. Here, we are interested in such structured environments and on that matter we present a novel multi-agent structured bandit called

Graphical Bilinear Bandits. The specificity of this environment lies in the interdependence of the rewards obtained by the neighboring agents in the graph and in the assumption that these rewards are bilinear, which appears to us as the natural extension of linear rewards when agents are pairwise dependent. Indeed, while MA-MAB problems have been studied in the setting of unstructured bandits with independent and dependent agents (see *e.g.*, [3, 7, 13, 15, 18, 49, 58, 85, 87, 101]), only the setting of structured bandits with independent agents has been explored (see *e.g.*, [6, 28, 30]). Through this thesis and the papers it refers to, we want to lay a first stone to the building.

1.2 Problem setting

1.2.1 Stochastic Graphical Bilinear Bandits

Let $\mathcal{G} = (V, E)$ be the directed graph defined by V the finite set of nodes representing the agents and E the set of edges representing the agent interactions. We assume that if $(i, j) \in E$ then $(j, i) \in E$. The graph could be considered as undirected but we assume that the interactions between two neighbors are not necessarily symmetrical with respect to the obtained rewards, so we choose to keep the directed graph to emphasize this potential asymmetry. For all agent $i \in V$, we denote $\mathcal{N}_i$ the set of its neighboring agents. Let $n = |V|$ denote the number of nodes, $m = |E|$ the number of edges and let $\mathcal{X} \subset \mathbb{R}^d$ be a finite arm set where $K = |\mathcal{X}|$ denote the number of arms. The graphical bilinear bandit with a graph $\mathcal{G}$ and an arm set $\mathcal{X}$ consists in the following sequential decision problem:

Stochastic Graphical Bilinear Bandits

For each round $t > 0$,

1. Each agent $i \in V$ chooses an arm $x_t^{(i)}$ in $\mathcal{X}$
2. Then, each agent $i \in V$ receives a noisy bilinear reward for each of its neighbors $j \in \mathcal{N}_i$

$$y_t^{(i,j)} = x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} + \eta_t^{(i,j)} \quad , \tag{1.1}$$

where $\mathbf{M}_\star \in \mathbb{R}^{d \times d}$ is an unknown matrix, and $\eta_t^{(i,j)}$ a zero-mean σ-sub-Gaussian random variable.

The reward $y_t^{(i,j)}$ reflects the quality of the interaction between the neighboring nodes i and j when pulling respectively the arm $x_t^{(i)}$ and $x_t^{(j)}$ at time t. The bilinear setting appears as a natural extension of the commonly studied linear setting to model the interaction between two agents.

Note that this setting can be considered either in a decentralized setting where agents take actions without consultation with others agents or in the centralized setting where a central entity chooses the arms of all the agents as well as aggregates the obtained rewards and designs a global strategy for the agents in the graph.

In this thesis, we only consider the centralized setting where a central entity manages all the agents, chooses at each time t the joint arm $(x_t^{(1)}, \dots, x_t^{(n)})$ and then receives the associated rewards $y_t^{(i,j)}$ for all $(i,j) \in E$. We illustrate the sequential decision problem at a given round t in Figure 1.1.

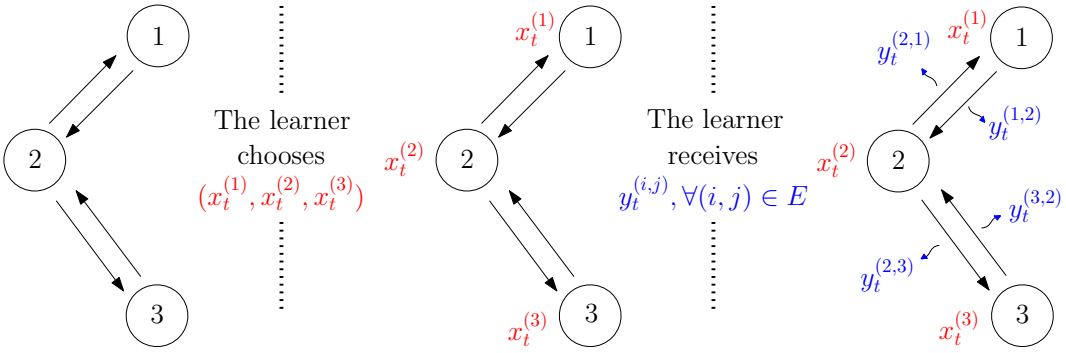


Figure 1.1: Illustration of the learner's decision process at a given round t for a simple graph of three nodes

1.2.2 Objectives

As briefly mentioned earlier, there are two different main goals that a learner (here the central entity) may want to achieve in a bandit problem.

Identifying the best joint arm. The first objective that we want to deal with in this thesis is where the learner is interested in finding within a minimum of rounds the best joint arm $(x_\star^{(1)}, \dots, x_\star^{(n)})$ that maximizes the expected global reward over the graph:

$$(x_\star^{(1)}, \dots, x_\star^{(n)}) = \underset{(x^{(1)},\dots,x^{(n)}) \in \mathcal{X}^n}{\arg\max} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)} \quad .$$

This objective implies that the central entity do not mind choosing a suboptimal joint arm $(x_t^{(1)}, \dots, x_t^{(n)})$ at each time t as long as it gives enough information on the unknown parameter $\mathbf{M}_\star$ in order to construct an accurate estimate $\hat{\mathbf{M}}$. This objective is known as *pure exploration* or *best arm identification* [10, 24].

Maximizing the cumulative rewards. The second objective is the most commonly considered in the bandit literature where the learner wishes to maximise the sum of the (expected) rewards obtained over the rounds. In our setting, the central entity wants to maximize the cumulative expected global rewards given by

$$\sum_{t=1}^{T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \quad .$$

While the first goal allows the learner to be in a pure exploration setting, regardless of the rewards obtained throughout the process, the goal of maximizing the cumulative rewards requires a trade-off between exploring the different possible arms to have an accurate estimate $\hat{\mathbf{M}}$ of $\mathbf{M}_\star$ and exploiting the arms that seems to be the most optimal given $\hat{\mathbf{M}}$ in order to obtain the maximum cumulative rewards.

In both objectives (*i.e.*, the best arm identification or the maximization of the cumulative rewards) and given an estimate $\hat{\mathbf{M}}$, the learner will have to solve at some point the following optimization problem

$$\max_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top}\hat{\mathbf{M}}x^{(j)} \quad . \tag{1.2}$$

Indeed, for the best arm identification, this optimization problem must be solved at the end when the learner wants to return the best joint arm given the estimate $\hat{\mathbf{M}}$ constructed during the learning procedure. For the maximization of the cumulative rewards, this optimization problem may need to be solved during the learning procedure when the learner wants to exploit and return the best estimated joint arm given its current knowledge of the environment which is the constructed estimate $\hat{\mathbf{M}}$.

Solving this optimization problem is not trivial, so for both objectives we consider the common underlying objective of solving this problem.

1.2.3 Outline of the thesis and contributions

In Chapter 2, we introduce and formalize the stochastic multi-armed bandit problem and more particularly the stochastic linear bandit problem with the algorithms and guarantees that exist for the best arm identification problem and the maximization of the cumulative rewards. Indeed, many tools developed in the corresponding literature will be used to solve the problems related to the graphical bilinear bandits setting. Then we put our graphical bilinear bandits model in perspective with some multi-agent bandit models that use structured bandits and bandits in graphs.

In Chapter 3, we tackle the underlying objective of solving the optimization problem given in (1.2). For this part we consider that the learner already has the best estimate $\hat{\mathbf{M}} = \mathbf{M}_\star$. We show that the problem is NP-Hard and we give two α-approximation algorithms with $\alpha \geq 1/2$.

In Chapter 4, we formalize the best arm identification problem relative to the graphical bilinear bandits. By efficiently exploiting the geometry of this bandit problem, we propose an allocation strategy based on randomized sampling with theoretical guarantees. In particular, we characterize the influence of the graph structure (e.g. star, complete or circle) on the convergence rate and propose empirical experiments that confirm these dependencies.

In Chapter 5, we present a regret-based algorithm (*i.e.*, an algorithm that aims to maximize the cumulative rewards) for graphical bilinear bandits using the principle of optimism in the face of uncertainty. Theoretical analysis of this new method yields an upper bound of $\tilde{O}(\sqrt{T})$ on the α-regret (a useful measure that we introduce in chapter 2) and evidences the impact of the graph structure on the rate of convergence. We show through various experiments the validity of our approach.

Finally, in Chapter 6 we present the conclusion of this thesis and discuss the different research perspectives that the graphical bilinear bandits model offers.

2 Background

Contents

In this chapter, we first present the basics of the stochastic multi-armed bandit and the stochastic linear bandit. The different notions and algorithms that appear in the following sections do not cover the whole field and do not necessarily include the most recent or the most optimal ones since we only want to (i) introduce the reader to this domain and give the tools that allow a good understanding of the following chapters of this thesis and (ii) explain why the existing algorithms cannot be straightforwardly applied to the graphical bilinear bandit setting. For a more in-depth view of the field, we refer the reader to the book [59] which provides a detailed and comprehensive overview of bandit problems.

Besides, in the last section of this chapter, we present some specific works that model structured multi-agent bandits. Since the studied models are very different from ours, a direct comparison of the methods would not be appropriate, however they bring perspective to our work. Furthermore some of the methods and tools used in the cited papers may still be useful for graphical bilinear bandits problems.

2.1 An introduction to the stochastic multi-armed bandit problem

2.1.1 Motivations and formalization

The bandit problem was first introduced in [93] to model sequential clinical trials where a learner chooses for each patient a drug and then observes the associated effects. Then, the stochastic

multi-armed bandit problem was formalized in [79] to present the general sequential decision problem: it considers a learner who has access to several actions (most often called *arms*) and during a given number of rounds, the learner has to choose an arm at each round that will reveal an associated perturbed reward. The expected reward of each arm is unknown to the learner. One of the most popular cases illustrating this situation is that of a gambler who is in a casino and has access to slot machines with different expected payoffs. Given his or her budget, the learner has a finite number of tries and must choose which slot machine to play at each time and then obtain the associated payout. The common goal of the player is to maximize the cumulative payoffs, but other goals such as identifying the best slot machine in a minimum number of tries can be interesting.

Consider a finite set of arms $\mathcal{X}$ with $K = |\mathcal{X}|$ the number arms and a collection of distributions $\nu = \{P_x : x \in \mathcal{X}\}$. Given a time horizon $T > 0$, the learner faces the following sequential decision problem:

Stochastic Multi-Armed Bandit

For each round $t = 1, \dots, T$,

1. the learner chooses an arm x_t in a finite arm set $\mathcal{X}$
2. the environment samples a reward $y_t \in \mathbb{R}$ from P_{x_t} and reveals y_t to the learner.

In the next section, we present an algorithm that solves the problem of maximizing the cumulative rewards obtained during the learning procedure. The algorithms that solves the problem of identifying the best arm for multi-armed bandits use very different techniques than those used for linear bandits and by extension those we use for graphical bilinear bandits. For this reason and to maintain a clearer narrative, we do not present algorithms that tackle the best-arm identification problem for multi-armed bandits.

2.1.2 Maximizing the cumulative rewards

As we briefly mentioned in the previous section, a natural goal the learner might have is to maximize the cumulative rewards obtained during the learning process. We recall that the learner does not know the expected reward of each arm. Hence it has to *explore* and try out the different arms to have an estimate of their associated rewards, but also *exploit* the arm that appear to have the highest reward. These two subgoals are complementary: on the one hand, by exploring, the learner gets to correctly estimate the different expected rewards associated with all arms; however, this implies pulling suboptimal arms, which have bad impacts on the cumulative rewards. On the other hand, by exploiting, the learner gets to pull arms that appears to give the greatest reward; however, since other arms are disregarded, this could lead to less accurate estimation of the different expected rewards and thus to over exploiting arms that are actually suboptimal. Hence the learner has to do a tradeoff between *exploration* and *exploitation* in order to maximise the cumulative rewards.

Wanting to maximize cumulative noisy rewards is the same as wanting to pull at each round the arm that gives the best expected reward $\mu^\star$ where $\mu^\star = \max_{x \in \mathcal{X}} \int_{-\infty}^{\infty} u dP_x(u)$. Indeed, the

learner does not control the randomness coming from the environment, but pulling the arms with the highest expected reward would give him, in expectation, the highest cumulative rewards. By defining $\mu_x = \int_{-\infty}^{\infty} u dP_x(u)$ the expected reward when pulling arm x, we have more formally that maximizing the cumulative rewards is equivalent to minimizing the *pseudo-regret*

$$R(T) = T\mu^\star - \sum_{i=1}^{T} \mu_{x_t} \ ,$$

The notion of pseudo-regret simply describes the difference between what the learner would have done with full information, *i.e.*, pull the best arm during the T rounds, and what is actually done without initial information on the expected rewards of the arms. In [57], the authors show that asymptotically a regret of order $\log(T)$ is unavoidable. The objective for the learner is to have a pseudo-regret $R(T)$ such that

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0 \ .$$

This ensures that the learner chooses the optimal arm almost all the time when T tends to infinity.

One of the most popular algorithm for stochastic multi-armed bandit that does the exploration-exploitation tradeoff is the Upper Confidence Bound (UCB) algorithm [4, 12] using the principle of Optimism in the Face of Uncertainty (OFU). The idea of the algorithm is to select at each time t the arm that seems to be the most optimistically optimal. The notion of optimism takes into account the value of the estimated reward of the arms but also the number of samples used for the estimations, in other words the precision of the estimations. Indeed, at each time t and for each arm $x \in \mathcal{X}$, the learner has an estimate $\hat{\mu}_x$ of the reward of x. Hence, instead of only pulling the arm that has the highest estimated reward $\max_{x\in\mathcal{X}} \hat{\mu}_x$ (*i.e.*, only exploiting), it uses the upper confidence bound (UCB) on the estimate $\hat{\mu}_x$ and chooses the arm with the highest UCB. The less an arm has been pulled the higher the UCB. Intuitively it means that at each round, the learner either exploits with high confidence or explores other arms that have been less pulled and that might give (optimistically) a better reward. A lot of versions exist for the UCB algorithm, we recall here the original method presented in [12] in Algorithm 1.

Algorithm 1: UCB1

Input : arm set $\mathcal{X} = \{1, \dots, K\}$
Pull each arm $x \in \mathcal{X}$ once and set $\hat{\mu}_x$ the obtained reward and $n_x = 1$;
for $t = K+1$ *to* T **do**
  The learner pulls the arm $x_t = \arg\max_{x\in\mathcal{X}} \hat{\mu}_x + \sqrt{\frac{2\log(t)}{n_x}}$;
  $n_{x_t} = n_{x_t} + 1$;
  The learner observes $y_t \sim P_{x_t}$;
  $\hat{\mu}_{x_t} = \frac{(n_{x_t}-1)\hat{\mu}_{x_t} + y_t}{n_{x_t}}$; // Update the estimate
end

We state the guarantee of the algorithm on the pseudo-regret in the following proposition that we borrow from [12].

Proposition 2.1 (Theorem 1 in [12]). *For any $K \geq 1$ and arm set $\mathcal{X} = \{1, \dots, K\}$, if the policy UCB1 is run on $\mathcal{X}$ with the associated reward distribution $\nu = (P_x : x \in \mathcal{X})$ with support in $[0, 1]$, then, for any number of plays T, its pseudo-regret is such that*

$$R(T) \leq \left[8 \sum_{\substack{x \in \mathcal{X} \\ \mu_x < \mu^\star}} \frac{\ln T}{\mu^\star - \mu_x} \right] + \left(1 + \frac{\pi^2}{3}\right) \left(\sum_{x \in \mathcal{X}} \mu^\star - \mu_x \right) , \tag{2.1}$$

where for all $x \in \mathcal{X}$, $\mu_x = \int_{-\infty}^{\infty} u dP_x(u)$

This proposition tells us that, given the sublinear bound on the regret, the learner is constantly improving his choice, otherwise the regret would have been of order T with the learner remaining stuck and drawing a suboptimal arm. The reader can refer to [25] for instance for an improved and asymptotically optimal algorithm.

As stated in the introduction, in bandit theory, one can distinguish *unstructured bandits* from *structured bandits*. A bandit is called unstructured when it is not possible for the learner to learn information about one arm by drawing another. In other words, we can define an unstructured bandit as one where ν is a product of distributions (that may be of different classes), so that drawing an arm gives a reward from the associated distribution without helping the learner understand the other distributions. In contrast, structured bandits such as linear bandits allow the learner to pull an arm and infer information about the rewards of other arms.

2.2 The stochastic linear bandit problem

2.2.1 Formalization

We present the sequential decision problem for the linear bandit in the following:

Stochastic Linear Bandit

For each round $t = 1, \dots, T$,

1. the learner chooses an arm x_t in a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ with $d > 1$
2. the learner obtains from the environment the associated reward

$$y_t = \langle x_t, \theta_\star \rangle + \eta_t \tag{2.2}$$

where $\theta_\star \in \mathbb{R}^d$ is an unknown parameter vector, and η_t is a random variable sampled from a certain distribution

When the learner draws an arm $x \in \mathcal{X}$ and receives a noisy linear reward $\langle x, \theta_\star \rangle + \eta_t$, it gives information about $\theta_\star$ and by extension about the other expected reward $\langle x', \theta_\star \rangle$ for any $x' \in \mathcal{X}$. This setting is particularly interesting because it might be enough for the learner to draw d arms that span $\mathbb{R}^d$ to start having a reasonable estimate of $\theta_\star$. Therefore, when the number of arms is large, the learner does not necessarily have to draw all the arms to get a good estimate of $\theta_\star$ and thus the estimated rewards for all the arms. Notice that this bandit can be formulated for instance with $\nu = \{\mathcal{N}(\langle x, \theta_\star \rangle, \sigma) : x \in \mathcal{X}\}$ if we consider that the noise is a gaussian random variable. For the rest of this chapter, we consider that the noise terms are σ-sub-Gaussian random variables.

2.2.2 Experimental designs serving the pure exploration setting

When the objective of the learner is to identify the best arm $x_\star = \arg\max_{x \in \mathcal{X}} \langle x, \theta_\star \rangle$ within a minimum number of rounds, it is equivalent to look for the arm $x \in \mathcal{X}$ such that for all $x' \in \mathcal{X}$,

$$\langle x - x', \theta_\star \rangle \geq 0 \ . \tag{2.3}$$

However, one does not have access to $\theta_\star$, so we have to use its empirical estimate.

For $t > 0$, we consider a sequence of arms $\mathbf{x}_t = (x_1, \dots, x_t) \in \mathcal{X}^t$ and the corresponding noisy rewards $(y_1, \dots, y_t)$. We assume that the noise terms in the rewards are i.i.d., following a σ-sub-Gaussian distribution. Let $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \in \mathbb{R}^d$ be the solution of the ordinary least squares problem with $\mathbf{A}_t = \sum_{s=1}^t x_s x_s^\top \in \mathbb{R}^{d \times d}$ and $b_t = \sum_{s=1}^k x_s y_s \in \mathbb{R}^d$. We suppose that $\mathbf{A}_t$ is nonsingular for all $t > 0$. We first recall the following property.

Proposition 2.2 (Proposition 1 in [90])**.** *Let $c = 2\sigma\sqrt{2}$. For every fixed sequence $\mathbf{x}_t$, with probability $1 - \delta$, for all $t > 0$ and for all $x \in \mathcal{X}$, we have*

$$\left| x^\top \theta_\star - x^\top \hat{\theta}_t \right| \leq c \|x\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6t^2 K}{\delta \pi}\right)} \ .$$

As it is done in [90], let us consider a confidence set $\hat{S}(\mathbf{x}_t)$ centered at $\hat{\theta}_t \in \hat{S}(\mathbf{x}_t)$ and such that $\mathbb{P}\Big(\theta_\star \notin \hat{S}(\mathbf{x}_t)\Big) \leq \delta$, for some $\delta > 0$. Since $\theta_\star$ belongs to $\hat{S}(\mathbf{x}_t)$ with probability at least $1 - \delta$, one can stop pulling arms when an arm has been found, such that the condition (2.3) is verified for any $\theta \in \hat{S}(\mathbf{x}_t)$. More formally, the best arm identification task will be considered successful when an arm $x \in \mathcal{X}$ verifies the following condition for any $x' \in \mathcal{X}$ and any $\theta \in \hat{S}(\mathbf{x}_t)$:

$$\langle x - x', \hat{\theta}_t - \theta \rangle \leq \hat{\Delta}_t(x, x') \ ,$$

where $\hat{\Delta}_t(x, x') = (x - x')^\top \hat{\theta}_t$ is the empirical gap between x and x'.

Corollary 2.1. *For all $t > 0$, let $\hat{S}(\mathbf{x}_t)$ be such that*

$$\hat{S}(\mathbf{x}_t) = \left\{ \theta \in \mathbb{R}^d \ : \ \forall x \in \mathcal{X}, \, \forall x' \in \mathcal{X}, \langle x - x', \hat{\theta}_t - \theta \rangle \leq c \|x - x'\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6t^2 K^2}{\delta \pi}\right)} \right\} \ .$$

With probability $1 - \delta$, $\theta_\star$ is in $\hat{S}(\mathbf{x}_t)$.

Proof. Using the upper bound in Proposition 2.2 and replacing x by $(x - x')$, we get the result. □

Then, the stopping condition can be reformulated as follows:

$$\exists x \in \mathcal{X},\ \forall x' \in \mathcal{X},\ c\|x - x'\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6t^2K^2}{\delta\pi}\right)} \leq \hat{\Delta}_t(x, x') \ . \tag{2.4}$$

As mentioned in [90], by noticing that $\max_{(x,x')\in\mathcal{X}^2} \|x - x'\|_{\mathbf{A}_t^{-1}} \leq 2\max_{x\in\mathcal{X}} \|x\|_{\mathbf{A}_t^{-1}}$, an admissible strategy is to pull arms minimizing $\max_{x\in\mathcal{X}} \|x\|_{\mathbf{A}_t^{-1}}$ in order to satisfy the stopping condition as soon as possible. More formally, one wants to find the sequence of arms $\mathbf{x}_t^\star = (x_1^\star, \dots, x_t^\star)$ such that:

$$\mathbf{x}_t^\star \in \underset{(x_1,\dots,x_t)}{\arg\min} \max_{x'\in\mathcal{X}} \ {x'}^\top \left(\sum_{i=1}^{t} x_i x_i^\top\right)^{-1} x' \ . \tag{G-opt-$\mathcal{X}$}$$

This is known as *G-allocation* (see *e.g.*, [76, 90]) and is NP-hard to compute [32, 104]. One way to find an approximate solution is to rely on a convex relaxation of the optimization problem (G-opt-$\mathcal{X}$) and first compute a real-valued allocation $\gamma^\star \in \mathcal{S}_\mathcal{X}$ such that

$$\gamma^\star \in \underset{\gamma\in\mathcal{S}_\mathcal{X}}{\arg\min} \max_{x'\in\mathcal{X}} {x'}^\top \left(\sum_{x\in\mathcal{X}} \gamma_x x x^\top\right)^{-1} x' \ . \tag{G-relaxed-$\mathcal{X}$}$$

One could either use random sampling to draw arms as *i.i.d.*samples from the $\gamma^\star$ distribution or rounding procedures to efficiently convert each component in $\gamma^\star$ into an integer and thus constructed the optimal matrix $\mathbf{A}_t^{-1}$.

Another way to visualize the effect of the constructed covariance matrix $\mathbf{A}_t^{-1}$ is through the confidence ellipsoids which are of the form $\mathcal{E} = \{\theta \in \mathbb{R}^d, (\hat{\theta}_t - \theta)^\top \mathbf{A}_t^{-1} (\hat{\theta}_t - \theta) \leq \tau\}$ where τ depends on the confidence level. The ellipsoid associated with a confidence parameter $\delta \in (0, 1)$ represents the region that contains the true parameter $\theta_\star$ with probability $1 - \delta$ (see Figure 2.1). For a fixed confidence parameter δ, one would want to minimize the region covered by the ellipsoid to ensure that the approximated parameter $\hat{\theta}_t$ is as close as possible to the real parameter $\theta_\star$. Since the ellipsoid depends on $\mathbf{A}_t^{-1}$, one way to do so, instead of estimating $\hat{\theta}_t$ by choosing all the arms $x \in \mathcal{X}$ (also called experiments), is to select the one that are the most statistically efficient. This problem is known as *experimental design* [76] and one criterion that has been studied is the *G*-optimal design that minimises $\max_{x\in\mathcal{X}} x^\top \mathbf{A}_t^{-1} x$. The *G*-optimal design minimizes the worst possible predicted variance and one can see that this objective coincides exactly with the one formulated in (G-opt-$\mathcal{X}$).

To approximate the solution of (G-opt-$\mathcal{X}$), the authors of [90] give a greedy strategy that at each time t chooses the arm $x_t \in \mathcal{X}$ such that

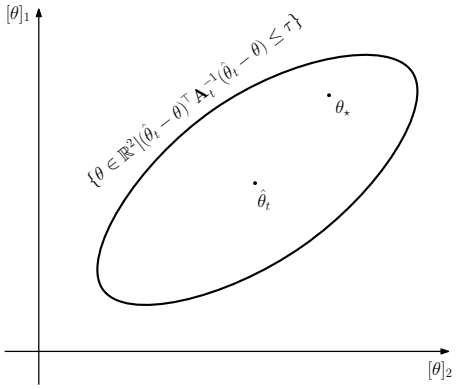


Figure 2.1: The δ-confidence level ellipsoid

$$x_t = \arg\min_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left(\mathbf{A}_{t-1} + xx^\top \right)^{-1} x' \quad . \tag{2.5}$$

The greedy strategy appears to be a fine way to approximate this problem because the number of pulls to satisfy the stopping condition is not known in advance, hence, converting directly the distribution γ to integers is not relevant. Moreover finding respectively the optimal sequence $(x_1, \ldots, x_t)$ and $(x_1, \ldots, x_{t+1})$ by rounding procedure and for a certain t with respect to the G-allocation strategy gives the same sequence modulo the extra x_{t+1} arm.

In Algorithm 2, we share the method.

Algorithm 2: Best-arm Identification in Linear Bandit : greedy G-Allocation strategy

Input : arm set $\mathcal{X} \subset \mathbb{R}^d$, confidence $\delta > 0$
Set $t = 0$; $\mathbf{A}_0 = \mathbf{I}_d$; $b_0 = 0$;
while (2.4) *is not true* **do**
  $t = t + 1$;
  $x_t = \arg\min_{x \in \mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left(\mathbf{A}_{t-1} + xx^\top \right)^{-1} x'$
  The learner observes $y_t = \langle x_t, \theta_\star \rangle + \eta_t$;
  $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t$;
end
return $\arg\max_{x \in \mathcal{X}} \langle x, \hat{\theta}_t \rangle$;

The authors give a guarantee on the sample complexity of any algorithm that gives a β-approximation of the solution of (G-opt-$\mathcal{X}$):

Proposition 2.3 ([90], Theorem 1). *If the G-allocation strategy is implemented with a β-approximate method and the stopping condition (2.4) is used, then with probability at least $1-\delta$, $\arg\max_{x\in\mathcal{X}}\langle x, \hat{\theta}_t\rangle = x_\star$ and*

$$t \leq \frac{16c^2 d(1+\beta)\log\left(\frac{6t^2K^2}{\delta\pi}\right)}{\Delta_{\min}^2} \ ,$$

where $\Delta_{\min} = \min_{x\in\mathcal{X}\setminus\{x_\star\}}\langle x_\star - x, \theta_\star\rangle$ and $c = 2\sigma\sqrt{2}$

For Algorithm 2, the greedy algorithm gives a β that depends on t, we note it β_t and is equal to $\frac{d+d^2+2}{2t}$.

2.2.3 Optimism in the face of uncertainty for linear bandits (OFUL)

For the objective of maximizing the cumulative rewards with a budget of T rounds, the existing methods use the same idea of optimism in the face of uncertainty but adapted to the linear setting. Indeed, at each time t we saw in the previous section that the learner can build an estimate $\hat{\theta}_t$. Although the objectives are completely different, one can nevertheless consider again the confidence ellipsoids of the form $\mathcal{E} = \{\theta \in \mathbb{R}^d, (\hat{\theta}_t - \theta)^\top \mathbf{A}_t^{-1}(\hat{\theta}_t - \theta) \leq \tau\}$ as shown in Figure 2.1. Given a confidence level δ, one can construct this ellipsoid and tell with probability $1-\delta$ that the true parameter vector $\theta_\star$ is in it. Hence, a strategy using the optimism in the face of uncertainty would be to select the arm that gives the best reward with respect to the best $\theta \in \mathcal{E}$. More formally at each time t the learner would select $x_t = \arg\max_{x\in\mathcal{X}} \max_{\theta\in\mathcal{E}}\langle x, \theta\rangle$. This method has been presented in [2] and we recall their approach in the following.

Let us define

$$\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \ , \tag{2.6}$$

where,

$$\mathbf{A}_t = \lambda \mathbf{I}_d + \sum_{s=1}^{t} x_s x_s^\top \ ,$$

with $\lambda > 0$ a regularization parameter and

$$b_t = \sum_{s=1}^{t} x_s y_s \ .$$

We also define the confidence set

$$C_t(\delta) = \left\{\theta : \|\theta - \hat{\theta}_t\|_{\mathbf{A}_t^{-1}} \leq \sigma\sqrt{d\log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S\right\} \ ,$$

where we assume that for any $x \in \mathcal{X}$, $\|x\|_2 \leq L$ and $\|\theta_\star\|_2 \leq S$. We know from Theorem 2 in [2] that with probability $1-\delta$, $\theta_\star$ is in $C_t(\delta)$ for all $t \in \{1, \dots, T\}$, and $\delta \in (0, 1]$.

Algorithm 3: OFUL Algorithm

Input : arm set $\mathcal{X}$
for $t = 1$ *to* T **do**
  $\left(x_t, \tilde{\theta}_{t-1}\right) = \arg\max_{(x,\theta)\in\mathcal{X}\times C_{t-1}} \langle x, \theta\rangle$;
  Obtain the rewards y_t;
  Compute $\hat{\theta}_t$ as in (2.6)
end
return $\hat{\theta}_t$

The pseudo-regret in the linear setting can be formulated as follows:

$$R(T) = \sum_{t=1}^{T} \langle x_\star, \theta_\star\rangle - \langle x_t, \theta_\star\rangle = \sum_{t=1}^{T} \langle x_\star - x_t, \theta_\star\rangle \ , \tag{2.7}$$

where we recall that $x_\star = \arg\max_{x\in\mathcal{X}} \langle x, \theta_\star\rangle$

Proposition 2.4 ([2], Theorem 3)**.** *Assume that for all t and all $x \in \mathcal{X}$, $\langle x, \theta_\star\rangle \in [-1, 1]$. Then with probability $1-\delta$, the pseudo-regret of the OFUL algorithm satisfies*

$$R(T) \leq 4\sqrt{Td\log(\lambda + TL/d)}\left(\sqrt{\lambda}S + \sigma\sqrt{2\log(1/\delta) + d\log(1 + TL/(\lambda d))}\right) \ ,$$

where for any $x \in \mathcal{X}$, $\|x\|_2 \leq L$ and $\|\theta_\star\|_2 \leq S$

In the next section, we present the bilinear bandit which appears as the natural extension of the linear bandit that models the interaction between two agents in the obtained rewards.

2.2.4 Bilinear bandits are linear bandits in a higher dimensional space

In [51], the authors introduce the *Bilinear Bandit* model where at each round t a learner chooses an arm x_t from a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ that contains K arms and a second arm x'_t from another finite arm set $\mathcal{X}' \subset \mathbb{R}^{d'}$ that contains K' arms, and obtains an associated reward that is bilinear with respect to the two chosen arms and an unknown parameter matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d'}$. More formally, this sequential decision problem is defined as follows:

Stochastic Bilinear Bandit

For each round $t = 1, \dots, T$,

1. the learner chooses an arm $x_t \in \mathcal{X}$ and $x'_t \in \mathcal{X}'$
2. the learner obtains from the environment the associated reward

$$y_t = x_t^\top \mathbf{M}_\star x'_t + \eta_t \tag{2.8}$$

where $\mathbf{M}_\star \in \mathbb{R}^{d \times d'}$ is an unknown parameter matrix, and η_t is a random variable sampled from a certain distribution

This kind of setting can model different real life applications, such as drug discovery applications [65] or in the context of recommender systems as explained in [51].

The bilinear reward can be written as a linear reward in a higher dimensional space:

$$y_t = \left\langle \mathrm{vec}\left(x_t x_t'^\top\right), \mathrm{vec}\left(\mathbf{M}_\star\right) \right\rangle + \eta_t \ , \tag{2.9}$$

where for any matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$, $\mathrm{vec}\left(\mathbf{A}\right)$ denotes the vector in $\mathbb{R}^{d^2}$ which is the concatenation of all the columns of $\mathbf{A}$.

Therefore, for both the best arm identification problem and the cumulative rewards maximization problem, solving this bilinear bandit problem is equivalent to solving a linear bandit problem of dimension $d \times d'$ with an arm set $\mathcal{Z} = \{\mathrm{vec}\left(xx'^\top\right) | (x, x') \in \mathcal{X} \times \mathcal{X}'\}$ of $K \times K'$ arms.

To directly apply the existing linear bandit algorithms to the bilinear bandit model, the learner must coordinate the choices of the two chosen arms (x_t, x'_t) at time t, which is equivalent to choosing an arm $z_t = \mathrm{vec}\left(x_t x_t'^\top\right) \in \mathcal{Z}$.

Although approaching a bilinear reward from a linear angle is useful, it is less trivial to use linear bandit algorithms for more complex models such as graphical bilinear bandits. Indeed, our model exposed in section 1.2 can be viewed as a bilinear bandit problem between each pair of neighbors, where each agent chooses an arm from a set of arms $\mathcal{X}$ (in our framework $\mathcal{X} = \mathcal{X}'$). Thus, if the learner coordinates the choice of two neighboring agents and chooses the pair $(x, x') \in \mathcal{X}$ and thus its associated arm in $\mathcal{Z}$, it constrains the choices related to all the pairs of neighbors $(j, k) \in E$ since it is already composed of the arm x' associated with agent j. Due to the interdependencies of the bilinear bandit problems, it is not possible to consider the graphical bilinear bandits as simple linear bandits in parallel and directly use the linear bandit algorithms present in the literature.

Another idea that one could have is to notice that since the unknown parameters matrix $\mathbf{M}_\star$ is common to all the edges (i, j) of the graph, the expected global reward at time t can also be written as the scalar product $\left\langle \sum_{(i,j)\in E} \mathrm{vec}\left(x_t^{(i)} x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star\right) \right\rangle$. Therefore, solving the best-arm identification problem or maximizing the cumulative rewards in the described graphical bilinear bandits reduces to solving the same problems in a global linear bandit. Although this trick allows

the use of classical algorithms in linear bandits, the number of joint arms grows exponentially with the number of nodes, which makes these methods impractical.

Of course, some tools are still very useful and relevant to solve the presented problems for the graphical bilinear bandits model and we use them in this thesis.

2.3 Multi-agent bandits and combinatorial bandits

2.3.1 Parallelizing contextual linear bandits

Centralized multi-agent bandit problems where the learner has to choose the actions of all the agents at each round implies to parallelize the learning process on the agents. In the context of linear rewards where all the agents share the same reward function (*i.e.*, , the same parameter $\theta_\star$), the authors in [30] give a detailed analysis of the problem of maximizing the cumulative rewards and show that a sublinear regret in T can be reached but with an additional cost specific to the parallelization.

More formally, they consider P agents, and at each round t, a context $\mathcal{X}_t^{(i)} \subset \mathbb{R}$ is revealed to agent i and a central entity has to choose the arm $x_t^{(i)} \in \mathcal{X}_t^{(i)}$ for each agent $i \in \{1, \dots, P\}$. Then the learner receives for all agent $i \in \{1, \dots, P\}$ the rewards $y_t^{(i)} = \langle x_t^{(i)}, \theta_\star \rangle + \eta_t^{(i)}$ where $\eta_t^{(i)}$ is a σ-sub-gaussian random variable.

Here the pseudo-regret is formulated as follows:

$$R(T) = \sum_{t=1}^{T} \sum_{p=1}^{P} \langle x_{\star,t}^{(i)}, \theta_\star \rangle - \langle x_t^{(i)}, \theta_\star \rangle \tag{2.10}$$

where $x_{\star,t}^{(i)} = \arg\max_{x \in \mathcal{X}_t^{(i)}} \langle x, \theta_\star \rangle$.

On can construct the estimate $\hat{\theta}_t$ as follows:

$$\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \ ,$$

where

$$\mathbf{A}_t = \lambda \mathbf{I}_d + \sum_{s=1}^{t} \sum_{i=1}^{P} x_s^{(i)} x_s^{(i)\top} \ ,$$

with $\lambda > 0$ a regularization parameter and

$$b_t = \sum_{s=1}^{t} \sum_{i=1}^{P} x_s^{(i)} y_s^{(i)} \ .$$

We define also the confidence set

$$C_t(\delta) = \left\{ \theta : \|\theta - \hat{\theta}_t\|_{\mathbf{A}_t^{-1}} \leq \sigma \sqrt{d \log\left(\frac{1 + tPL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S \right\} .$$

The authors show that applying the OFUL algorithm on each agent at time t where $\hat{\theta}_{t-1}$ and $C_{t-1}(\delta)$ aggregate the information of all the draws and rewards of the previous rounds, gives the following bound on the regret:

$$R(T) \leq \tilde{O}(d\sqrt{TP}) + O(dP \log(TP)) \tag{2.11}$$

where $\tilde{O}$ hides logarithmic factors.

Their analysis shows that parallelizing agents that play the same contextual bandit and applying the OFUL algorithm for each of them with an aggregation of the information at the end of each round to construct $\hat{\theta}_t$ and $C_t(\delta)$ give an upper bound on the regret that is the sum of the near-optimal regret of a single agent pulling TP arms plus a second term that represents the cost of parallelizing.

The similarities between the graphical bilinear bandits and their setting is that if $m = |E|$ is the number of edges in our considered graph, the central entity in our model plays m bilinear bandits in parallel that can be seen as m linear bandits in parallel. However, the main difference is that they consider independent agents whereas we assume interactions between the agents. In particular, the arm associated with each (bi-)linear bandit problem cannot be chosen independently at each round since some of them share a mutual information. So we have to deal with both the parallel aspect and the dependent aspect at the same time.

2.3.2 Bandit problems in graphs

Graphs are often used to bring structure to a bandit problem. But one can distinguish two main representations.

Single agent. In [99] and [67] for instance, the arms are the nodes of a graph and pulling an arm gives information on the rewards of the neighboring arms. This kind of setting considers only one agent which is out of the scope of this thesis. The reader can also refer to [98] for an account on such problems.

Multi-agents. As in [28], each node is an instance of a linear bandit and the neighboring nodes are assumed to have similar unknown regression coefficients.

More precisely, they consider an undirected graph $G = (V, E)$ where each node $i \in G$ has an associated parameter $\theta_\star^{(i)}$ where the authors make the assumption that

$$\sum_{(i,j) \in E} \|\theta_\star^{(i)} - \theta_\star^{(j)}\|_2^2 \tag{2.12}$$

is small compared to $\sum_{i\in V}\|\theta_\star^{(i)}\|_2^2$.

At each time t, the learner receives the index I_t of one of the nodes in V as well as a set of contexts $\mathcal{X}_t^{(I_t)}$ where it has to choose a context $x_t \in \mathcal{X}_t^{(I_t)}$ and then receives an associated linear reward of the form $\langle x_t, \theta_\star^{(i)}\rangle + \eta_t$. Note that at each round, only the instance of the linear bandit of node I_t is used, the learner does not choose an arm for each node of the graph. However, given the assumption that near-by nodes have similar associated parameter vectors $\theta_\star^{(i)}$, the learner may still get some information on the behaviors of other nodes.

Again, the main difference with our model is that the rewards of the nodes are independent, and although playing an arm at a given node may give information about its neighbor's rewards, the reward is not directly affected by the possible choices of its neighbors.

2.3.3 Link with unstructured multi-agents bandits

In the same way that a linear bandit with canonical arms can be seen as a classical multi-armed bandit[1], hence loosing its structured property, the graphical bilinear bandit can also be seen as an unstructured graphical multi-agent bandit when the arm set $\mathcal{X}$ is the canonical basis $\mathcal{X} = (e_1, \dots, e_d)$.

As mentioned in the introduction, some works have studied this unstructured setting. In particular, the authors in [7] consider a graph $\mathcal{G} = (V, E)$ where at each round t and for each node i in $V = \{1, \dots, n\}$, a learner receives a context $c_t^{(i)}$, then has to choose an arm $x_t^{(i)}$ and finally gets a global reward $F(\mathbf{x}_t, \mathbf{c}_t)$ where $\mathbf{x}_t = (x_t^{(1)}, \dots, x_t^{(n)})$ and $\mathbf{c}_t = (c_t^{(1)}, \dots, c_t^{(n)})$. They assume that the reward can be decomposed as a sum of subfunctions that depend only on subset of neighboring nodes. More formally, given a collection of subset $\mathcal{P} \subset 2^V$, we have

$$F(\mathbf{x}_t, \mathbf{c}_t) = \sum_{P\in\mathcal{P}} f_P(\mathbf{x}_P, \mathbf{c}_P) \tag{2.13}$$

where $\mathbf{x}_P = (x_t^{(i)})_{i\in P}$, $\mathbf{c}_P = (c_t^{(i)})_{i\in P}$ and f_P are unknown functions for any $P \in \mathcal{P}$.

Note that a subset $P \in \mathcal{P}$ contains only nodes that are neighbors one to another.

The main similarity with our framework is that the global function can be decomposed as the sum of local rewards that depend on the arms of neighboring nodes, which highlights the dependencies between the nodes. The reader can also refer to [13] for such works. However, all the algorithms that we presented for the graphical bilinear bandit leverage the structured aspect where pulling an arm informs about the rewards of the other arms through the unknown parameter matrix $\mathbf{M}_\star$, which is not done in the unstructured setting.

2.3.4 Combinatorial bandits

A combinatorial bandit consists of a sequential decision problem where a learner has access to a set of K arms, at each round t selects a subset of arms under some constraints and then receives the associated reward. More formally, consider the arm set $\mathcal{X} = \{0, 1\}^d$ where an arm $x \in \mathcal{X}$ is often

[1] If $\mathcal{X} = (e_1, \dots, e_d)$, when the learner chooses $x_t = e_i$, the reward $y_t = \langle e_i, \theta_\star\rangle + \eta_t = [\theta_\star]_i + \eta_t$, which does not share any information with the rewards of the other arms.

called a super-arm. When the i-th coordinate of $x \in \mathcal{X}$ equals 1 it means that the learner selects the i-th arm. An example of a constraint that the learner can have is to select an arm $x \in \mathcal{X}$ such that $\|x\|_1 \leq m$ for $m > 0$, which means that the learner can at most select a subset of m arms per round. The type of rewards associated with a super arm varies from one setting to another. For instance, let consider $\nu = \{P_1, \dots P_K\}$ where P_i is the distribution associated with the i-th arm. At each round, let us denote $X_{i,t}$ the random variable drawn at time t from P_i, and denote X_t the vector in $\mathbb{R}^K$ containing in its coordinates all the $X_{i,t}$ for all $i \in \{1, \dots, K\}$. The reward at time t associated with the super arm $x_t \in \mathcal{X}$ is given by

$$y_t = \langle x_t, X_{i,t} \rangle \ .$$

In this particular example, we can notice that the reward is linear *i.e.*, the reward y_t is the sum of all the rewards associated with the selected arms by the learner, but other forms of reward can be considered and the learner may not even know how it is calculated.

Semi-bandit feedback. In what is called the semi-bandit feedback, the learner receives each of the rewards $X_{i,t}$ if the i-th coordinate of x_t is equal to 1.

The number of super-arms being exponential in K and when we do not have the knowledge on how the reward is computed, these kinds of problems can be hard to solve and the knowledge of an oracle is often assumed to return the optimal super-arm to play at a round t according to the estimates of the learner. More precisely, given the estimates $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \dots, \hat{\mu}_K)$ of the expected reward associated with each arm, the learner asks the oracle which super-arm to play at time t. A relaxation in [31] considers an (α, β)-Approximation-oracle that returns with probability β, the α-optimal super-arm given the estimates. More formally, if $\text{opt}_{\hat{\boldsymbol{\mu}}}$ is the value of the best super-arm given $\hat{\boldsymbol{\mu}}$, the (α, β)-Approximation-oracle returns a super arm that gives at least an expected reward equals to $\alpha \cdot \text{opt}_{\hat{\boldsymbol{\mu}}}$.

Given that (α, β)-Approximation-oracle, the authors used the $\alpha\beta$-pseudo-regret [52] which is defined as follows :

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \alpha\beta \text{opt}_\star - y_t\right] \ ,$$

where $\text{opt}_\star$ is the optimal expected reward, and where the expectation is on the randomness of both the environment and the learner's policy.

The combinatorial bandit setting can be viewed as a similar model to ours in that we solve a bandit problem with an exponential number of arms and the underlying problem of returning an estimated best joint-arm (which can be viewed as a super arm under some particular constraint) at each round is NP-Hard and may require knowledge of an oracle. Although we do not assume knowledge of an oracle, we instead design an α-approximation algorithm to solve the underlying problem (see Chapter 3). And as is the case in the combinatorial bandit literature, the use of α-approximation algorithms makes the notion of α-regret a relevant measure of the performance

for our algorithms.[2] Moreover, we present algorithms that takes into account the graph structure, which has not been considered in the combinatorial framework.

[2] We drop the β parameter, because in our case it is equal to 1.

3 Computing the best allocation for known parameter matrices

Contents

In this chapter, we focus on the optimization problem of finding the joint arm $\left(x^{(1)}, \dots, x^{(n)}\right)$ that maximizes $\sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ while knowing the parameter matrix $\mathbf{M}_\star$. This problem being non-trivial, it is natural to understand the guarantees on the solutions with the full information of the matrix $\mathbf{M}_\star$ (we relax this assumption in the next chapters where the matrix $\mathbf{M}_\star$ will be considered as unknown). Hence the objective of this chapter is the following:

Objective: *Given the parameter matrix* $\mathbf{M}_\star$*, design an algorithm that returns the allocation of arms* $(x_\star^{(1)}, \dots, x_\star^{(n)})$ *that maximises* $\sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$

We follow the notations we established in section 1.2.1.

3.1 An NP-Hard problem

3.1.1 Reduction to the max-cut problem

We address the problem of finding the best joint arm given $\mathbf{M}_\star$ and we denote it as follows:

$$(x_\star^{(1)}, \dots, x_\star^{(n)}) = \underset{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n}{\arg\max} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)} \quad . \tag{3.1}$$

Notice that if the couple $(x_\star, x'_\star) = \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x'$ is such that $x_\star = x'_\star$ then finding the best joint arm is trivial and the solution is to assign $x_\star$ to all nodes. Conversely,

if $x_\star \neq x'_\star$, the problem may be harder: according to the graph $\mathcal{G}$, the optimal joint arm could either be composed exclusively of the couple $(x_\star, x'_\star)$ or be composed of other arms in $\mathcal{X}$. One might want to use dynamic programming as in [7] to solve this optimization problem, however in this particular setting, it would lead to use a non-polynomial time algorithm. Indeed, the following theorem states that, even with the knowledge of the true parameter $\mathbf{M}_\star$, identifying the best join-arm $(x_\star^{(1)}, \dots, x_\star^{(n)})$ is NP-hard with respect to the number of nodes n.

Theorem 3.1. *Consider a given matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ and a finite arm set $\mathcal{X} \subset \mathbb{R}^d$. Unless P=NP, there is no polynomial time algorithm guaranteed to find the optimal solution of*

$$\max_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star\, x^{(j)} \quad .$$

Proof. We prove the statement by reduction to the Max-Cut problem that is NP-Hard itself. Let $\mathcal{G} = (V, E)$ be a graph with $V = \{1, \dots, n\}$. Let $\mathcal{X} = \{e_0, e_1\}$, where $e_0 = (1, 0)^\top$ and $e_1 = (0, 1)^\top$. Let $\mathbf{M}_\star = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. For any joint arm assignment $\left(x^{(1)} \dots x^{(n)}\right) \in \mathcal{X}^n$, let $F \subseteq E$ be defined as $F = \left\{i : x^{(i)} = e_1\right\}$. Note that

$$\sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)} = \sum_{(i,j)\in E} \mathbb{1}\left[x^{(i)} \neq x^{(j)}\right] = 2 \times \sum_{(i,j)\in E} \mathbb{1}[i \in F, j \notin F],$$

where $\mathbb{1}[\cdot]$ is the indicator function.

The assignment $\left(x^{(1)}, \dots, x^{(n)}\right)$ induces a cut $(F, V\backslash F)$, and the value of the assignment is *precisely* twice the value of the cut. Thus, if there were a polynomial time algorithm solving our problem, this algorithm would also solve the Max-Cut problem.

□

Hence, given the true parameter matrix $\mathbf{M}_\star$, the learner is not guaranteed to find in polynomial time the joint arm $\left(x_\star^{(1)}, \dots, x_\star^{(n)}\right)$ maximizing the expected global reward. In the next sections, we give polynomial time approximation algorithms that have guarantees on the returned expected global reward with respect to the optimal one.

3.1.2 Approximation algorithm and guarantees

Given the true parameter $\mathbf{M}_\star$, the objective is to design an algorithm that returns a joint arm $\left(x^{(1)}, \dots, x^{(n)}\right)$ such that its associated expected global reward $y = \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)}$ has the guarantee to be close to the optimal expected global reward $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top}\mathbf{M}_\star x_\star^{(j)}$. In other words, we want to find a approximation parameter $0 < \alpha \leq 1$ such that, $y \geq \alpha y_\star$. Although the optimization problem we seek to solve can be found in the literature on *Markov Random Fields* when dealing with a multi-labeled graph (see *e.g.*, [5]), to the best of our knowledge, algorithms that give an approximation ratio on the optimal solution have not been explored.

Assumption 3.1 (Positive rewards)**.** *A classical assumption in the linear bandit literature is that expected rewards are positive. For any* $(x, x') \in \mathcal{X}^2$*, since the bilinear reward* $x^\top \mathbf{M}_\star x'$ *can be formulated as a linear reward* $\langle \mathrm{vec}\,(xx'^\top), \mathrm{vec}\,(\mathbf{M}_\star)\rangle$ *and that in the rest of this thesis we will use tools from the linear bandit literature, we make the same assumption. We consider that for any* $(x, x') \in \mathcal{X}^2$*, the associated expected reward* $x^\top \mathbf{M}_\star x'$ *is positive,* $x^\top \mathbf{M}_\star x' \geq 0$*.*

The approach we present in this section is first to consider the problem locally, *i.e.*, at the edge level. Indeed, let us consider two neighboring nodes i and j in V and only the expected rewards related to these nodes, which are $x^{(i)\top}\mathbf{M}_\star x^{(j)}$ and $x^{(j)\top}\mathbf{M}_\star x^{(i)}$. By summing those two quantities,[1] we get $x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(j)\top}\mathbf{M}_\star x^{(i)} = x^{(i)\top}\big(\mathbf{M}_\star + \mathbf{M}_\star^\top\big)x^{(j)}$ which represents the expected reward between the two neighbors (i) and (j). A local strategy that the central entity should carry out is thus to allocate $(x^{(i)}, x^{(j)}) = \arg\max_{(x,x')\in\mathcal{X}^2} x^\top\big(\mathbf{M}_\star + \mathbf{M}_\star^\top\big)x' = (x_\star, x'_\star)$. Naturally, while this local strategy is easy to apply for a couple of neighbors (i, j), it becomes infeasible to simultaneously extend it to all the other couples in the graph since some of them share the same nodes. However, one can learn something from this strategy, which is that given the optimal joint arm $(x_\star^{(1)}, \ldots, x_\star^{(n)})$, we have

$$x_\star^{(i)\top}\Big(\mathbf{M}_\star + \mathbf{M}_\star^\top\Big)x_\star^{(j)} \leq x_\star^\top\Big(\mathbf{M}_\star + \mathbf{M}_\star^\top\Big)x'_\star \ . \tag{3.2}$$

Hence, instead of looking for the optimal joint arm (which is NP-Hard), one can alternatively aim at seeking the allocation that, for any edge $(i, j) \in E$, constructs as many pairs $(x^{(i)}, x^{(j)}) = (x_\star, x'_\star)$ as possible. Assigning $x_\star$ to a subset of nodes and $x'_\star$ to the complementary is equivalent to cutting the graph into two pieces and creating two distinct sets of nodes V_1 and V_2 such that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. Thus, the described strategy boils down to finding a cut passing through the maximum number of edges.

This problem is known as the Max-Cut problem (see *e.g.*, [44, 84]), which is also NP-Hard. However, the extensive attention this problem has received allows us to use one of the many approximation algorithms (see, *e.g.*, Algorithm 4) which are guaranteed to yield a cut passing through at least a given fraction of the edges in the graph. Most of the guarantees for the approximation of the Max-Cut problem are stated with respect to the optimal Max-Cut solution, which is not exactly the guarantee we are looking for: we need a guarantee as a proportion of the total number of edges. We thus have to be careful on the algorithm we choose.

From Algorithm 4, one can have a guarantee on the proportion of cut edges with respect to the total number of edges $m = |E|$. We state this guarantee in the following Proposition.

Proposition 3.1. *Given a graph* $\mathcal{G} = (V, E)$*, Algorithm 4 returns a couple* (V_1, V_2) *such that*

$$\big|\{(i, j) \in E \,|\, (i \in V_1 \wedge j \in V_2) \vee (i \in V_2 \wedge j \in V_1)\}\big| \geq \frac{m}{2} \ .$$

[1] Those quantities are not equal since the matrix $\mathbf{M}_\star$ is not necessarily symmetric.

Algorithm 4: Approx-MAX-CUT [84]

Input : $\mathcal{G} = (V, E)$
Set $V_1 = \emptyset, V_2 = \emptyset$
for i *in* V **do**
  $n_1 = |\{(i,j) \in E \mid j \in V_1\}|$;
  $n_2 = |\{(i,j) \in E \mid j \in V_2\}|$;
  if $n_1 > n_2$ **then** $V_2 \leftarrow V_2 \cup \{i\}$ **else** $V_1 \leftarrow V_1 \cup \{i\}$;
end
return (V_1, V_2)

Proof. At each iteration, we take a node i that is neither in V_1 nor in V_2, count its neighbors already in V_1 and V_2 and save the results respectively in n_1 and n_2. For the sake of simplicity in the proof, we will denote them $n_1^{(i)}$ and $n_2^{(i)}$ to distinguish from one node to the other.

Since $n_1^{(i)}$ represents the number of neighbors of i already assigned to V_1, if the node i is added to V_2, $2 \times n_1^{(i)}$ edges would be cut (the factor 2 comes from the fact that between two nodes i and j, there are the edges (i,j) and (j,i)). Similarly, since $n_2^{(i)}$ represents the number of neighbors already assigned in V_2, if the node i is added to V_1, $2 \times n_2^{(i)}$ edges would be cut. In the algorithm, the node i is added to V_1 or V_2 such that we cut the most edges, hence by denoting m_i the number of additional cut edges implied by the assignment of node i in V_1 or V_2, we have:

$$m_i = \max\left(2n_1^{(i)}, 2n_2^{(i)}\right) \geq \frac{2n_1^{(i)} + 2n_2^{(i)}}{2} = n_1^{(i)} + n_2^{(i)} \ .$$

By summing for all the nodes in the graph :

$$\begin{aligned} \sum_{i=1}^{n} m_i &\geq \sum_{i=1}^{n} n_1^{(i)} + n_2^{(i)} \\ &= \frac{m}{2} \ . \end{aligned}$$

By definition $\sum_{i=1}^{n} m_i$ is the total number of edges that are cut which also means that

$$\sum_{i=1}^{n} m_i = |\{(i,j) \in E \mid (i \in V_1 \wedge j \in V_2) \vee (i \in V_2 \wedge j \in V_1)\}| \ .$$

□

Given this guarantee with respect to the total number of edges, it only remains to present the full strategy that is to allocate to the nodes in V_1 the arm $x_\star$ and to the nodes in V_2 the arm $x'_\star$. We give the strategy in Algorithm 5.

Algorithm 5: Approximation algorithm of our NP-Hard problem

Input : Graph $\mathcal{G} = (V, E)$, arm set $\mathcal{X}$, parameter matrix $\mathbf{M}_\star$
$(V_1, V_2) =$ Approx-MAX-CUT$(\mathcal{G})$;
Find $(x_\star, x'_\star) \in \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x'$;
for i *in* V_1 **do**
  $x_t^{(i)} = x_\star$; // Can be done in parallel
end
for i *in* V_2 **do**
  $x_t^{(i)} = x'_\star$; // Can be done in parallel
end
return $(x^{(1)}, \dots, x^{(n)})$

With this algorithm, given the returned allocation $(x^{(1)}, \dots, x^{(n)})$, for some edges $(i, j) \in E$, the associated allocated arms will be the optimal couples $(x_\star, x'_\star)$ or $(x'_\star, x_\star)$ and for other edges $(i, j) \in E$ the associated allocated arms will be the suboptimal and unwanted couples $(x_\star, x_\star)$ or $(x'_\star, x'_\star)$.

Before we state the guarantee of this algorithm with respect to the optimal global reward, let us introduce m_1 (respectively m_2) the number of edges that go from nodes in V_1 (respectively V_2) to nodes in V_1 as well (respectively V_2) and $m_{1\to2}$ (respectively $m_{2\to1}$) the number of edges that goes from nodes in V_1 (respectively V_2) to nodes in V_2 (respectively V_1). Notice that the total number of edges $m = m_{1\to2} + m_{2\to1} + m_1 + m_2$ and that by definition of the edge set E and using Proposition 3.1 we have $m_{1\to2} = m_{2\to1} \geq m/4$ and $m_1 + m_2 \leq m/2$.

Theorem 3.2. *Let us consider the graph $\mathcal{G} = (V, E)$, a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ and the matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ given as input to Algorithm 5. Let $(x_\star^{(1)}, \dots, x_\star^{(n)})$ be the optimal joint arm as defined in (3.1) and let $0 \leq \xi \leq 1$ be a problem-dependent parameter defined by*

$$\xi = \min_{x\in\mathcal{X}} \frac{x^\top \mathbf{M}_\star x}{\frac{1}{m}\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}} \ ,$$

and set $\alpha = \frac{1+\xi}{2}$. Then, the expected global reward $y = \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ associated with the allocation $\left(x^{(1)}, \dots, x^{(n)}\right) \in \mathcal{X}^n$ returned by Algorithm 5 verifies:

$$y \geq \alpha y_\star \ .$$

where $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$.
Finally, the complexity of the algorithm is in $\mathcal{O}(K^2 + n^2)$.

Proof. Given the allocation $(x^{(1)}, \dots, x^{(n)})$ return by Algorithm 5, the associated reward y can be written as

$$y = \underbrace{m_{1\to 2} \times x_\star^\top \mathbf{M}_\star x'_\star + m_{2\to 1} \times x'^\top_\star \mathbf{M}_\star x_\star}_{(a)} + \underbrace{m_1 \times x_\star^\top \mathbf{M}_\star x_\star + m_2 \times x'^\top_\star \mathbf{M}_\star x'_\star}_{(b)}$$

Let us analyse (a):

$$\begin{aligned}
(a) &= m_{1\to 2} \times x_\star^\top \mathbf{M}_\star x'_\star + m_{2\to 1} \times x'^\top_\star \mathbf{M}_\star x_\star \\
&= \frac{m_{1\to 2} + m_{2\to 1}}{2} \times x_\star^\top \Big(\mathbf{M}_\star + \mathbf{M}_\star^\top\Big) x'_\star \qquad \text{(because } m_{1\to 2} = m_{2\to 1}\text{)} \\
&= \frac{m_{1\to 2} + m_{2\to 1}}{m} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} x_\star^\top \Big(\mathbf{M}_\star + \mathbf{M}_\star^\top\Big) x'_\star \\
&\geq \frac{m_{1\to 2} + m_{2\to 1}}{m} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} x_\star^{(i)\top} \Big(\mathbf{M}_\star + \mathbf{M}_\star^\top\Big) x_\star^{(j)} \qquad (3.3) \\
&= \frac{m_{1\to 2} + m_{2\to 1}}{m} \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} \\
&= \frac{m_{1\to 2} + m_{2\to 1}}{m} y_\star \ ,
\end{aligned}$$

where (3.3) comes from Equation (3.2).
Now, we analyse (b) by using the definition of ξ where for any $x' \in \mathcal{X}$,

$$\begin{aligned}
x'^\top \mathbf{M}_\star x' &\geq \min_{x\in\mathcal{X}} x^\top \mathbf{M}_\star x \qquad (3.4) \\
&= \frac{1}{m}\xi \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} \ . \qquad (3.5)
\end{aligned}$$

Hence we have,

$$\begin{aligned}
(b) &\geq \frac{m_1}{m}\xi \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} + \frac{m_2}{m}\xi \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} \\
&= \frac{m_1 + m_2}{m}\xi y_\star \ .
\end{aligned}$$

Thus,

$$
\begin{aligned}
y &= (a) + (b) \\
&\geq \frac{m_{1\to 2} + m_{2\to 1}}{m} y_\star + \frac{m_1 + m_2}{m} \xi y_\star \\
&= \left(\frac{m_{1\to 2} + m_{2\to 1}}{m} + \frac{m_1 + m_2}{m} \xi \right) y_\star \\
&= \left(\frac{m_{1\to 2} + m_{2\to 1}}{m} + \left(1 - \frac{m_{1\to 2} + m_{2\to 1}}{m} \right) \xi \right) y_\star \\
&= \left(\frac{m_{1\to 2} + m_{2\to 1} + m\xi - (m_{1\to 2} + m_{2\to 1})\xi}{m} \right) y_\star \\
&= \left(\frac{m_{1\to 2} + m_{2\to 1}}{m} (1 - \xi) + \xi \right) y_\star \\
&\geq \left(\frac{1}{2} (1 - \xi) + \xi \right) y_\star \\
&= \frac{1 + \xi}{2} y_\star \ .
\end{aligned}
\tag{3.6}
$$

Moreover, the Approx-Max-Cut Algorithm has a complexity in $O(n^2)$, then we do K^2 estimations to find the best couple $(x_\star, x'_\star) \in \mathcal{X}^2$, and each round in the first and second for loop of the algorithm is in $O(1)$ and there are $|V_1| + |V_2| = n$ rounds.

Hence the complexity is in $O(K^2 + n^2)$. □

What is ξ and what value can we expect? This parameter measures what minimum gain with respect to the optimal reward one could get by allocating the unwanted and suboptimal couple of arms of the form (x, x) for two neighbors. For example, if there exists $x_0 \in \mathcal{X}$ such that $x_0^\top \mathbf{M}_\star x_0, = 0$, then $\xi = 0$ as well and we are in the worst case scenario where we can only have a guarantee on an α-approximation with $\alpha = 1/2$. In practice, having $\xi = 0$ is reached when given the couple $(x_\star, x'_\star) = \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x'$, this x_0 is either $x_\star$ or $x'_\star$. In other words, if the unwanted couples of arms $(x_\star, x_\star)$ and $(x'_\star, x'_\star)$ give low rewards, then the guarantee on the reward will be badly impacted. Hence we can wonder how to prevent this phenomenon. We answer this question in the next section by taking into account both the proportion of undesirable couple of arms and their potential rewards at the selection of the pair $(x_\star, x'_\star)$ in order to improve the performance of the algorithm both in practice and theoretically.

3.1.3 Improved algorithm using the graph structure

In this section, we want to capitalize on Algorithm 5 and its $\frac{1+\xi}{2}$-optimal solution to refine the allocation of the arms $x_\star$ and $x'_\star$ such that the obtained suboptimal rewards $x_\star^\top \mathbf{M}_\star x_\star$ and $x_\star'^\top \mathbf{M}_\star x'_\star$ penalize as less as possible the global reward.

Indeed, in the Algorithm 5, the choice of the couple $(x_\star, x'_\star)$ is only guided by the potential gain that could be obtained at the level of the cut edges (*i.e.*, that goes from a node in V_1 to a node in V_2 or vice versa). It does not take into account all the m_1 rewards of the form $x_\star^\top \mathbf{M}_\star x_\star$ and

the m_2 rewards of the form $x_\star'^\top \mathbf{M}_\star x_\star'$ that one gets when allocating $x_\star$ to the nodes in V_1 and $x_\star'$ to the nodes in V_2.

Here, the improvement we can make is to include them in the optimization problem and to weight the different rewards obtained through the graph using the proportions $m_{1\to2}$, $m_{2\to1}$, m_1 and m_2. By denoting $(\tilde{x}_\star, \tilde{x}_\star')$ the solution of the following optimization problem,

$$\max_{(x,x')\in\mathcal{X}^2} m_{1\to2} \cdot x^\top \mathbf{M}_\star x' + m_{2\to1} \cdot x'^\top \mathbf{M}_\star x + m_1 \cdot x^\top \mathbf{M}_\star x + m_2 \cdot x'^\top \mathbf{M}_\star x' \quad , \tag{3.7}$$

we are optimizing the total global reward that one would obtain when allocating only two arms $(x, x') \in \mathcal{X}^2$ in the graph. This strategy is described in Algorithm 6.

Algorithm 6: Improved approximation algorithm of our NP-Hard problem

Input : Graph $\mathcal{G} = (V, E)$, arm set $\mathcal{X}$, parameter matrix $\mathbf{M}_\star$
$(V_1, V_2) =$ Approx-MAX-CUT$(\mathcal{G})$;
$m_{1\to2} = |\{(i,j) \in E | i \in V_1 \wedge j \in V_2\}|$;
$m_{2\to1} = |\{(i,j) \in E | i \in V_2 \wedge j \in V_1\}|$;
$m_1 = |\{(i,j) \in E | i \in V_1 \wedge j \in V_1\}|$;
$m_2 = |\{(i,j) \in E | i \in V_2 \wedge j \in V_2\}|$;
Find $(\tilde{x}_\star, \tilde{x}_\star')$ solution of (3.7);
for *i in* V_1 **do**
  $x_t^{(i)} = \tilde{x}_\star$; // Can be done in parallel
end
for *i in* V_2 **do**
  $x_t^{(i)} = \tilde{x}_\star'$; // Can be done in parallel
end
return $(x^{(1)}, \dots, x^{(n)})$

To understand and analyse this new algorithm, let us define $\Delta \geq 0$ the global reward difference of allocating $(\tilde{x}_\star, \tilde{x}_\star')$ instead of $(x_\star, x_\star')$ as follows:

$$\begin{aligned}\Delta = m_{1\to2}\left(\tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}_\star' - x_\star^\top \mathbf{M}_\star x_\star'\right) + m_{2\to1}\left(\tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}_\star - x_\star'^\top \mathbf{M}_\star x_\star\right) \\ + m_1\left(\tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}_\star - x_\star^\top \mathbf{M}_\star x_\star\right) + m_2\left(\tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}_\star' - x_\star'^\top \mathbf{M}_\star x_\star'\right) \quad .\end{aligned}$$

The new guarantees that we get on the reward of the allocation obtained by Algorithm 6 are stated in the following theorem.

Theorem 3.3. *Let us consider the graph $\mathcal{G} = (V, E)$, a finite arm set $\mathcal{X} \subset \mathbb{R}^d$ and the matrix $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ given as input to Algorithm 6. Let $(x_\star^{(1)}, \dots, x_\star^{(n)})$ be the optimal joint arm as defined in (3.1) and let $0 \leq \xi \leq 1$ be defined as in Theorem 3.2. Let $0 \leq \epsilon \leq \frac{1}{2}$ be a problem dependent parameter that measures the relative gain of optimizing on the suboptimal rewards defined as:*

$$\epsilon = \frac{\Delta}{\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}} \ ,$$

and set $\alpha = \frac{1+\xi}{2} + \epsilon$. *Then, the global reward* $y = \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ *associated with the allocation* $\left(x^{(1)}, \ldots, x^{(n)}\right) \in \mathcal{X}^n$ *returned by Algorithm 6 verifies:*

$$y \geq \alpha y_\star \ .$$

where $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$.

Proof. The obtained reward y can be writen as follows:

$$\begin{aligned}
y &= m_{1\to 2} \tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}'_\star + m_{2\to 1} \cdot \tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}_\star + m_1 \cdot \tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}_\star + m_2 \cdot \tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}'_\star \\
&= m_{1\to 2} x_\star^\top \mathbf{M}_\star x'_\star + m_{2\to 1} \cdot x_\star'^\top \mathbf{M}_\star x_\star + m_1 \cdot x_\star^\top \mathbf{M}_\star x_\star + m_2 \cdot x_\star'^\top \mathbf{M}_\star x'_\star \\
&\quad + \Delta \\
&= m_{1\to 2} x_\star^\top \mathbf{M}_\star x'_\star + m_{2\to 1} \cdot x_\star'^\top \mathbf{M}_\star x_\star + m_1 \cdot x_\star^\top \mathbf{M}_\star x_\star + m_2 \cdot x_\star'^\top \mathbf{M}_\star x'_\star \\
&\quad + \epsilon \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(i)} \\
&= m_{1\to 2} x_\star^\top \mathbf{M}_\star x'_\star + m_{2\to 1} \cdot x_\star'^\top \mathbf{M}_\star x_\star + m_1 \cdot x_\star^\top \mathbf{M}_\star x_\star + m_2 \cdot x_\star'^\top \mathbf{M}_\star x'_\star + \epsilon y_\star \\
&= \frac{1+\xi}{2} y_\star + \epsilon y_\star \\
&= \left(\frac{1+\xi}{2} + \epsilon\right) y_\star \ .
\end{aligned}$$

□

What is ϵ *and what value can we expect?* This parameter measures the gain with respect to the optimal reward one could get by considering the undesirable couple of arms of the form (x, x). The value of ϵ is high when the rewards associated with the couples $(\tilde{x}_\star, \tilde{x}'_\star)$ and $(\tilde{x}'_\star, \tilde{x}_\star)$ are close to those of $(x_\star, x'_\star)$ and $(x'_\star, x_\star)$ respectively **and** when the rewards associated with the suboptimal couples $(\tilde{x}_\star, \tilde{x}_\star)$ and $(\tilde{x}'_\star, \tilde{x}'_\star)$ are much higher than those of $(x_\star, x_\star)$ and $(x'_\star, x'_\star)$ respectively. On the contrary, ϵ is low (and close to 0) if the suboptimal couples of arms $(x_\star, x_\star)$ and $(x'_\star, x'_\star)$ already give high rewards (or the highest among the other suboptimal couples of the form (x, x)), hence the central entity does not gain a lot by choosing another couple of arms than $(x_\star, x'_\star)$. Notice that $\alpha = \frac{1+\xi}{2} + \epsilon \leq 1$ by construction. In fact when $\xi = 1$ it means that the couple of arms of the form (x, x) gives the highest reward, hence $(x_\star, x'_\star) = (\tilde{x}_\star, \tilde{x}'_\star)$ which gives $\epsilon = 0$ and $\alpha = 1$.

Corollary 3.1. *Let us consider the same setting as in Theorem 3.3, the approximation ratio can be defined with the parameters* $m_{1\to 2}$, $m_{2\to 1}$, m_1 *and* m_2 *that depend on the graph and the approximation algorithm of the max-cut problem such that*

$$\alpha = \frac{m_{1\to 2} + m_{2\to 1}}{m} + \frac{m_1 + m_2}{m}\xi + \epsilon \ .$$

Proof. The proof follows exactly the one of Theorem 3.2 where we stop the reasoning at Equation 3.6 and then plug the result into the proof of Theorem 3.3 □

This corollary is useful to understand in practice the kind of guarantees we can have depending on the graph structure and the approximation algorithm we use to solve the Max-Cut problem. For instance, in the most favorable graphs, which are bipartite graphs (*i.e.*, graphs where all the m edges goes from nodes in V_1 to nodes in V_2 or vice versa), we have $m_{1\to 2} + m_{2\to 1} = m$ and $m_2 + m_1 = 0$. Also it implies that $(x_\star, x'_\star) = (\tilde{x}_\star, \tilde{x}'_\star)$, so $\epsilon = 0$ which gives $\alpha = 1$ and makes the Algorithm 10 find the optimal solution of the problem. What may also be of interest is to understand how α varies with respect to ξ, ϵ and the quantity m_1 and m_2 for graphs that are between a complete graph (that is the worst case scenario in terms of constraints) and a bipartite graph. We investigate experimentally this dependency in Section 3.2.

3.2 Numerical experiments: influence of the parameters on the solution

In this section, we give some insights on the problem-dependent parameters ξ and ϵ and the corresponding α. Let α_1 and α_2 be the α stated respectively in Theorem 3.2 and Corollary 3.1. In the first experiment, we show the dependence of α_1 and α_2 on the graph type and the chosen approximation algorithm for the max-cut problem with respect to ξ and ϵ. We also highlight the differences between the two parameters α_1 and α_2 as well as the significant improvement in guarantees that one can obtain using Algorithm 6 depending on the type of the graph. The results are presented in Table 3.1.

Table 3.1: Values of several parameters with respect to the type of graph. Experiments were performed on graphs of $n = 100$ nodes, and results for the random graph are averaged over 100 draws.

	Graph types				
	Complete	Random	Circle	Star	Matching
$\frac{m1+m2}{m}$	0.495	0.453	0.01	0	0
α_1	$0.5 + 0.5\xi$				
α_2	$0.505 + 0.495\xi + \epsilon$	$0.547 + 0.453\xi + \epsilon$	$0.99 + 0.01\xi + \epsilon$	1	1

One can notice that the complete graph seems to give the worst guarantee on the α-approximation with respect to ϵ and ξ. Thus, we conducted a second experiment where we consider the worst case scenario in terms of the graph type –*e.g.*, the complete graph– and where there are $n = 10$ agents. This second experiment studies the variation of ϵ and ξ with respect to the unknown

parameter matrix $\mathbf{M}_\star$. To design such experiment, we consider the arm-set $\mathcal{X}$ as the vectors $(e_1, \dots, e_d)$ of the canonical base in $\mathbb{R}^d$. We generate the matrix $\mathbf{M}_\star$ randomly in the following way: first, all elements of the matrix are drawn *i.i.d.*from a standard normal distribution, and then we take the absolute value of each of these elements to ensure that the matrix only contains positive numbers. The choice of the vectors of the canonical base as the arm-set allows us to modify the matrix $\mathbf{M}_\star$ and to illustrate the dependence on ξ and ϵ in a simple way. Consider the best couple $(i^\star, j^\star) = \arg\max_{(i,j)\in\{1,\dots,d\}^2} e_i^\top \mathbf{M}_\star e_j$, we want to see how the rewards of the suboptimal couples of arms $(e_{i^\star}, e_{i^\star})$ and $(e_{j^\star}, e_{j^\star})$ impact the values of ξ, ϵ and thus α. Notice that the reward associated with the couple of arms $(e_{i^\star}, e_{i^\star})$ (respectively $(e_{j^\star}, e_{j^\star})$) is $[\mathbf{M}_\star]_{i^\star i^\star}$ (respectively $[\mathbf{M}_\star]_{j^\star j^\star}$). Hence we define $0 \leq \zeta < 1$ and set $[\mathbf{M}_\star]_{i^\star i^\star} = [\mathbf{M}_\star]_{j^\star j^\star} = \zeta \times \frac{1}{2}([\mathbf{M}_\star]_{i^\star j^\star} + [\mathbf{M}_\star]_{j^\star i^\star})$. We study the variation of ξ, ϵ, α_1 and α_2 with respect to ζ. The results are presented in Figure 3.1. One can see that when the associated rewards of $(e_{i^\star}, e_{i^\star})$ and $(e_{j^\star}, e_{j^\star})$ are low (thus ξ is low and ϵ high), Algorithm 6 gives a much better guarantees than Algorithm 5 since it focuses on other arms than $e_{i^\star}$ and $e_{j^\star}$ that give a higher global reward. Moreover, even when the unwanted couples $(e_{i^\star}, e_{i^\star})$ and $(e_{j^\star}, e_{j^\star})$ give high rewards, the guarantees on the regret of Algorithm 10 are still stronger because it takes into consideration the quantities m_1 and m_2 of the constructed suboptimal couple of arms $(e_{i^\star}, e_{i^\star})$ and $(e_{j^\star}, e_{j^\star})$.

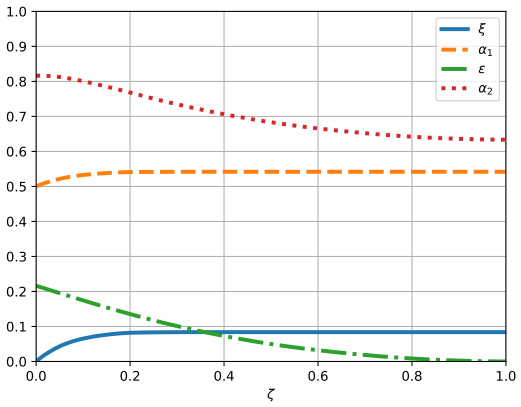


Figure 3.1: Variation of ϵ, ξ, α_1 and α_2 with respect to the parameter ζ. The closer ζ is to 0 the lower the reward of the unwanted couples $(e_{i^\star}, e_{i^\star})$ and $(e_{j^\star}, e_{j^\star})$, the closer ζ is to 1 the higher the rewards of the unwanted couples. The dimension d of the arm-set is 10 (which gives linear reward with unknown parameter $\theta_\star$ of dimension 100). The plotted curve represents the average value of the parameters over 100 different matrices $\mathbf{M}_\star$ initiated randomly with positive values.

3.3 Conclusion and perspectives

In this chapter, we showed that the underlying optimization problem is NP-hard and that one has to rely on approximation algorithms. To that matter, we designed a first α-approximation algorithm based on the max-cut problem, with $\alpha \geq 1/2$. We showed that by exploiting the graph structure and the typology of the problem, one can both improve the performance in practice and have a better theoretical guarantee on α. The method presented in this chapter can be extended

in many ways, especially in the choice of the max-cut approximation algorithm. For instance, one can consider cutting the graph into 3 or more pieces, which is equivalent to approximating the problem of a *Max-k-Cut* [43] with $k \geq 3$. With the knowledge of such a partition of nodes $V_1, \ldots, V_k$, one may want to look for a k-tuple of arms maximizing the optimistic allocated reward rather than a pair, therefore introducing an elegant tradeoff between the optimality of the solution and the computational complexity of the arms allocation.

4 Best-arm identification in graphical bilinear bandits

Contents

In this chapter, we assume that we do not know the parameter matrix $\mathbf{M}_\star$ and as described in the problem setting in Section 1.2, a central entity faces a graphical bilinear bandits problem where at each round it chooses an arm for each node of the graph and observes a bilinear reward for each edge of the graph. In this chapter, we will focus on the best-arm identification objective that reads as follows:

Objective: *Find, within a minimum number of rounds, the joint arm* $(x_\star^{(1)}, \dots, x_\star^{(n)})$ *such that the expected global reward* $\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$ *is maximized, where* $\mathbf{M}_\star$ *is unknown to the learner.*

We follow the notations we established in section 1.2.1.

4.1 Preliminaries

4.1.1 A two-stage algorithm template

For simplicity, we consider for now that the unknown parameter $\mathbf{M}_\star$ is symmetric, which greatly simplifies the reasoning, and we will relax this assumption in Section 4.2.3. In Chapter 3, we designed polynomial-time algorithms that allow us to compute an α-approximation solution to the NP-Hard problem of finding the best joint arm given $\mathbf{M}_\star$. Notice that in the α-approximation Algorithm 5, $\mathbf{M}_\star$ is only used to identify the best pair $(x_\star, x'_\star)$ as follows:

$$\begin{aligned}(x_\star, x'_\star) &= \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x' \\ &= \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \mathbf{M}_\star x' \ . \end{aligned} \tag{4.1}$$

Thus, using an estimate $\hat{\mathbf{M}}$ of $\mathbf{M}_\star$ having the following property:

$$\underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \hat{\mathbf{M}} x' = \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \mathbf{M}_\star x' = (x_\star, x'_\star) \ , \tag{4.2}$$

allows us to identify the pair $(x_\star, x'_\star)$, and thus gives us the same guarantees as the ones presented in Theorem 3.2. We thus address the problem of computing $\hat{\mathbf{M}}$ such that, in a minimum number of rounds, with high probability, we are able to identify the pair $(x_\star, x'_\star)$ and apply the Algorithm 5.

Our general algorithm can thus be thought of as a two-part algorithm where, in the first instance, the central entity applies a pure exploration algorithm and draws the arms of the nodes at each round to compute the best possible estimate $\hat{\mathbf{M}}$ (in a certain sense), and then, in the second instance, uses $\hat{\mathbf{M}}$ to apply the α-approximation-Algorithm 5 described in chapter 3. We describe the general procedure in Algorithm 7.

Algorithm 7: General framework for BAI in GBB : a two-stage algorithm

Input : graph $\mathcal{G} = (V, E)$, arm set $\mathcal{X}$
$\hat{\mathbf{M}}$ = Pure-Exploration-Algorithm($\mathcal{G}$,$\mathcal{X}$); // This chapter
$\left(x^{(1)}, \ldots, x^{(n)}\right)$ = α-Approximation-Algorithm($\mathcal{G}$, $\mathcal{X}$, $\hat{\mathbf{M}}$); // Done in chapter 3
return $\left(x^{(1)}, \ldots, x^{(n)}\right)$

4.1.2 Stopping condition

Notice that the optimization problem (4.1) is equivalent to the following optimization problem

$$(x_\star, x'_\star) = \underset{(x,x')\in\mathcal{X}}{\arg\max} \left\langle \operatorname{vec}\left(x x'^\top\right), \operatorname{vec}\left(\mathbf{M}_\star\right) \right\rangle \ .$$

Let us simplify the notations and denote $\theta_\star \triangleq \text{vec}(\mathbf{M}_\star)$ the vectorized version of the unknown matrix $\mathbf{M}_\star$. Let us also use the notation $z_{xx'} \triangleq \text{vec}\left(xx'^\top\right)$, and define $\mathcal{Z} = \{z_{xx'} | (x, x') \in \mathcal{X}^2\}$ the set containing such vectors. Then looking for the couple $(x_\star, x'_\star)$ is the same as looking for the vector $z_\star \in \mathcal{Z}$ where

$$z_\star = \arg\max_{z \in \mathcal{Z}} \langle z, \theta_\star \rangle \ .$$

In other words, we want to find an arm $z \in \mathcal{Z}$, such that for all $z' \in \mathcal{Z}$, $(z - z')^\top \theta_\star \geq 0$. However, one does not have access to $\theta_\star$, so we have to use its empirical estimate.

Hence at each round t, the central entity can choose for each couple of neighbors (i, j) an arm $z \in \mathcal{Z}$ and get a noisy linear reward of the form $\langle z, \theta_\star \rangle + \eta$ where η is a σ-subgaussian random variable, that can be used to compute an estimate $\hat{\theta}_t$.

For more clarity, we refer to any $x \in \mathcal{X}$ as a *node-arm* and any $z \in \mathcal{Z}$ will be referred as an *edge-arm*. If $x_t^{(i)} \in \mathcal{X}$ represents the node-arm allocated to the node $i \in V$ at time t, for each edge $(i, j) \in E$ we will denote the associated edge-arm by $z_t^{(i,j)} \triangleq \text{vec}\left(x_t^{(i)} x_t^{(j)\top}\right) \in \mathcal{Z}$.

Assumption 4.1 (Bounded edge-arm norm)**.** *We consider that there exists $L > 0$, for all edge-arm $z \in \mathcal{Z}$, such that $\|z\|_2^2 \leq L$.*

Assumption 4.2 (Positive rewards)**.** *We consider that for any $z \in \mathcal{Z}$, the associated expected reward $\langle z, \theta_\star \rangle$ is such that $\langle z, \theta_\star \rangle \geq 0$*

Assumption 4.3 (Spanning the action space)**.** *We consider that $\mathcal{X}$ spans $\mathbb{R}^d$.*

The goal here is to define the optimal sequence $(z_1, \ldots, z_{mt}) \in \mathcal{Z}^{mt}$ that should be pulled in the first t rounds so that (4.2) is reached as soon as possible. A natural approach is to rely on classical strategies developed for best arm identification in linear bandits. We define $(y_1, \ldots, y_{mt})$ the corresponding noisy rewards of the sequence $(z_1, \ldots, z_{mt})$. We assume that the noise terms in the rewards are *i.i.d.*, following a σ-sub-Gaussian distribution. Let $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \in \mathbb{R}^{d^2}$ be the solution of the ordinary least squares problem with $\mathbf{A}_t = \sum_{i=1}^{mt} z_i z_i^\top \in \mathbb{R}^{d^2 \times d^2}$ and $b_t = \sum_{i=1}^{mt} z_i y_i \in \mathbb{R}^{d^2}$. We first recall the Proposition 2.2 with the notation of our problem.

Proposition 4.1 (Proposition 1 in [90])**.** *For every fixed sequence $(z_1, \ldots, z_{mt}) \in \mathcal{Z}^{mt}$, with probability $1 - \delta$, for all $t > 0$ and for all $z \in \mathcal{Z}$, we have*

$$\left| z^\top \theta_\star - z^\top \hat{\theta}_t \right| \leq 2\sigma\sqrt{2} \|z\|_{\mathbf{A}_t^{-1}} \sqrt{\log\left(\frac{6m^2 t^2 K^2}{\delta \pi}\right)} \ .$$

Following the steps of [90] and the ones developed in Section 2.2.2, we can show that if there exists $z \in \mathcal{Z}$ such that for all $z' \in \mathcal{Z}$ the following holds:

$$\|z - z'\|_{\mathbf{A}_t^{-1}} \sqrt{8\sigma^2 \log\left(\frac{6m^2 t^2 K^4}{\delta \pi^2}\right)} \leq \hat{\Delta}_t(z, z') \ , \tag{4.3}$$

where $\hat{\Delta}_t(z, z') = (z - z')^\top \hat{\theta}_t$ is the empirical gap between z and z', then with probability at least $1-\delta$, the OLS estimate $\hat{\theta}_t$ leads to the best edge-arm $z_\star$, which means that $\arg\max_{z\in\mathcal{Z}} \langle z, \hat{\theta}_t \rangle = \arg\max_{z\in\mathcal{Z}} \langle z, \theta_\star \rangle$. Therefore, when the Equation (4.3) is true, the learner can stop pulling arms, we call it the *stopping condition*.

As mentioned in [90], by noticing that $\max_{(z,z')\in\mathcal{Z}^2} \|z - z'\|_{\mathbf{A}_t^{-1}} \leq 2 \max_{z\in\mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$, an admissible strategy is to pull edge-arms minimizing $\max_{z\in\mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$ in order to satisfy the stopping condition as soon as possible.

4.1.3 A Constrained G-Allocation

Given the stopping condition (4.3) derived in the previous section, one wants to find the sequence of edge-arms $\mathbf{z}^\star_{mt} = (z^\star_1, \ldots, z^\star_{mt})$ such that:

$$\mathbf{z}^\star_{mt} \in \underset{(z_1,\ldots,z_{mt})\in\mathcal{Z}^{mt}}{\arg\min} \max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i z_i^\top \right)^{-1} z' \ . \tag{G-opt-$\mathcal{Z}$}$$

This is known as *G-allocation* (see *e.g.*, [76, 90]) and is NP-hard to compute ([32, 104]). One way to find an approximate solution is to rely on a convex relaxation of the optimization problem (G-opt-$\mathcal{Z}$) and first compute a real-valued allocation $\Gamma^\star \in \mathcal{S}_\mathcal{Z}$ such that

$$\Gamma^\star \in \underset{\Gamma\in\mathcal{S}_\mathcal{Z}}{\arg\min} \max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{z\in\mathcal{Z}} \Gamma_z z z^\top \right)^{-1} z' \ . \tag{G-relaxed-$\mathcal{Z}$}$$

One could either use random sampling to draw edge-arms as *i.i.d.* samples from the $\Gamma^\star$ distribution or rounding procedures to efficiently convert each $\Gamma^\star_z$ into an integer. However, these methods do not take into account the graphical structure of the problem. Indeed, at a given round, the chosen edge-arms may result in two different assignments for the same node, we call this phenomenon a *collision*. In Figure 4.1, we characterise a collision that occurs when the central entity allocates respectively two edge-arms z and z' to two edges that share the same node.

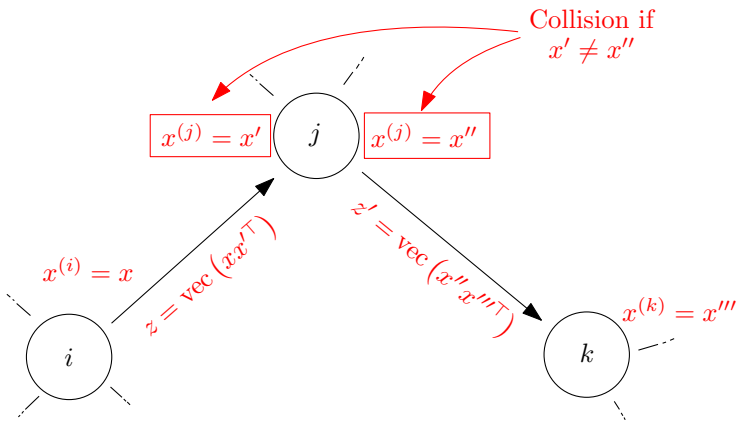


Figure 4.1: Collision when allocating directly edge-arms to the edges

Therefore, random sampling or rounding procedures cannot be straightforwardly used to select edge-arms in $\mathcal{Z}$. Nevertheless, (G-relaxed-$\mathcal{Z}$) still gives valuable information on the number of times, in proportion, each edge-arm $z \in \mathcal{Z}$ must be allocated to the graph. In the next section, we present an algorithm satisfying both the proportion requirements and the graphical constraints.

4.2 Algorithm and guarantees

4.2.1 Random Allocation over the Nodes

Our algorithm is based on a randomized method directly allocating node-arms to the nodes and thus avoiding the difficult task of choosing edge-arms and trying to allocate them to the graph while ensuring that every node has an unique assignment. The validity of this random allocation is based on Theorem 4.1 below showing that one can draw node-arms in $\mathcal{X}$ and allocate them to the graph such that the associated edge-arms follow the probability distribution $\Gamma^\star$ solution of (G-relaxed-$\mathcal{Z}$).

Theorem 4.1. *Let $\gamma^\star$ be a solution of the following optimization problem:*

$$\min_{\gamma \in \mathcal{S}_\mathcal{X}} \max_{x' \in \mathcal{X}} {x'}^\top \left(\sum_{x \in \mathcal{X}} \gamma_x x x^\top \right)^{-1} x' \ . \qquad \text{(G-relaxed-}\mathcal{X}\text{)}$$

Let $\Gamma^\star \in \mathcal{S}_\mathcal{Z}$ be defined for all $z = \operatorname{vec}\left(x{x'}^\top\right) \in \mathcal{Z}$ by $\Gamma^\star_z = \gamma^\star_x \gamma^\star_{x'}$. Then, $\Gamma^\star$ is a solution of (G-relaxed-$\mathcal{Z}$).

To prove Theorem 4.1, we first state a useful lemma. For any finite set $X \subset \mathbb{R}^d$ and $\gamma \in \mathcal{S}_X$, let $\Sigma_X(\gamma) = \sum_{x \in X} \gamma_x x x^\top$. We define the function $h_X : \mathcal{S}_X \to \mathbb{R} \cup \{+\infty\}$ as follows: for any $\gamma \in \mathcal{S}_X$,

$$h_X(\gamma) = \begin{cases} \max_{x' \in X} {x'}^\top \Sigma_X(\gamma)^{-1} x' & \text{if } \Sigma_X(\gamma) \text{ is invertible} \\ +\infty & \text{otherwise} \ . \end{cases}$$

Lemma 4.1. *Let $\mathcal{X} \subset \mathbb{R}^d$ be a finite set spanning $\mathbb{R}^d$ and let $\mathcal{Z} = \{\operatorname{vec}\left(x{x'}^\top\right) \mid (x, x') \in \mathcal{X}^2\}$. If $\gamma^\star \in \mathcal{S}_\mathcal{X}$ is a minimizer of $h_\mathcal{X}$, then $\gamma^\star$ is a solution of*

$$\min_{\gamma \in \mathcal{S}_\mathcal{X}} \max_{z \in \mathcal{Z}} z^\top \left(\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \gamma_x \gamma_{x'} \operatorname{vec}\left(x{x'}^\top\right) \operatorname{vec}\left(x{x'}^\top\right)^\top \right)^{-1} z \ .$$

Proof. First, let us notice that, for any $\mathcal{X} \subset \mathbb{R}^d$, one has $h_\mathcal{X} \geq 0$. Thus, $\gamma^\star$ is also a minimizer of $h^2_\mathcal{X}$. In addition, $\mathcal{X}$ is spanning $\mathbb{R}^d$ so $h_\mathcal{X}(\gamma^\star) < +\infty$. Developing $h_\mathcal{X}(\gamma^\star)^2$ yields:

$$h_\mathcal{X}(\gamma^\star) \times h_\mathcal{X}(\gamma^\star) = \left(\max_{x \in \mathcal{X}} \ x^\top \Sigma_\mathcal{X}(\gamma^\star)^{-1} x \right) \times \left(\max_{x \in \mathcal{X}} \ x^\top \Sigma_\mathcal{X}(\gamma^\star)^{-1} x \right)$$

$$
\begin{aligned}
&= \max_{x\in\mathcal{X}} \max_{x'\in\mathcal{X}} \; x^\top \Sigma_{\mathcal{X}}(\gamma^\star)^{-1} x x'^\top \Sigma_{\mathcal{X}}(\gamma^\star)^{-1} x' \\
&= \max_{x\in\mathcal{X}} \max_{x'\in\mathcal{X}} \; \operatorname{vec}\left(xx'^\top\right)^\top \operatorname{vec}\left(\Sigma_{\mathcal{X}}(\gamma^\star)^{-1} x x'^\top \Sigma_{\mathcal{X}}(\gamma^\star)^{-1}\right) \\
&= \max_{x\in\mathcal{X}} \max_{x'\in\mathcal{X}} \; \operatorname{vec}\left(xx'^\top\right)^\top \left(\Sigma_{\mathcal{X}}(\gamma^\star)^{-1} \otimes \Sigma_{\mathcal{X}}(\gamma^\star)^{-1}\right) \operatorname{vec}\left(xx'^\top\right) \\
&= \max_{z\in\mathcal{Z}} \; z^\top \left(\Sigma_{\mathcal{X}}(\gamma^\star)^{-1} \otimes \Sigma_{\mathcal{X}}(\gamma^\star)^{-1}\right) z \;\;,
\end{aligned}
$$

where $\otimes$ denotes the Kronecker product. We can now focus on the central term:

$$
\begin{aligned}
\Sigma_{\mathcal{X}}(\gamma^\star)^{-1} \otimes \Sigma_{\mathcal{X}}(\gamma^\star)^{-1} &= \left(\sum_{x\in\mathcal{X}} \gamma^\star_x x x^\top\right)^{-1} \otimes \left(\sum_{x\in\mathcal{X}} \gamma^\star_x x x^\top\right)^{-1} \\
&= \left(\sum_{x\in\mathcal{X}} \gamma^\star_x x x^\top \otimes \sum_{x\in\mathcal{X}} \gamma^\star_x x x^\top\right)^{-1} \\
&= \left(\sum_{x\in\mathcal{X}} \sum_{x'\in\mathcal{X}} \gamma^\star_x \gamma^\star_{x'} \left(xx^\top \otimes x'x'^\top\right)\right)^{-1} \\
&= \left(\sum_{x\in\mathcal{X}} \sum_{x'\in\mathcal{X}} \gamma^\star_x \gamma^\star_{x'} \operatorname{vec}\left(xx'^\top\right) \operatorname{vec}\left(xx'^\top\right)^\top\right)^{-1} \;\;,
\end{aligned}
$$

and the result holds. □

Proof of Theoreme 4.1. From [53], we know that $\min_{\Gamma\in\mathcal{S}_\mathcal{Z}} h_\mathcal{Z}(\Gamma) = d^2$ and $\min_{\gamma\in\mathcal{S}_\mathcal{X}} h_\mathcal{X}(\gamma) = d$. Then, using Lemma 4.1, one has

$$
\begin{aligned}
d^2 &= h_\mathcal{X}(\gamma^\star) \times h_\mathcal{X}(\gamma^\star) \\
&= \max_{z'\in\mathcal{Z}} \; z'^\top \left(\sum_{x\in\mathcal{X}} \sum_{x'\in\mathcal{X}} \gamma^\star_x \gamma^\star_{x'} \operatorname{vec}\left(xx'^\top\right) \operatorname{vec}\left(xx'^\top\right)^\top\right)^{-1} z' \\
&= \max_{z'\in\mathcal{Z}} \; z'^\top \left(\sum_{z\in\mathcal{Z}} \Gamma^\star_z z z^\top\right)^{-1} z' \;\;.
\end{aligned}
$$

This result implies that $h_\mathcal{Z}(\Gamma^\star) = d^2$. Since $\min_{\Gamma\in\mathcal{S}_\mathcal{Z}} h_\mathcal{Z}(\Gamma) = d^2$, $\Gamma^\star$ is a minimizer of $h_\mathcal{Z}$.

□

This theorem implies that, at each round $t > 0$ and each node $i \in V$, if $x_t^{(i)}$ is drawn from $\gamma^\star$, then for all pairs of neighbors $(i, j) \in E$ the probability distribution of the associated edge-arms $z_t^{(i,j)}$ follows $\Gamma^\star$. Moreover, as $\gamma^\star$ is a distribution over the node-arm set $\mathcal{X}$, $\Gamma^\star$ is a joint (product) probability distribution on $\mathcal{X}^2$ with marginal $\gamma^\star$.

On the computation of $\gamma^\star$. Let us first state the following proposition:

Proposition 4.2. *Let $d > 0$, for any set $X \subset \mathbb{R}^d$, h_X is convex.*

Proof. Let $(\gamma, \gamma') \in \mathcal{S}_X^2$ be two distributions in $\mathcal{S}_X$. If either $\Sigma_X(\gamma)$ or $\Sigma_X(\gamma')$ are not invertible, then for any $t \in [0,1]$ one has

$$h_X(t\gamma + (1-t)\gamma') \leq t h_X(\gamma) + (1-t) h_X(\gamma') = +\infty \ .$$

Otherwise, for $t \in [0,1]$, we define the positive definite matrix $\mathbf{Z}(t) \in \mathbb{R}^{d\times d}$ as follows:

$$\mathbf{Z}(t) = t\Sigma_X(\gamma) + (1-t)\Sigma_X(\gamma') \ .$$

Simple linear algebra [74] yields

$$\frac{\partial \mathbf{Z}(t)^{-1}}{\partial t} = \mathbf{Z}(t)^{-1}\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1} \ .$$

Using this result and the fact that $\partial^2\mathbf{Z}(t)/\partial t^2 = 0$, we obtain

$$\frac{\partial^2 \mathbf{Z}(t)^{-1}}{\partial t^2} = 2\mathbf{Z}(t)^{-1}\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1} \ .$$

Therefore, for any $x \in X$,

$$\begin{aligned}\frac{\partial^2 x^\top \mathbf{Z}(t)^{-1} x}{\partial t^2} &= 2x^\top \mathbf{Z}(t)^{-1}\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x \\ &= 2\left(\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x\right)^\top \mathbf{Z}(t)^{-1}\left(\frac{\partial \mathbf{Z}(t)}{\partial t}\mathbf{Z}(t)^{-1}x\right) \\ &\geq 0 \ ,\end{aligned}$$

which shows convexity for any fixed $x \in X$. The final results yields from the fact that h_X is a maximum over convex functions.

□

Although we face a min-max optimization problem and given the convexity of $h_{\mathcal{X}}$, we apply the Frank-Wolfe algorithm [41] to compute the solution $\gamma^\star$ of (G-relaxed-$\mathcal{X}$), as it is more suited to optimization tasks on the simplex than projected gradient descent. The convergence of the algorithm has been proven in [34].

Given the characterization in Theorem 4.1 and our objective to verify the stopping condition in (4.3), we present our sampling procedure in Algorithm 8. We also note that at each round the sampling of the node-arms can be done in parallel.

This sampling procedure implies that each edge-arm follows the optimal distribution $\Gamma^\star$. However, if we take the number of times each $z \in \mathcal{Z}$ appears in the m pulled edge-arms of a given round, we might notice that the observed proportion is not close to $\Gamma_z^\star$, regardless of the size of m. This is due to the fact that the m edge-arms are not independent because of the graph structure

Algorithm 8: Pure-Exploration-Algorithm : Randomized G-Allocation for GBB

Input : graph $\mathcal{G} = (V, E)$, arm set $\mathcal{X}$
Set $A_0 = I$; $b_0 = 0$; $t = 1$;
Apply the Frank-Wolfe algorithm to find $\gamma^\star$ solution of (G-relaxed-$\mathcal{X}$).
while *stopping condition* (4.3) *is not verified* **do**
  // Sampling the node-arms
  Draw $x_t^{(1)}, \dots, x_t^{(n)} \overset{\text{iid}}{\sim} \gamma^\star$ and obtain for all (i,j) in E the rewards $y_t^{(i,j)}$;
  // Estimating $\hat{\theta}_t$ with the associated edge-arms
  $\mathbf{A}_t = \mathbf{A}_{t-1} + \sum_{(i,j)\in E} z_t^{(i,j)} z_t^{(i,j)\top}$;
  $b_t = b_{t-1} + \sum_{(i,j)\in E} z_t^{(i,j)} y_t^{(i,j)}$;
  $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t$
  $t \leftarrow t + 1$;
end
return $\hat{\theta}_t$

(*cf.* Section 4.3.1). Conversely, since each group of m edge-arms are independent from one round to another, the proportion of each $z \in \mathcal{Z}$ observed among the mt pulled edge-arms throughout t rounds is close to $\Gamma_z^\star$.

One may wonder whether deterministic rounding procedures could be used instead of random sampling on $\gamma^\star$, as it is done in many standard linear bandit algorithms [39, 90]. Applying rounding procedure on $\gamma^\star$ gives the number of times each node-arm $x \in \mathcal{X}$ should be allocated to the graph. However, it does not provide the actual allocations that the learner must choose over the t rounds to optimally pull the associated edge-arms (*i.e.,* pull edge-arms following $\Gamma^\star$). Thus, although rounding procedures give a more precise number of times each node-arm should be pulled, the problem of allocating them to the graph remains open, whereas by concentration of the measure, randomized sampling methods imply that the associated edge-arms follow the optimal probability distribution $\Gamma^\star$. In this thesis, we present a simple and standard randomized G-allocation strategy, but other more elaborated methods could be considered, as long as they include the necessary randomness.

On the choice of the G-allocation problem. We have considered the G-allocation optimization problem (G-opt-$\mathcal{Z}$), however, one could want to directly minimize $\max_{(z,z')\in\mathcal{Z}^2} \|z - z'\|_{\mathbf{A}_t^{-1}}$, known as the XY-allocation [39, 90]. Hence, one may want to construct edge-arms that follow the distribution $\Gamma_{\mathrm{XY}}^\star$ solution of the relaxed XY-allocation problem:

$$\min_{\Gamma\in\mathcal{S}_\mathcal{Z}} \max_{(z',z'')\in\mathcal{Z}^2} \left(z' - z''\right)^\top \left(\sum_{z\in\mathcal{Z}} \Gamma_z z z^\top\right)^{-1} \left(z' - z''\right) .$$

Although efficient in the linear case, this approach outputs a distribution $\Gamma_{\mathrm{XY}}^\star$ which is not a joint probability distribution of two independent random variables, and so cannot be decomposed as the product of its marginals. Hence, there is no algorithm that allocates *identically* and *indepen-*

dently the nodes of the graph to create edge-arms following $\Gamma^\star_{\mathrm{XY}}$. Thus, we will rather deal with the upper bound given by the G-allocation as it allows sampling over the nodes.

Static design versus adaptive design. Adaptive designs as proposed for example in [90] and [39] provide a strong improvement over static designs in the case of linear bandits. In our particular setting however, it is crucial to be able to adapt the edge-arms sampling rule to the node-arms, which is possible thanks to Theorem 4.1. This result requires a set of edge-arms $\mathcal{Z}$ expressed as a product of node-arms set $\mathcal{X}$. Extending the adaptive design of [39] to our setting would eliminate edge-arms from $\mathcal{Z}$ at each phase, without trivial guarantees that the newly obtained edge-arms set $\mathcal{Z}' \subset \mathcal{Z}$ could still be derived from another node-arms set $\mathcal{X}' \subset \mathcal{X}$. An adaptive approach is definitely a natural and promising extension of our method, and is left for future work.

4.2.2 Convergence Analysis

We now prove the validity of the random sampling procedure detailed in Algorithm 8 by controlling the quality of the approximation $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ with respect to the optimum of the G-allocation optimization problem $\max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top}\right)^{-1} z'$ described in (G-opt-$\mathcal{Z}$). As is usually done in the optimal design literature (see *e.g.*, [76, 83, 90]) we bound the relative error β_t:

$$\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z \leq (1+\beta_t) \max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top}\right)^{-1} z' \ .$$

Our analysis relies on several results from matrix concentration theory. One may refer for instance to [96] and references therein for an extended introduction on that matter. We first introduce a few additional notations.

Let $f_{\mathcal{Z}}$ be the function such that for any non-singular matrix $\mathbf{Q} \in \mathbb{R}^{d^2\times d^2}$, $f_{\mathcal{Z}}(\mathbf{Q}) = \max_{z\in\mathcal{Z}} z^\top \mathbf{Q}^{-1} z$ and for any distribution $\Gamma \in \mathcal{S}_{\mathcal{Z}}$ we recall that $\Sigma_{\mathcal{Z}}(\Gamma) \triangleq \sum_{z\in\mathcal{Z}} \Gamma_z z z^\top$ is the associated covariance matrix. Finally let $\mathbf{A}_t^\star = \sum_{i=1}^{mt} z_i^\star z_i^{\star\top}$ be the G-optimal design matrix constructed during t rounds.

Theorem 4.2. *Let $\Gamma^\star$ be a solution of the optimization problem* (G-relaxed-$\mathcal{Z}$)*. Let $0 < \delta \leq 1$ and let t_0 be such that*

$$t_0 = 2Ld^2 \log\big(2d^2/\delta\big)/\lambda_{\min} \ ,$$

where $\lambda_{\min}$ is the smallest eigenvalue of the covariance matrix $\frac{1}{K^2}\sum_{z\in\mathcal{Z}} zz^\top$. Then, at each round $t \geq t_0$ with probability at least $1-\delta$, the randomized G-allocation strategy for graphical bilinear bandits in Algorithm 8 produces a matrix $\mathbf{A}_t$ such that:

$$f_{\mathcal{Z}}(\mathbf{A}_t) \leq (1+\beta_t) f_{\mathcal{Z}}(\mathbf{A}_t^\star) \ ,$$

where

$$\beta_t = \frac{Ld^2}{m\lambda_{\min}^2}\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{\sqrt{t}}\right) \ ,$$

and $v \triangleq \left\|\mathbb{E}\left[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2\right]\right\|$.

To prove this confidence bound, we need the two following propositions.

Proposition 4.3 ([96], Chapter 5 and 6)**.** *Let $\mathbf{Z}_1, \dots, \mathbf{Z}_t$ be i.i.d. positive semi-definite random matrices in $\mathbb{R}^{d^2 \times d^2}$, such that there exists $L > 0$ verifying $\mathbf{0} \preceq \mathbf{Z}_1 \preceq mL\mathbf{I}$. Let $\mathbf{A}_t$ be defined as $\mathbf{A}_t \triangleq \sum_{s=1}^{t} \mathbf{Z}_s$. Then, for any $0 < \varepsilon < 1$, one can lowerbound $\lambda_{\min}(\mathbf{A}_t)$, the minimum eigenvalue of $\mathbf{A}_t$, as follows:*

$$\mathbb{P}(\lambda_{\min}(\mathbf{A}_t) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{A}_t)) \leq d^2 e^{-\frac{t\varepsilon^2 \lambda_{\min}(\mathbb{E}\mathbf{Z}_1)}{2mL}} \ .$$

If in addition, there exists some $v > 0$, such that $\|\mathbb{E}\big[(\mathbf{Z}_1 - \mathbb{E}\mathbf{Z}_1)^2\big]\| \leq v$, then for any $u > 0$, one has

$$\mathbb{P}(\|\mathbf{S}_t\| \geq u) \leq 2d^2 e^{-\frac{u^2}{2mLu/3 + 2tv}} \ ,$$

From the second inequality, [78] derived a slightly different inequality that we use here :

Proposition 4.4 ([78], Appendix A.3)**.** *Let $\mathbf{Z}_1, \dots, \mathbf{Z}_t$ be t* i.i.d. *random symmetric matrices in $\mathbb{R}^{d^2 \times d^2}$ such that there exists $L > 0$ such that $\|\mathbf{Z}_1\| \leq mL$, almost surely. Let $\mathbf{A}_t \triangleq \sum_{i=1}^{t} \mathbf{Z}_i$. Then, for any $u > 0$, one has:*

$$\mathbb{P}\left(\|\mathbf{A}_t - \mathbb{E}\mathbf{A}_t\| \geq \sqrt{2tvu} + \frac{mLu}{3} \right) \leq d^2 e^{-u} \ .$$

where $v \triangleq \big\|\mathbb{E}\big[(\mathbf{Z}_1 - \mathbb{E}\mathbf{Z}_1)^2\big]\big\|$.

Finally, to prove our main theorem, we need the following lemma.

Lemma 4.2. *One has $\big\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\big\| \leq \frac{d^2}{\lambda_{\min}}$, where $\lambda_{\min}$ is the smallest eigenvalue of the covariance matrix $\frac{1}{K^2}\sum_{z \in \mathcal{Z}} z^\top z$.*

Proof. Define $\mathcal{B} = \Big\{ z \in \mathbb{R}^{d^2} : \|z\| = 1 \Big\}$. First, for any semi-definite matrix $\mathbf{A} \in \mathbb{R}^{d^2 \times d^2}$, we have $\|\mathbf{A}\| = \max_{z \in \mathcal{B}} \ z^\top \mathbf{A} z$. Because $\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}$ is positive definite and symmetric, and by Rayleigh-Ritz theorem,

$$\big\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\big\| = \max_{z \in \mathcal{B}} \ \frac{z^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z}{z^\top z} = \max_{z \in \mathcal{B}} \ z^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z \ .$$

Let $\mathbf{Z} \in \mathbb{R}^{K^2 \times d^2}$ be the matrix whose rows are vectors of $\mathcal{Z}$ in an arbitrary order. Notice that $\mathcal{Z}$ spans $\mathbb{R}^{d^2}$, since $\mathcal{X}$ spans $\mathbb{R}^d$. Now for any $z \in \mathcal{B}$, define $\beta^{(z)} \in \mathbb{R}^{K^2}$ as a vector such that $z = \mathbf{Z}^\top \beta^{(z)}$. Then,

$$\big\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\big\| = \max_{z \in \mathcal{B}} \ {\beta^{(z)}}^\top \mathbf{Z} \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} \mathbf{Z}^\top \beta^{(z)}$$

$$= \max_{z\in\mathcal{B}} \sum_{i=1}^{d^2}\sum_{j=1}^{d^2} \beta_i^{(z)}\beta_j^{(z)} z_i^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z_j$$
$$\leq \max_{z\in\mathcal{B}} \left\|\beta^{(z)}\right\|_1^2 \times \max_{i,j}\ z_i^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z_j \ .$$

Define $\tilde{z}_i = \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-\frac{1}{2}} z_i$. Clearly, $\max_{i,j}\ z_i^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z_j = \max_{i,j}\ \tilde{z}_i^\top \tilde{z}_j = \max_i\ \tilde{z}_i^2$. So we have

$$\left\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\right\| \leq \max_{z\in\mathcal{B}} \left\|\beta^{(z)}\right\|_1^2 \times \max_{z'\in\mathcal{Z}}\ z'^\top \Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1} z'$$
$$\leq \max_{z\in\mathcal{B}} \left\|\beta^{(z)}\right\|_1^2 d^2 \ .$$

The last inequality comes from Kiefer and Wolfowitz equivalence theorem [53]. Now observe that $\beta^{(z)}$ can be obtained by least square regression : $\beta^{(z)} = \left(\mathbf{Z}\mathbf{Z}^\top\right)^{-1}\mathbf{Z}z = \left(\mathbf{Z}^\top\right)^\dagger z$ where $(\cdot)^\dagger$ is the Moore-Penrose pseudo-inverse. Note that $\mathbf{Z}\mathbf{Z}^\top$ is a Gram matrix. It is known that for a matrix having singular values $\{\sigma_i\}_i$, its pseudo-inverse has singular values $\begin{cases} \frac{1}{\sigma_i} & \text{if } \sigma_i \neq 0 \\ 0 & \text{otherwise} \end{cases}$ for all i. So for $z \in \mathcal{B}$, we have:

$$\left\|\beta^{(z)}\right\|_1^2 \leq K^2 \left\|\beta^{(z)}\right\|_2^2 \leq K^2 \left\|\left(\mathbf{Z}^\top\right)^\dagger\right\|^2 \leq \frac{K^2}{\sigma_{\min}(\mathbf{Z})^2} \ ,$$

where $\sigma_{\min}(\cdot)$ refers to the smallest singular value. Let $\lambda_{\min}(\cdot)$ refer to the smallest eigenvalue. Noting that

$$\sigma_{\min}(\mathbf{Z})^2 = \lambda_{\min}\left(\mathbf{Z}^\top\mathbf{Z}\right) = K^2\lambda_{\min}\left(\frac{1}{K^2}\sum_{z\in\mathcal{Z}} zz^\top\right) \ ,$$

yields the desired result.

□

We can now prove the main theorem.

Proof of Theorem 4.2. Let $(X_s^{(1)})_{s=1,\dots,t}, \dots, (X_s^{(n)})_{s=1,\dots,t}$ be nt i.i.d. random vectors in $\mathbb{R}^d$ such that for all $x \in \mathcal{X}$, $\mathbb{P}\left(X_1^{(1)} = x\right) = \gamma_x^\star$. For $(i,j) \in E$ and $1 \leq s \leq t$, we define the random matrix $\mathbf{Z}_s^{(i,j)}$ by

$$\mathbf{Z}_s^{(i,j)} = \mathrm{vec}\left(X_s^i X_s^{j\top}\right)\mathrm{vec}\left(X_s^i X_s^{j\top}\right)^\top \ .$$

Finally, let us define for all $1 \leq s \leq t$, the edge-wise sum $\mathbf{Z}_s \in \mathbb{R}^{d^2 \times d^2}$, that is

$$\mathbf{Z}_s = \sum_{(i,j)\in E} \mathbf{Z}_s^{(i,j)} \ .$$

One can easily notice that $\mathbf{Z}_1, \ldots, \mathbf{Z}_t$ are i.i.d. random matrices. We define the overall sum $\mathbf{A}_t = \sum_{s=1}^{t} \mathbf{Z}_s$ and our goal is to measure how close $f_{\mathcal{Z}}(\mathbf{A}_t)$ is to $f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star))$, where mt corresponds to the total number of sampled arms $z \in \mathcal{Z}$ during the t rounds of the learning procedure. By definition of $\mathbf{A}_t$, one has

$$\begin{aligned}
\max_{z\in\mathcal{Z}} \ z^\top (\mathbb{E}\mathbf{A}_t)^{-1} z &= \max_{z\in\mathcal{Z}} \ z^\top \left(\sum_{s=1}^{t} \sum_{(i,j)\in E} \mathbb{E}\left[\mathbf{Z}_s^{(i,j)}\right] \right)^{-1} z \\
&= \max_{z\in\mathcal{Z}} \ z^\top \left(\sum_{s=1}^{t} \sum_{(i,j)\in E} \sum_{x,x'\in\mathcal{X}} \gamma_x^\star \gamma_{x'}^\star \operatorname{vec}\left(xx'^\top\right) \operatorname{vec}\left(xx'^\top\right)^\top \right)^{-1} z \\
&= \max_{z\in\mathcal{Z}} \ z^\top \left(\sum_{s=1}^{t} \sum_{(i,j)\in E} \sum_{z'\in\mathcal{Z}} \Gamma_{z'}^\star z' z'^\top \right)^{-1} z \\
&= f_{\mathcal{Z}}(mt\Sigma_{\mathcal{Z}}(\Gamma^\star)) \ .
\end{aligned}$$

This allows us to bound the relative error as follows:

$$\begin{aligned}
\beta_t &= \frac{f_{\mathcal{Z}}(\mathbf{A}_t)}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star))} - 1 \\
&= \frac{\max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} + (\mathbb{E}\mathbf{A}_t)^{-1} \right) z}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star))} - 1 \\
&\leq \frac{\max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} \right) z}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star))} \ .
\end{aligned}$$

Using the fact that $f_{\mathcal{Z}}(mt\Sigma_{\mathcal{Z}}(\Gamma^\star)) = d^2/mt$ [53], we obtain

$$\begin{aligned}
\beta_t &\leq \frac{mt}{d^2} \times \max_{z\in\mathcal{Z}} \ z^\top \left(\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1} \right) z \\
&\leq \frac{mt}{d^2} \times \max_{z\in\mathcal{Z}} \ \|z\|^2 \|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\| \\
&\leq \frac{mtL}{d^2} \times \|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\| \ .
\end{aligned}$$

Therefore, controlling the quantity $\|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\|$ will allow us to provide an upper bound on the relative error. Notice that

$$\begin{aligned}\|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\| &= \|\mathbf{A}_t^{-1}(\mathbb{E}\mathbf{A}_t - \mathbf{A}_t)(\mathbb{E}\mathbf{A}_t)^{-1}\| \\ &\leq \|\mathbf{A}_t^{-1}\| \, \|\mathbb{E}\mathbf{A}_t - \mathbf{A}_t\| \, \|(\mathbb{E}\mathbf{A}_t)^{-1}\| \ .\end{aligned}$$

Using Proposition 4.3, we know that for any $d^2 e^{-\frac{t\lambda_{\min}(\mathbb{E}\mathbf{Z}_1)}{mL}} < \delta_h < 1$, the following holds:

$$\|\mathbf{A}_t^{-1}\| \leq \frac{\|(\mathbb{E}\mathbf{A}_t)^{-1}\|}{1 - \sqrt{\frac{2mL}{t}\|(\mathbb{E}\mathbf{Z}_1)^{-1}\| \log(d^2/\delta_h)}} \ ,$$

with probability at least $1 - \delta_h$. Similarly, using Proposition 4.4, for any $0 < \delta_b < 1$, we have

$$\|\mathbf{A}_t - \mathbb{E}\mathbf{A}_t\| \leq \frac{mL}{3} \log \frac{d^2}{\delta_b} + \sqrt{2tv^2 \log \frac{d^2}{\delta_b}} \ ,$$

with probability at least $1 - \delta_b$. Combining these two results with a union bound leads to the following bound, with probability $1 - (\delta_b + \delta_h)$:

$$\left\|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\right\| \leq \left\|(\mathbb{E}\mathbf{A}_t)^{-1}\right\|^2 \frac{(mL/3)\log\big(d^2/\delta_b\big) + \sqrt{2tv\log(d^2/\delta_b)}}{1 - \sqrt{(2mL/t)\left\|(\mathbb{E}\mathbf{Z}_1)^{-1}\right\| \log(d^2/\delta_h)}} \ .$$

In order to obtain a unified bound depending on one confidence parameter $1-\delta$, one could optimize over δ_b and δ_h, subject to $\delta_b + \delta_h = \delta$. This leads to a messy result and a negligible improvement. One can use simple values $\delta_b = \delta_h = \delta/2$, so the overall bound becomes, with probability $1 - \delta$:

$$\|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\| \leq \frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\right\|^2 \sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} \left(\frac{1 + \sqrt{\frac{m^2L^2\log(2d^2/\delta)}{18vt}}}{1 - \sqrt{\frac{2L\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\|\log(2d^2/\delta)}{t}}}\right) \ .$$

This can finally be formulated as follows:

$$\left\|\mathbf{A}_t^{-1} - (\mathbb{E}\mathbf{A}_t)^{-1}\right\| \leq \frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\right\|^2 \sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{t\sqrt{t}}\right) \ .$$

Using the obtained bound on $\|\mathbf{A}_t^{-1} - \mathbb{E}(\mathbf{A}_t)^{-1}\|$ yields

$$\frac{f_{\mathcal{Z}}(\mathbf{A}_t)}{f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star))} - 1 \leq \frac{mtL}{d^2} \times \left(\frac{1}{tm^2}\left\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\right\|^2 \sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{t\sqrt{t}}\right)\right)$$

$$\leq \frac{L}{md^2}\left\|\Sigma_{\mathcal{Z}}(\Gamma^\star)^{-1}\right\|^2\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{\sqrt{t}}\right) \ ,$$

By noticing that $f_{\mathcal{Z}}(mt \times \Sigma_{\mathcal{Z}}(\Gamma^\star)) \leq f_{\mathcal{Z}}(\mathbf{A}_t^\star)$ and by using Lemma 4.2, the result holds. □

We have just shown that the approximation value $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ converges to the optimal value with a rate of $O\big(\sqrt{v}/(m\sqrt{t})\big)$. In Section 4.3.1, we show that the best case graph implies a $v = O(m)$ matching the convergence rate $O\big(1/\sqrt{mt}\big)$ of a linear bandit algorithm using randomized sampling to pull mt edge-arms without (graphical) constraints. Moreover, we will see that the worst case graph implies that $v = O\big(m^2\big)$.

Since we filled the gap between our constraint objective and the problem of best arm identification in linear bandits, thanks to Theorem 4.1 and 4.2, we are able to extend known results for best arm identification in linear bandits on the sample complexity and its associated lower bound.

Corollary 4.1 ([90], Theorem 1)**.** *If the G-allocation is implemented with the random strategy of Algorithm 8, resulting in an β_t-approximation, then with probability at least $1-\delta$, the best arm obtained with $\hat{\theta}_t$ is $z_\star$ and*

$$t \leq \frac{128\sigma^2 d^2(1+\beta_t)\log\left(\frac{6m^2t^2K^4}{\delta\pi}\right)}{m\Delta_{\min}^2} \ ,$$

where $\Delta_{\min} = \min_{z\in\mathcal{Z}\setminus\{z_\star\}}(z_\star - z)^\top\theta_\star$.

Moreover, let τ be the number of rounds sufficient for any algorithm to determine the best arm with probability at least $1-\delta$. A lower bound on the expectation of τ can be obtained from the one derived for the problem of best arm identification in linear bandits (see *e.g.*, Theorem 1 in [39]):

$$\mathbb{E}[\tau] \geq \min_{\Gamma\in\mathcal{S}_{\mathcal{Z}}}\max_{z\in\mathcal{Z}\setminus\{z_\star\}} \log\left(\frac{1}{2.4\delta}\right)\frac{2\sigma^2\|z_\star - z\|^2_{\Sigma_{\mathcal{Z}}(\Gamma)^{-1}}}{m\Big((z_\star - z)^\top\theta_\star\Big)^2} \ .$$

As observed in [90] this lower bound can be upper bounded, in the worst case, by $4\sigma^2 d^2/(m\Delta_{\min}^2)$ which matches our bound up to log terms and the relative error β_t. Note, however, that since we borrow this lower bound from the standard linear bandit literature, it does not take into account the graph constraint, so it may never be reached.

4.2.3 Case where $\mathbf{M}_\star$ is not symmetric

Consider the relaxation of the assumption we made at the beginning of the chapter, namely that $\mathbf{M}_\star$ is symmetric. We now consider that $\mathbf{M}_\star$ is not necessarily symmetric. We recall here that in the graph $\mathcal{G} = (V, E)$ associated with the graphical bilinear bandits framework, $(i, j) \in E$ if and only if $(j, i) \in E$. Therefore, for a given allocation $(x^{(1)}, \ldots, x^{(n)}) \in \mathcal{X}^n$, we can write the associated expected global reward as follows:

$$\begin{aligned}
\sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)} &= \sum_{i=1}^{n}\sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(j)\top}\mathbf{M}_\star x^{(i)} \\
&= \sum_{i=1}^{n}\sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + \left(x^{(j)\top}\mathbf{M}_\star x^{(i)}\right)^\top \\
&= \sum_{i=1}^{n}\sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(i)\top}\mathbf{M}_\star^\top x^{(j)} \\
&= \sum_{i=1}^{n}\sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\left(\mathbf{M}_\star x^{(j)} + \mathbf{M}_\star^\top x^{(j)}\right) \\
&= \sum_{i=1}^{n}\sum_{\substack{j\in\mathcal{N}(i)\\ j>i}} x^{(i)\top}\left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right)x^{(j)} \quad .
\end{aligned}$$

Let us denote $\tilde{\mathbf{M}}_\star = \mathbf{M}_\star + \mathbf{M}_\star^\top$. One can notice that $\tilde{\mathbf{M}}_\star$ is symmetric. Consider the edge set $\tilde{E} = \{(i,j) \in E \mid i \in V, j \in V, j > i\}$, there are exactly $m/2$ edges in $\tilde{E}$ and the objective of the central entity is to find, within a minimum number of rounds, the joint arm that maximises

$$\sum_{(i,j)\in\tilde{E}} x^{(i)\top}\tilde{\mathbf{M}}_\star x^{(j)} \quad . \tag{4.4}$$

Every time the central entity chooses a joint arm $(x_t^{(1)}, \dots, x_t^{(n)})$ at time t, it receives for each edge $(i,j) \in \tilde{E}$ the rewards $y_t^{(i,j)} = x_t^{(i)\top}\mathbf{M}x^{(j)}$ and $y_t^{(j,i)} = x_t^{(j)\top}\mathbf{M}x^{(i)}$. By summing $y_t^{(i,j)}$ and $y^{(j,i)}$, one has,

$$\begin{aligned}
y_t^{(i,j)} + y_t^{(j,i)} &= \left\langle \mathrm{vec}\left(x_t^{(i)}x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star\right)\right\rangle + \left\langle z_{x_t^{(j)}x_t^{(i)}}, \mathrm{vec}\left(\mathbf{M}_\star\right)\right\rangle + \left(\eta_t^{(i,j)} + \eta_t^{(j,i)}\right) \\
&= \left\langle \mathrm{vec}\left(x_t^{(i)}x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star\right)\right\rangle + \left\langle \mathrm{vec}\left(x_t^{(i)}x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star^\top\right)\right\rangle \\
&\quad + \left(\eta_t^{(i,j)} + \eta_t^{(j,i)}\right) \\
&= \left\langle \mathrm{vec}\left(x_t^{(i)}x_t^{(j)\top}\right), \mathrm{vec}\left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right)\right\rangle + \left(\eta_t^{(i,j)} + \eta_t^{(j,i)}\right) \\
&= \left\langle \mathrm{vec}\left(x_t^{(i)}x_t^{(j)\top}\right), \mathrm{vec}\left(\tilde{\mathbf{M}}_\star\right)\right\rangle + \left(\eta_t^{(i,j)} + \eta_t^{(j,i)}\right) \\
&= \left\langle z_t^{(i,j)}, \tilde{\theta}_\star\right\rangle + \underbrace{\left(\eta_t^{(i,j)} + \eta_t^{(j,i)}\right)}_{\sqrt{2}\sigma\text{-sub-gaussian random variable}} \quad ,
\end{aligned}$$

where $\tilde{\theta}_\star = \mathrm{vec}\left(\tilde{\mathbf{M}}_\star\right)$ is the vectorized version of the true parameter matrix $\tilde{\mathbf{M}}_\star$.

Hence at each round t, and for each edge $(i,j) \in \tilde{E}$, the central entity aggregate the reward $y_t^{(i,j)}$ and $y_t^{(j,i)}$ to get a reward of the form $\langle z, \tilde{\theta}_\star \rangle + \eta$ with $z \in \mathcal{Z}$ and where η is a $\sqrt{2}\sigma$-subgaussian random variable, that can be used to compute an estimate $\hat{\theta}_t$ of $\tilde{\theta}_\star$. Notice that the amount of rewards $y_t^{(i,j)}$ obtained during a round is m, but since the central entity needs to sum $y_t^{(i,j)} + y_t^{(j,i)}$ to compute $\hat{\theta}_t$, we thus consider only $m/2$ obtained information per round.

Solving this best arm identification problem for the graphical bilinear bandits with a non-symmetric matrix $\mathbf{M}_\star$ and an edge set E is exactly solving the best arm identification problem for graphical bilinear bandits with the symmetric matrix $\tilde{\mathbf{M}}_\star$ and the edge set $\tilde{E}$. The solution of this problem is exactly what we proposed throughout the previous sections.

4.3 Influence of the graph structure on the convergence rate

4.3.1 Characterization of the variance associated with the randomized strategy

The convergence bound in Theorem 4.2 depends on $v = \|\mathbb{E}\big[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2\big]\|$. In this section, we characterize the impact of the graph structure on this quantity and, by extension, on the convergence rate.

For $i \in \{1, \dots, n\}$ and $s \in \{1, \dots, t\}$, let $X_s^{(i)}$ be *i.i.d.* random vectors in $\mathcal{X}$ such that for all $x \in \mathcal{X}$,

$$\mathbb{P}\Big(X_1^{(1)} = x\Big) = \gamma_x^\star \ .$$

Each $X_s^{(i)}$ is to be viewed as the random arm pulled at round s for the node i. Hence one can write

$$\mathbf{A}_1 = \sum_{(i,j)\in E} \mathrm{vec}\left(X_1^{(i)} X_1^{(j)\top}\right) \mathrm{vec}\left(X_1^{(i)} X_1^{(j)\top}\right)^\top \ .$$

Let denote $\mathbf{A}_1^{(i,j)} = \mathrm{vec}\,(X_1^{(i)} X_1^{(j)\top})\,\mathrm{vec}\,(X_1^{(i)} X_1^{(j)\top})^\top$ such that $\mathbf{A}_1 = \sum_{(i,j)\in E} \mathbf{A}_1^{(i,j)}$ and let define for any random matrices $\mathbf{A}$ and $\mathbf{B}$ the operators $\mathrm{Var}(\mathbf{A}) \triangleq \mathbb{E}\big[(\mathbf{A} - \mathbb{E}[\mathbf{A}])^2\big]$ and $\mathrm{Cov}(\mathbf{A}, \mathbf{B}) \triangleq \mathbb{E}\big[(\mathbf{A} - \mathbb{E}[\mathbf{A}])(\mathbf{B} - \mathbb{E}[\mathbf{B}])\big]$. We can derive the variance of $\mathbf{A}_1$ as follows:

$$\begin{aligned}\mathrm{Var}(\mathbf{A}_1) = &\sum_{(i,j)\in E} \mathrm{Var}\Big(\mathbf{A}_1^{(i,j)}\Big) \\ &+ \sum_{(i,j)\in E} \sum_{\substack{(k,l)\in E \\ (k,l)\neq(i,j)}} \mathrm{Cov}\Big(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(k,l)}\Big) \ .\end{aligned}$$

One can decompose the sum of the covariances into three groups: a first group where $k \neq i, j$ and $l \neq i, j$ which means that the two edges do not share any node and $\mathrm{Cov}(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(k,l)}) = \mathbf{0}$, and two other groups where the edges share at least one node. For all edges $(i,j) \in E$ we consider

either the edges $(i,k) \in E$ where $k \neq j$, yielding $\mathrm{Cov}(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(i,k)})$ or the edges $(j,k) \in E$, yielding $\mathrm{Cov}(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(j,k)})$.

Hence, one has

$$\begin{aligned}\mathrm{Var}(\mathbf{A}_1) = &\sum_{(i,j)\in E} \mathrm{Var}\Big(\mathbf{A}_1^{(i,j)}\Big) \\ &+ \sum_{i=1}^{n} \sum_{j\in\mathcal{N}(i)} \sum_{\substack{k\in\mathcal{N}(i)\\k\neq j}} \mathrm{Cov}\Big(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(i,k)}\Big) \\ &+ \sum_{i=1}^{n} \sum_{j\in\mathcal{N}(i)} \sum_{k\in\mathcal{N}(j)} \mathrm{Cov}\Big(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(j,k)}\Big) \ .\end{aligned}$$

Let $P \geq 0$ be such that for all $(i,j) \in E$, $\mathrm{Var}\Big(\mathbf{A}_1^{(i,j)}\Big) \preceq P \times \mathbf{I}$ and $M, N \geq 0$ such that for all $(i,j) \in E$:

$$\begin{aligned}\forall k \in \mathcal{N}(i), \mathrm{Cov}\Big(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(i,k)}\Big) &\preceq M \times \mathbf{I} \ , \\ \forall k \in \mathcal{N}(j), \mathrm{Cov}\Big(\mathbf{A}_1^{(i,j)}, \mathbf{A}_1^{(j,k)}\Big) &\preceq N \times \mathbf{I} \ .\end{aligned}$$

We want to compare the quantity $\|\mathrm{Var}(\mathbf{A}_1)\|$ for different types of graphs: star, complete, circle and a matching graph. To have a fair comparison, we want graphs that reveal the same number of rewards at each round of the learning procedure. Hence, we denote respectively n_{S}, n_{Co}, n_{Ci} and n_{M} the number of nodes in a star, complete, circle and matching graph of m edges and get:

Star graph. We have,

$$\mathrm{Var}(\mathbf{A}_1) \preceq m \times P \cdot \mathbf{I} + (n_{\mathrm{S}} - 1)(n_{\mathrm{S}} - 2)M \cdot \mathbf{I} + n_{\mathrm{S}}(n_{\mathrm{S}} - 1)N \cdot \mathbf{I} \ .$$

Since the star graph of m edges has a number of nodes $n_{\mathrm{S}} = m/2 + 1$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\| \leq m \times P + (M + N) \times O\big(m^2\big) \ .$$

Complete graph. As for the star graph,

$$\mathrm{Var}(\mathbf{A}_1) \preceq m \times P \cdot \mathbf{I} + n_{\mathrm{Co}}(n_{\mathrm{Co}} - 1)(n_{\mathrm{Co}} - 2)M \cdot \mathbf{I} + n_{\mathrm{Co}}(n_{\mathrm{Co}} - 1)(n_{\mathrm{Co}} - 1)N \cdot \mathbf{I} \ .$$

Since the complete graph of m edges has a number of nodes $n_{\mathrm{Co}} = \big(1 + \sqrt{4m+1}\big)/2$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\| \leq m \times P + (M + N) \times O\big(m\sqrt{m}\big) \ .$$

Circle graph. Again,

$$\mathrm{Var}(\mathbf{A}_1) \preceq m \times P \cdot \mathbf{I} + 2n_{\mathrm{Ci}}M \cdot \mathbf{I} + 4n_{\mathrm{Ci}}N \cdot \mathbf{I} \ .$$

Since the circle graph of m edges has a number of nodes $n_{\mathrm{Ci}} = m/2$, we have

$$\|\mathrm{Var}(\mathbf{Z}_1)\| \leq m \times P + (M+N) \times O(m) \ .$$

Matching graph. Finally,

$$\mathrm{Var}(\mathbf{A}_1) \preceq m \times P \cdot \mathbf{I} + n_{\mathrm{M}}N \cdot \mathbf{I} \ .$$

Since the matching graph of m edges has a number of nodes $n_{\mathrm{M}} = m$, we have

$$\|\mathrm{Var}(\mathbf{A}_1)\| \leq m \times P + m \times N \ .$$

We thus obtain the bounds stated in Table 4.2.

Graph	Upper bound on $\|\mathrm{Var}(\mathbf{A}_1)\|$	β_t
Star	$mP + (M+N)O(m^2)$	$O\big(1/\sqrt{t}\big)$
Complete	$mP + (M+N)O(m\sqrt{m})$	$O\Big(1/\Big(m^{\frac{1}{4}}\sqrt{t}\Big)\Big)$
Circle	$mP + (M+N)O(m)$	$O\big(1/\sqrt{mt}\big)$
Matching	$mP + mN$	$O\big(1/\sqrt{mt}\big)$

Table 4.2: Upper bound on the variance and convergence rate of Algorithm 8 for the star, complete, circle and matching graph with respect to the number of edges m and the number of rounds t.

These four examples evidence the strong dependency of the variance on the structure of the graph. The more independent the edges are (*i.e.*, with no common nodes), the smaller the quantity $\|\mathrm{Var}(\mathbf{A}_1)\|$ is. For a fixed number of edges m, the best case is the matching graph where no edge share the same node and the worst case is the star graph where all the edges share a central node.

4.3.2 Experimental results validating the dependence on the graph

In this section, we consider the modified version of a standard experiment introduced by [90] and used in most papers on best arm identification in linear bandits [39, 92, 106, 109] to evaluate the sample complexity of our algorithm on different graphs. We consider $d+1$ node-arms in $\mathcal{X} \subset \mathbb{R}^d$ where $d \geq 2$. This node-arm set is made of the d vectors $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ forming the canonical basis of $\mathbb{R}^d$ and one additional arm $x_{d+1} = (\cos(\omega), \sin(\omega), 0, \ldots, 0)^\top$ with $\omega \in]0, \pi/2]$. Note that by construction, the edge-arm set $\mathcal{Z}$ contains the canonical basis $(\mathbf{e}'_1, \ldots, \mathbf{e}'_{d^2})$ of $\mathbb{R}^{d^2}$. The parameter matrix $\mathbf{M}_\star$ has its first coordinate equals to 2 and the others equal to 0 which makes $\theta_\star = \mathrm{vec}\,(\mathbf{M}_\star) = (2, 0, \ldots, 0)^\top \in \mathbb{R}^{d^2}$. The best edge-arm is thus $z_\star = z^{(1,1)} = \mathbf{e}'_1$. One can note that when ω tends to 0, it is harder to differentiate this arm from $z^{(d+1,d+1)} =$

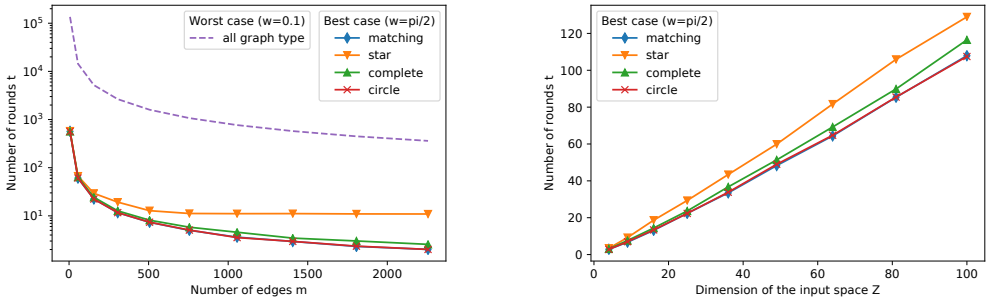


Figure 4.3: Number of rounds t needed to verify the stopping condition (4.3) with respect to **left:** the number of edges m where the dimension of the edge-arm space $\mathcal{Z}$ is fixed and equal to 25 and **right:** the dimension of the edge-arm space $\mathcal{Z}$ where the number of edges is fixed and equal to 156. For both experiments we run 100 times and plot the average number of rounds needed to verify the stopping condition.

$\text{vec}\left(x_{(d+1)}x_{(d+1)}^{\top}\right)$ than from the other arms. We set $\eta_t^{(i,j)} \sim \mathcal{N}(0,1)$, for all edges (i,j) and round t.

We consider two cases where $\omega = 0.1$ which makes the edge-arms $z^{(1,1)}$ and $z^{(d+1,d+1)}$ difficult to differentiate, and $\omega = \pi/2$ which makes the edge-arm $z^{(1,1)}$ easily identifiable as the optimal edge-arm. For each of these two cases, we evaluate the influence of the graph structure, the number of edges m and the edge-arm space dimension d^2 on the sampling complexity. Results are shown in Figure 4.3.

When $\omega = 0.1$, the type of the graph does not impact the number of rounds needed to verify the stopping condition. This is mainly due to the fact that the magnitude of its associated variance is negligible with respect to the number of rounds. Hence, even if we vary the number of edges or the dimension, we get the same performance for any type of graph including the matching graph. This implies that our algorithm performs as well as a linear bandit that draws m edge-arms in parallel at each round. When $\omega = \pi/2$, the number of rounds needed to verify the stopping condition is smaller and the magnitude of the variance is no longer negligible. Indeed, when the number of edges or the dimension increases, we notice that the star graph takes more times to satisfy the stopping condition. Moreover, note that the sample complexities obtained for the circle and the matching graph are similar. This observation is in line with the dependency on the variance shown in Table 4.2.

4.4 Conclusion & Perspectives

We provided an algorithm based on the G-allocation strategy that uses randomized sampling over the nodes to return a good estimate $\hat{\mathbf{M}}$ that can be used instead of $\mathbf{M}_\star$ to identify the couple $(x_\star, x'_\star)$. Moreover, we highlighted the impact of the graph structure on the convergence rate of our algorithm and validated our theoretical results with experiments.

While the estimate $\hat{\mathbf{M}}$ is constructed in order to identify the couple $(x_\star, x'_\star)$ that is used in algoriothm 5 and gives a $\frac{1+\xi}{2}$-approximation solution, a perspective of improvement can be to

construct the estimate $\hat{\mathbf{M}}$ that identifies with high probability the couple $(\tilde{x}_\star, \tilde{x}'_\star)$ that is used in Algorithm 6 that gives the better $\left(\frac{1+\xi}{2} + \epsilon\right)$-approximation guarantee. We let this extension for future work.[1] Moreover, in this chapter we based our algorithm on the G-allocation strategy that minimises $\max_{z\mathcal{Z}} 2\|z\|_{A_t^{-1}}$ that is upperbound of $\max_{(z,z')\mathcal{Z}^2} \|z - z'\|_{A_t^{-1}}$, objective of XY-allocation strategy. An algorithm based on the XY-allocation represents a promising extensions for our model.

[1] The design of Algorithm 6 and its associated guarantee are posterior to the best arm identification algorithm that we proposed in this chapter. This extension is an ongoing work and might be added to the final version of this thesis.

5 Regret based algorithms for graphical bilinear bandits

Contents

In this chapter, we also assume that we do not know the parameter matrix $\mathbf{M}_\star$ and as described in the problem setting in Section 1.2, a central entity faces the graphical bilinear bandits problem where at each round it chooses an arm for each node of the graph and observes a bilinear reward for each edge of the graph. The objective of the central entity is the following:

Objective: *Design an algorithm that maximizes the expected cumulative global reward obtained during T rounds $\sum_{t=1}^{T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)}$.*

We will naturally rely on some ideas and results presented in Chapter 3 where the matrix was assumed to be known by the central entity. We follow the notations we established in section 1.2.1.

5.1 Optimism in the face of uncertainty for graphical bilinear bandits

5.1.1 Preliminaries

Let us recall that maximizing the cumulative rewards is equivalent to minimizing its associated regret. We thus define the global pseudo-regret over T rounds as follows:

$$R(T) = \sum_{t=1}^{T} \left[\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} - \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \right] .$$

We recall that the objective of the learner is to have a pseudo-regret $R(T)$, such that

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0$$

We know from Theorem 3.1 that finding the best joint arm

$$\left(x_\star^{(1)}, \dots, x_\star^{(n)}\right) = \underset{(x^{(1)},\dots,x^{(n)})\in\mathcal{X}}{\arg\max} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$$

is NP-Hard with respect to the number of agents n. We extend this result in the following corollary.

Corollary 5.1. *There does not exist a polynomial time algorithm in n such that*

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0 \ , \tag{5.1}$$

for any instance of the graphical bilinear bandits described in Section 1.2, unless $P = NP$.

Proof. Suppose that there exists a polynomial time algorithm in n such that $\lim_{T\to\infty} \frac{R(T)}{T} = 0$ for any instance of the graphical bilinear bandits described in Section 1.2. In particular consider the class of instances where the graph $\mathcal{G}$ is of degree 3 (*i.e.*, each agent has 3 neighbors), where $\mathcal{X} = \{e_0, e_1\}$ is the canonical basis in $\mathbb{R}^2$ and the parameter matrix $\mathbf{M}_\star = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. The pseudo-regret is such that

$$\exists f : \mathbb{R} \to \mathbb{R} \text{ with } \lim_{T\to\infty} f(T) = 0, \text{ such that } R(T) = T \times f(T)$$

with a computational time in $\text{poly}(n)$ per iteration.

Let us consider the best case scenario for the learner and assume that the matrix $\mathbf{M}_\star$ is known.

Then, for $\hat{t} = \arg\max_{t=1,\dots,T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)\top}$ we have

$$\begin{aligned} T \sum_{(i,j)\in E} x_{\hat{t}}^{(i)\top} \mathbf{M}_\star x_{\hat{t}}^{(j)\top} &\geq \sum_{t=1}^{T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)\top} \\ &\geq T \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} - T \times f(T) \ , \end{aligned}$$

which gives,

$$\sum_{(i,j)\in E} x_{\hat{t}}^{(i)\top} \mathbf{M}_\star x_{\hat{t}}^{(j)\top} \geq \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} - f(T) \ .$$

As $T \to \infty$, $f(T) \to 0$, hence consider the cases where

$$f(T) \leq c \times \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} \ ,$$

for a certain constant $c > 0$.

It gives that

$$\begin{aligned} \sum_{(i,j)\in E} x_{\hat{t}}^{(i)\top} \mathbf{M}_\star x_{\hat{t}}^{(j)\top} &\geq \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} - c \times \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} \\ &= (1-c) \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)\top} \ , \end{aligned} \tag{5.2}$$

with a computational time in $\mathrm{poly}(n)$ per iteration.

However, we know from the proof of Theorem 3.1 that solving the optimization problem $\max_{(x^{(1)},\dots,x^{(n)})} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)\top}$ is the same as solving the max-cut problem. Moreover, we know from Theorem 1 in [14] that there does not exist a polynomial time algorithm in the number of nodes that has an approximation ratio better than $\frac{330}{331} \approx 0.997$ of the optimal solution of the max-cut problem for any graphs of degree 3, unless $P = NP$. By taking $c = 0.002$, Eq. (5.2) gives an approximation ratio of 0.998. This concludes the proof. □

Hence, the objective of designing an algorithm with a sublinear regret in T is not feasible in polynomial time with respect to n. However, some NP-hard problems are α-approximable (for some $\alpha \in (0, 1]$), which means that there exists a polynomial-time algorithm guaranteed to produce solutions whose values are at least α times the value of the optimal solution. We refer the reader to chapter 3 for more information on approximating the optimal solution of our problem. For these kind of problems, it makes sense to consider the α-pseudo-regret as in [31, 52] which is defined for all $\alpha \in (0, 1]$ as follows

$$R_\alpha(T) \triangleq \sum_{t=1}^{T} \left[\alpha \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} - \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \right] \ ,$$

and set the objective of designing an algorithm with a sublinear α-regret.

Finally, as we did in chapter 4, let us recall that the reward obtained for each edge of the graph at each round can be seen as a noisy linear reward in higher dimension [51] with

$$y_t^{(i,j)} = \left\langle \text{vec}\left(x_t^{(i)} x_t^{(j)\top}\right), \text{vec}\left(\mathbf{M}_\star\right)\right\rangle + \eta_t^{(i,j)} \ .$$

To simplify the notation, let us refer to any $x \in \mathcal{X}$ as a *node-arm*. Let us use the notation $z_{xx'} \triangleq \text{vec}\left(xx'^\top\right)$, and define the arm set $\mathcal{Z} = \{z_{xx'} | (x, x') \in \mathcal{X}^2\}$ where any $z \in Z$ will be referred as an *edge-arm*. If the arm $x_t^{(i)} \in \mathcal{X}$ represents the node-arm allocated to the node $i \in V$ at time t, for each edge $(i, j) \in E$ we will denote the associated edge-arm by $z_t^{(i,j)} \triangleq \text{vec}\left(x_t^{(i)} x_t^{(j)\top}\right) \in \mathcal{Z}$ and define $\theta_\star = \text{vec}\left(\mathbf{M}_\star\right)$ the vectorized version of the unknown matrix $\mathbf{M}_\star$. With those notations, the (now) linear reward can be rewritten as follows:

$$y_t^{(i,j)} = \left\langle z_t^{(i,j)}, \theta_\star \right\rangle + \eta_t^{(i,j)} \ . \tag{5.3}$$

Assumption 5.1 (Bounded edge-arm norm and parameter norm)**.** *We consider that there exists $L > 0$, for all edge-arm $z \in \mathcal{Z}$, such that $\|z\|_2 \leq L$. Moreover, for some $S > 0$, the norm of the true parameter $\theta_\star$ is such that $\|\theta_\star\|_2 \leq S$.*

Assumption 5.2 (Positive and bounded rewards)**.** *We consider that for any $z \in \mathcal{Z}$, the associated expected reward $\langle z, \theta_\star \rangle$ is such that $0 \leq \langle z, \theta_\star \rangle \leq LS$*

In this chapter, we choose to design an algorithm based on the principle of optimism in the face of uncertainty [12], and in the case of a linear reward [2, 63], we need to maintain an estimator of the true parameter $\theta_\star$. To do so, let us define for all rounds $t \in \{1, \ldots, T\}$ the OLS estimator of $\theta_\star$ as follows:

$$\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \ , \tag{5.4}$$

where,

$$\mathbf{A}_t = \lambda \mathbf{I}_{d^2} + \sum_{s=1}^{t} \sum_{(i,j)\in E} z_s^{(i,j)} z_s^{(i,j)\top} \ ,$$

with $\lambda > 0$ a regularization parameter and

$$b_t = \sum_{s=1}^{t} \sum_{(i,j)\in E} z_s^{(i,j)} y_s^{(i,j)} \ .$$

We define also the confidence set

$$C_t(\delta) = \left\{ \theta : \|\theta - \hat{\theta}_t\|_{\mathbf{A}_t^{-1}} \leq \sigma \sqrt{d^2 \log\left(\frac{1 + tmL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S \right\} \ ,$$

where with probability $1 - \delta$, we have that $\theta_\star \in C_t(\delta)$ for all $t \in \{1, \dots, T\}$, and $\delta \in (0, 1]$.

5.1.2 Algorithm and analysis of the regret

In chapter 3, we presented two algorithms (Algorithm 5 and 6) that use the true parameter matrix $\mathbf{M}_\star$ to return an allocation of arm that achieve an α-approximation solution of maximizing the global reward. Naturally, since we do not have access to $\mathbf{M}_\star$, we cannot use it directly at each iteration to maximise the cumulative global reward (and thus minimize the associated regret). Nevertheless, one can use the constructed estimator $\hat{\theta}_t$ and the principal of optimism in the face of uncertainty to overcome the fact that $\mathbf{M}_\star$ is unknown.

Indeed, we recall that in Algorithm 5, the couple $(x_\star, x'_\star)$ is chosen as follows,

$$(x_\star, x'_\star) = \arg\max_{(x,x') \in \mathcal{X}^2} x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top \right) x' \tag{5.5}$$

$$= \arg\max_{(x,x') \in \mathcal{X}^2} \langle z_{xx'} + z_{x'x}, \theta_\star \rangle \ , \tag{5.6}$$

and is used to create as much as possible edge arms of the form $z_{xx'}$ and $z_{x'x}$ in the graph. Here instead, at each round t, the central entity chooses optimistically the couple (x_t, x'_t) as follows,

$$(x_t, x'_t) = \arg\max_{(x,x') \in \mathcal{X}^2} \max_{\theta \in C_{t-1}(\delta)} \langle z_{xx'} + z_{x'x}, \theta \rangle \ ,$$

and then allocates the node-arms to maximize the number of locally-optimal edge-arms $z_{x_t x'_t}$ and $z_{x'_t x_t}$. The method is presented in Algorithm 9

Theorem 5.1. *Given the Graphical Bilinear Bandits problem defined in Section 1.2.1, let* $0 \leq \xi \leq 1$ *be a problem-dependent parameter defined by*

$$\xi = \min_{x \in \mathcal{X}} \frac{\langle z_{xx}, \theta_\star \rangle}{\frac{1}{m} \sum_{(i,j) \in E} \left\langle z_\star^{(i,j)}, \theta_\star \right\rangle} \geq 0 \ ,$$

and set $\alpha = \frac{1+\xi}{2}$*, then the* α*-regret of Algorithm 9 satisfies*

$$R_\alpha(T) \leq \tilde{O}\left(\left(\sigma d^2 + S\sqrt{\lambda} \right) \sqrt{Tm \max(2, (LS)^2)} \right) + LSm \left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta} \right) \right\rceil \ ,$$

where $\tilde{O}$ *hides the logarithmic factors.*

Algorithm 9: Adaptation of OFUL algorithm for Graphical Bilinear Bandits

Input : graph $\mathcal{G} = (V, E)$, node-arm set $\mathcal{X}$
$(V_1, V_2) = \text{Approx-MAX-CUT}(\mathcal{G})$
for $t = 1$ *to* T **do**
  // Find the optimistic best couple of node-arms
  $\left(x_t, x'_t, \tilde{\theta}_{t-1}\right) = \arg\max_{(x,x',\theta)\in\mathcal{X}^2\times C_{t-1}} \langle z_{xx'} + z_{x'x}, \theta\rangle$;
  // Allocate x_t and x'_t in the graph
  $x_t^{(i)} = x_t$ for all i in V_1; $x_t^{(i)} = x'_t$ for all i in V_2;
  Obtain for all (i, j) in E the rewards $y_t^{(i,j)}$;
  Compute $\hat{\theta}_t$ as in (5.4)
end
return $\hat{\theta}_t$

Proof. To properly derive the regret bounds, we will have to make connections between our setting and a standard linear bandit that chooses sequentially Tm arms. For that matter, let us consider an arbitrary order on the set of edges E and denote $E[i]$ the i-th edge according to this order with $i \in \{1, \dots, m\}$. We define for all $t \in \{1, \dots, T\}$ and $p \in \{1, \dots, m\}$ the OLS estimator

$$\hat{\theta}_{t,p} = \mathbf{A}_{t,p}^{-1} b_{t,p} \ ,$$

where

$$\mathbf{A}_{t,p} = \lambda \mathbf{I}_{d^2} + \sum_{s=1}^{t-1}\sum_{b=1}^{m} z_s^{E[b]} z_s^{E[b]\top} + \sum_{k=1}^{p} z_t^{E[k]} z_t^{E[k]\top} \ ,$$

with λ a regularization parameter and

$$b_{t,p} = \sum_{s=1}^{t-1}\sum_{b=1}^{m} z_s^{E[b]} y_s^{E[b]} + \sum_{k=1}^{p} z_t^{E[k]} y_t^{E[k]} \ . \tag{5.7}$$

We define also the confidence set

$$C_{t,p}(\delta) = \left\{\theta : \|\theta - \hat{\theta}_{t,p}\|_{A_{t,p}^{-1}} \leq \sigma\sqrt{d^2 \log\left(\frac{1 + tmL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S\right\} \ , \tag{5.8}$$

where with probability $1 - \delta$, we have that $\theta_\star \in C_{t,p}(\delta)$ for all $t \in \{1, \dots, T\}$, $p \in \{1, \dots, m\}$ and $\delta \in (0, 1]$.

Notice that the confidence set $C_t(\delta)$ defined in Section 5.1.1 is exactly the confidence set $C_{t,m}(\delta)$ defined here. The definitions of the matrix $A_{t,m}$ and the vector $b_{t,m}$ follow the same reasoning.

Recall that $(x_\star^{(1)}, \dots, x_\star^{(n)}) = \arg\max_{(x^{(1)},\dots,x^{(n)})} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ is the optimal joint arm, and we define for each edge $(i,j) \in E$ the optimal edge arm $z_\star^{(i,j)} = \text{vec}\,(x_\star^{(i)} x_\star^{(j)\top})$.

Let us borrow the notion of *Critical Covariance Inequality* introduced in [30], that is for a given round $t \in \{1,\dots,T\}$ and $p \in \{1,\dots,m\}$, the expected covariance matrix $\mathbf{A}_{t,p}$ satisfies the critical covariance inequality if

$$\mathbf{A}_{t-1,m} \preccurlyeq \mathbf{A}_{t,p} \preccurlyeq 2\mathbf{A}_{t-1,m} \ . \tag{5.9}$$

Let us now define the event D_t as the event where at a given round t, for all $p \in \{1,\dots,m\}$, $\mathbf{A}_{t,p}$ satisfies the critical covariance inequality (CCI).

We can write the pseudo-regret as follows:

$$\begin{aligned} R(T) &= \sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star \right\rangle \\ &\quad + \sum_{t=1}^{T} \mathbb{1}[D_t^c] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star \right\rangle \\ &\leq \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star \right\rangle}_{(a)} + \underbrace{LSm \sum_{t=1}^{T} \mathbb{1}[D_t^c]}_{(b)} \ . \end{aligned}$$

We know that the approximation Max-CUT algorithm returns two subsets of nodes V_1 and V_2 such that there are at least $m/2$ edges between V_1 and V_2, and to be more precise: at least $m/4$ edges from V_1 to V_2 and at least $m/4$ edges from V_2 to V_1. Hence at each time t, if all the nodes of V_1 pull the node-arm x_t and all the nodes of V_2 pull the node-arm x'_t, we can derive the term (a) as follows:

$$\begin{aligned} (a) &= \sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \\ &= \sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \mathbb{1}[i \in V_1 \wedge j \in V_2]\langle z_t^{(i,j)}, \theta_\star\rangle \\ &\qquad\qquad - \mathbb{1}[i \in V_2 \wedge j \in V_1]\langle z_t^{(i,j)}, \theta_\star\rangle \end{aligned}$$

$$- \mathbb{1}[i \in V_1 \wedge j \in V_1]\langle z_t^{(i,j)}, \theta_\star\rangle$$
$$- \mathbb{1}[i \in V_2 \wedge j \in V_2]\langle z_t^{(i,j)}, \theta_\star\rangle \ .$$

Notice that $\sum_{(i,j)\in E} z_\star^{(i,j)} = \sum_{(i,j)\in E} \frac{1}{m} \sum_{(k,l)\in E} z_\star^{(k,l)}$, so one has

$$(a) = \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \mathbb{1}[i \in V_1 \wedge j \in V_2] \left(\left\langle \frac{1}{m} \sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \right)}_{(a_1)}$$
$$+ \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \mathbb{1}[i \in V_2 \wedge j \in V_1] \left(\left\langle \frac{1}{m} \sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \right)}_{(a_2)}$$
$$+ \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \mathbb{1}[i \in V_1 \wedge j \in V_1] \left(\left\langle \frac{1}{m} \sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \right)}_{(a_3)}$$
$$+ \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \mathbb{1}[i \in V_2 \wedge j \in V_2] \left(\left\langle \frac{1}{m} \sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \right)}_{(a_4)} \ .$$

Let us analyse the first term. By using the fact that $\mathbb{1}[D_t] \leq 1$, we have

$$(a_1) = \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in N_i \\ j>i}} \mathbb{1}[i \in V_1 \wedge j \in V_2] \left\langle \frac{2}{m} \sum_{(k,l)\in E} z_\star^{(k,l)} - \left(z_t^{(i,j)} + z_t^{(j,i)} \right), \theta_\star \right\rangle \ .$$

By defining $(x_\star, x'_\star) = \arg\max_{(x,x')\in\mathcal{X}^2} \langle z_{xx'} + z_{x'x}, \theta_\star\rangle$, and noticing that in the case where a node i is in V_1 and a neighbouring node j in is V_2, then $z_t^{(i,j)} = z_{x_t x'_t}$, we have,

$$\begin{aligned}
\frac{2}{m} \sum_{(k,l)\in E} \left\langle z_\star^{(k,l)}, \theta_\star \right\rangle &= \frac{2}{m} \sum_{k=1}^{n} \sum_{\substack{j\in\mathcal{N}_k \\ j>k}} \left\langle z_\star^{(k,l)} + z_\star^{(l,k)}, \theta_\star \right\rangle \\
&\leq \frac{2}{m} \sum_{k=1}^{n} \sum_{\substack{j\in\mathcal{N}_k \\ j>k}} \left\langle z_{x_\star x'_\star} + z_{x'_\star x_\star}, \theta_\star \right\rangle \\
&= \langle z_{x_\star x'_\star} + z_{x'_\star x_\star}, \theta_\star \rangle
\end{aligned}$$

$$\leq \langle z_{x_t x'_t} + z_{x'_t x_t}, \tilde{\theta}_{t-1,m} \rangle \quad \text{(optimistic reward)}$$
$$= \langle z_t^{(i,j)} + z_t^{(j,i)}, \tilde{\theta}_{t-1,m} \rangle \ .$$

Plugging this last inequality yields, with probability $1-\delta$,

$$(a_1) \leq \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j \in N_i \\ j>i}} \mathbb{1}[i \in V_1 \wedge j \in V_2] \left\langle z_t^{(i,j)} + z_t^{(j,i)}, \tilde{\theta}_{t-1,m} - \theta_\star \right\rangle$$
$$= \sum_{t=1}^{T} \sum_{(i,j)\in E} \mathbb{1}[i \in V_1 \wedge j \in V_2] \left\langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} - \theta_\star \right\rangle \ .$$

We define, as in Algorithm 9, $\mathbb{1}\left[z_t^{(i,j)} = z_{x_t x'_t}\right] \triangleq \mathbb{1}[i \in V_1 \wedge j \in V_2]$. Then, one has, with probability $1-\delta$,

$$(a_1) \leq \sum_{t=1}^{T} \sum_{(i,j)\in E} \mathbb{1}\left[z_t^{(i,j)} = z_{x_t x'_t}\right] \left\langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} - \theta_\star \right\rangle$$
$$= \sum_{t=1}^{T} \sum_{k=1}^{m} \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \left\langle z_t^{E[k]}, \tilde{\theta}_{t-1,m} - \theta_\star \right\rangle$$
$$= \sum_{t=1}^{T} \sum_{k=1}^{m} \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \left\langle z_t^{E[k]}, \tilde{\theta}_{t-1,m} - \hat{\theta}_{t-1,m} \right\rangle + \left\langle z_t^{E[k]}, \hat{\theta}_{t-1,m} - \theta_\star \right\rangle$$
$$\leq \sum_{t=1}^{T} \sum_{k=1}^{m} \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \|\tilde{\theta}_{t-1,m} - \hat{\theta}_{t-1,m}\|_{\mathbf{A}_{t,k-1}}$$
$$+ \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \|\hat{\theta}_{t-1,m} - \theta_\star\|_{\mathbf{A}_{t,k-1}}$$
$$\leq \sum_{t=1}^{T} \sum_{k=1}^{m} \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \sqrt{2} \|\tilde{\theta}_{t-1,m} - \hat{\theta}_{t-1,m}\|_{\mathbf{A}_{t-1,m}} \tag{5.10}$$
$$+ \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \sqrt{2} \|\hat{\theta}_{t-1,m} - \theta_\star\|_{\mathbf{A}_{t-1,m}}$$
$$\leq \sum_{t=1}^{T} \sum_{k=1}^{m} \mathbb{1}\left[z_t^{E[k]} = z_{x_t x'_t}\right] 2\sqrt{2\beta_t(\delta)} \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \tag{5.11}$$
$$\leq \sum_{t=1}^{T} \sum_{k=1}^{m} 2\sqrt{2\beta_t(\delta)} \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \ , \tag{5.12}$$

with $\sqrt{\beta_t(\delta)} \leq \sigma\sqrt{d^2 \log\left(\frac{1+tmL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S$ and where (5.10) uses the critical covariance inequality (5.9), (5.11) comes from the definition of the confidence set $C_{t-1,m}(\delta)$ (5.8) and (5.12) upper bounds the indicator functions by 1.

Using a similar reasoning, we obtain the same bound for (a_2):

$$(a_2) \leq \sum_{t=1}^{T}\sum_{k=1}^{m} 2\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} \ .$$

Let us bound the terms (a_3) and (a_4).

$$(a_3) \leq \sum_{t=1}^{T}\sum_{(i,j)\in E} \mathbb{1}\left[z_t^{(i,j)} = z_{x_t x_t}\right]\left(\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle\right) \ .$$

For all $x \in \mathcal{X}$, let ξ_x be the following ratio

$$\xi_x = \frac{\langle z_{xx}, \theta_\star\rangle}{\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle} \ , \tag{5.13}$$

and let ξ be the worst ratio

$$\xi = \min_{x\in\mathcal{X}} \frac{\langle z_{xx}, \theta_\star\rangle}{\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle} \ . \tag{5.14}$$

We have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{(i,j)\in E} \mathbb{1}\left[z_t^{(i,j)} = z_{x_t x_t}\right]\left(\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \xi_{x_t}\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle\right) \\
\leq &\sum_{t=1}^{T}\sum_{(i,j)\in E} \mathbb{1}\left[z_t^{(i,j)} = z_{x_t x_t}\right]\left(\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle - \xi\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle\right) \\
= &\sum_{t=1}^{T}\sum_{(i,j)\in E} \mathbb{1}\left[z_t^{(i,j)} = z_{x_t x_t}\right](1-\xi)\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle \\
\leq &T\frac{m}{4}(1-\xi)\left\langle \frac{1}{m}\sum_{(k,l)\in E} z_\star^{(k,l)}, \theta_\star \right\rangle
\end{aligned} \tag{5.15}$$

$$= \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{1}{4}(1-\xi)\left\langle z_\star^{(i,j)}, \theta_\star \right\rangle \ ,$$

where (5.15) comes from the fact that there is at most $m/4$ edges that goes from node in V_1 to other nodes in V_1.

The derivation of this bound for (a_3) gives the same one for (a_4)

$$(a_4) \leq \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{1}{4}(1-\xi)\left\langle z_\star^{(i,j)}, \theta_\star \right\rangle \ .$$

By rewriting (a), we have :

$$(a) \leq \sum_{t=1}^{T}\sum_{k=1}^{m} 4\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} + \frac{1}{2}(1-\xi)\langle z_\star^{(i,j)}, \theta_\star\rangle \ .$$

In [30], they bounded the term (b) as follows

$$LSm\sum_{t=1}^{T} \mathbb{1}[D_t^c] \leq LSm\left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ . \tag{5.16}$$

We thus have the regret bounded by

$$\begin{aligned} R(T) \leq & \sum_{t=1}^{T}\sum_{k=1}^{m} 4\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} + \frac{1}{2}(1-\xi)\langle z_\star^{(i,j)}, \theta_\star\rangle \\ & + LSm\left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ , \end{aligned}$$

which gives us

$$R_{\frac{1+\xi}{2}}(T) \leq \sum_{t=1}^{T}\sum_{k=1}^{m} 4\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}} + LSm\left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ .$$

Let us bound the first term with the double sum as it is done in [2, 30]:

$$\sum_{t=1}^{T}\sum_{k=1}^{m} 4\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}}$$

$$
\begin{aligned}
&\leq \sum_{t=1}^{T}\sum_{k=1}^{m} \min\left(2LS, 4\sqrt{2\beta_t(\delta)}\|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}}\right) \\
&\leq \sum_{t=1}^{T}\sum_{k=1}^{m} 4\sqrt{2\beta_t(\delta)} \min\left(LS, \|z_t^{E[k]}\|_{\mathbf{A}_{t,k-1}^{-1}}\right) \\
&\leq \sqrt{Tm \times 32\beta_T(\delta) \sum_{t=1}^{T}\sum_{k=1}^{m} \min\left((LS)^2, \|z_t^{E[k]}\|^2_{\mathbf{A}_{t,k-1}^{-1}}\right)} \\
&\leq \sqrt{32Tm\beta_T(\delta) \sum_{t=1}^{T}\sum_{k=1}^{m} \max(2,(LS)^2) \log\left(1 + \|z_t^{E[k]}\|^2_{\mathbf{A}_{t,k-1}^{-1}}\right)} \qquad (5.17) \\
&= \sqrt{32Tm\beta_T(\delta) \max(2,(LS)^2) \sum_{t=1}^{T}\sum_{k=1}^{m} \log\left(1 + \|z_t^{E[k]}\|^2_{\mathbf{A}_{t,k-1}^{-1}}\right)} \\
&\leq \sqrt{32Tm\beta_T(\delta) \max(2,(LS)^2) d^2 \log\left(1 + \frac{TmL^2/\lambda}{d^2}\right)} \qquad (5.18) \\
&\leq \sqrt{32Tmd^2 \max(2,(LS)^2) \log\left(1 + \frac{TmL^2/\lambda}{d^2}\right)} \sqrt{\beta_T(\delta)} \ ,
\end{aligned}
$$

where (5.17) uses the fact that for all $a, x \geq 0$, $\min(a, x) \leq \max(2, a)\log(1 + x)$, (5.18) uses the fact that $\sum_{t=1}^{T}\sum_{k=1}^{m} \log\left(1 + \|z_t^{E[k]}\|^2_{\mathbf{A}_{t,k-1}^{-1}}\right) \leq d^2 \log\left(1 + \frac{TmL^2/\lambda}{d^2}\right)$ from Lemma 19.4 in [59].

We get the final bound by noticing that

$$
\sqrt{\beta_T(\delta)} \leq \left(\sigma\sqrt{d^2 \log\left(\frac{1 + TmL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S\right) \ .
$$

□

One can notice that the first term of the regret-bound matches the one of a standard linear bandit that pulls sequentially Tm edge-arms. The second term captures the cost of parallelizing m draws of edge-arms per round. Indeed, the intuition behind this term is that the couple (x_t, x'_t) chosen at round t is relevant to pull the $(tm + 1)$-th edge-arm but not necessarily the other $(m - 1)$ edge-arms that are pulled in parallel. This is because the reward associated with the $(tm + 1)$-th edge-arms could have led to change the central entity choice if it had been done sequentially. In [30], they characterize this phenomenon and show that this potential change in choice occurs less and less often as we pull arms and get rewards, hence the dependence in $O(\log(Tm))$.

5.1.3 Improved algorithm and analysis of the regret

We address the problem of designing an improved version of the proposed algorithm using the idea presented in Algorithm 6 that improves the approximation ratio.

Let us recall that in the previous section in Algorithm 9, the central entity chooses the couple (x_t, x'_t) such that

$$(x_t, x'_t) = \underset{(x,x')\in\mathcal{X}^2}{\arg\max} \max_{\theta\in C_{t-1}(\delta)} \langle z_{xx'} + z_{x'x}, \theta\rangle \ ,$$

which maximize the optimistic reward obtained between two nodes if the central entity were able to allocate x_t to one node and x'_t to the second. This strategy being optimal locally but complicated by considering the dependencies between edges, the central entity could take into consideration the edge arms of the form z_{xx} and $z_{x'x'}$ that are created when allocating the graph node using only two node-arm x and x'. This idea follows the one presented in Algorithm 6 where we recall with different notations that the couple $(\tilde{x}_\star, \tilde{x}'_\star)$ chosen to allocate the nodes of the graph are such that

$$(\tilde{x}_\star, \tilde{x}'_\star) = \underset{(x,x')\in\mathcal{X}^2}{\arg\max} \langle m_{1\to 2}\cdot z_{xx'} + m_{2\to 1}\cdot z_{x'x} + m_1 z_{xx} + m_2 z_{x'x'}, \theta_\star\rangle \ .$$

As in the previous section, we do not have access to $\theta_\star$, we use the principle of optimism to find at each round the couple $(\tilde{x}_t, \tilde{x}'_t)$ such that

$$(\tilde{x}_t, \tilde{x}'_t) = \underset{(x,x')\in\mathcal{X}^2}{\arg\max} \max_{\theta\in C_{t-1}(\delta)} \langle m_{1\to 2}\cdot z_{xx'} + m_{2\to 1}\cdot z_{x'x} = m_1 z_{xx} + m_2 z_{x'x'}, \theta\rangle \ . \tag{5.19}$$

Here, instead of maximizing the local reward one can get between two nodes, the central entity maximizes the global optimistic reward that one would obtain when allocating only two arms $(x, x') \in \mathcal{X}^2$ in the graph. This strategy is described in Algorithm 10.

Before stating the guarantees on the α-regret, we recall that we defined in Chapter 3 the quantity $\Delta \geq 0$ to be the expected reward difference of allocating $(\tilde{x}_\star, \tilde{x}'_\star)$ instead of $(x_\star, x'_\star)$,

$$\begin{aligned}\Delta = \langle & m_{1\to 2}\big(z_{\tilde{x}_\star\tilde{x}'_\star} - z_{x_\star x'_\star}\big) + m_{2\to 1}\big(z_{\tilde{x}'_\star\tilde{x}_\star} - z_{x'_\star x_\star}\big) \\ & + m_1(z_{\tilde{x}_\star\tilde{x}_\star} - z_{x_\star x_\star}) + m_2\big(z_{\tilde{x}'_\star\tilde{x}'_\star} - z_{x'_\star x'_\star}\big), \theta_\star\rangle \ .\end{aligned}$$

The new guarantees that we get on the α-regret of Algorithm 10 are stated in the following theorem.

Theorem 5.2. *Given the Graphical Bilinear Bandits problem defined as in Section 1.2, let ξ be defined as in Theorem 5.1, let $0 \leq \epsilon \leq \frac{1}{2}$ be a problem dependent parameter that measures the gain of optimizing on the suboptimal and unwanted arms defined as:*

$$\epsilon = \frac{\Delta}{\sum_{(i,j)\in E}\langle z_\star^{(i,j)}, \theta_\star\rangle} \ ,$$

and set $\alpha = \frac{1+\xi}{2} + \epsilon$ where $\alpha \geq 1/2$ by construction, then the α-regret of Algorithm 10 satisfies

Algorithm 10: Improved OFUL for Graphical Bilinear Bandits

Input : graph $\mathcal{G} = (V, E)$, node-arm set $\mathcal{X}$
$(V_1, V_2) =$ Approx-MAX-CUT$(\mathcal{G})$;
$m_{1\to 2} = |\{(i,j) \in E | i \in V_1 \wedge j \in V_2\}|$;
$m_{2\to 1} = |\{(i,j) \in E | i \in V_2 \wedge j \in V_1\}|$;
$m_1 = |\{(i,j) \in E | i \in V_1 \wedge j \in V_1\}|$;
$m_2 = |\{(i,j) \in E | i \in V_2 \wedge j \in V_2\}|$;
for $t = 1$ *to* T **do**
  $\left(\tilde{x}_t, \tilde{x}'_t, \tilde{\theta}_{t-1}\right) =$
  $\arg\max_{(x,x',\theta)\in\mathcal{X}^2\times C_{t-1}} \langle m_{1\to 2} \cdot z_{xx'} + m_{2\to 1} \cdot z_{x'x} + m_1 \cdot z_{xx} + m_2 \cdot z_{x'x'}, \theta\rangle$;
  $x_t^{(i)} = \tilde{x}_t$ for all i in V_1;
  $x_t^{(i)} = \tilde{x}'_t$ for all i in V_2;
  Obtain for all (i,j) in E the rewards $y_t^{(i,j)}$;
  Compute $\hat{\theta}_t$ as in (5.4)
end
return $\hat{\theta}_t$

$$R_\alpha(T) \leq \tilde{O}\left(\left(\sigma d^2 + S\sqrt{\lambda}\right)\sqrt{Tm\max(2,(LS)^2)}\right) + LSm\left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ ,$$

where $\tilde{O}$ hides the logarithmic factors.

Proof. We can write the regret $R(T)$ as in the proof of Theorem 5.1:

$$\begin{aligned} R(T) &= \sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star\right\rangle \\ &\quad + \sum_{t=1}^{T} \mathbb{1}[D_t^c] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star\right\rangle \\ &\leq \underbrace{\sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \left\langle z_t^{(i,j)}, \theta_\star\right\rangle}_{(a)} + \underbrace{LSm \sum_{t=1}^{T} \mathbb{1}[D_t^c]}_{(b)} \ . \end{aligned}$$

Here, (b) doesn't change, we thus only focus on deriving (a).

$$
\begin{aligned}
(a) &= \sum_{t=1}^{T} \mathbb{1}[D_t] \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \\
&\leq \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_\star^{(i,j)}, \theta_\star\rangle - \langle z_t^{(i,j)}, \theta_\star\rangle \qquad \text{(where } \mathbb{1}[D_t] \leq 1\text{)} \\
&= \underbrace{\sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_{1\to 2} + m_{2\to 1}}{m} \langle z_\star^{(i,j)}, \theta_\star\rangle}_{(a_1)} + \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1 + m_2}{m} \langle z_\star^{(i,j)}, \theta_\star\rangle \\
&\quad - \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \theta_\star\rangle \ .
\end{aligned}
$$

We have

$$
\begin{aligned}
(a_1) &= \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{2m_{1\to 2}}{m} \langle z_\star^{(i,j)}, \theta_\star\rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2m_{1\to 2}}{m} \langle z_\star^{(i,j)} + z_\star^{(j,i)}, \theta_\star\rangle \\
&\leq \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2m_{1\to 2}}{m} \langle z_{x_\star x'_\star} + z_{x'_\star x_\star}, \theta_\star\rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2}{m} \langle m_{1\to 2} \cdot z_{x_\star x'_\star} + m_{2\to 1} \cdot z_{x'_\star x_\star}, \theta_\star\rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2}{m} \langle m_{1\to 2} \cdot z_{x_\star x'_\star} + m_{2\to 1} \cdot z_{x'_\star x_\star} + m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star\rangle \\
&\quad - \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2}{m} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star\rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j\in\mathcal{N}_i \\ j>i}} \frac{2}{m} \langle m_{1\to 2} \cdot z_{\tilde{x}_\star \tilde{x}'_\star} + m_{2\to 1} \cdot z_{\tilde{x}'_\star \tilde{x}_\star} + m_1 \cdot z_{\tilde{x}_\star \tilde{x}_\star} + m_2 \cdot z_{\tilde{x}'_\star \tilde{x}'_\star}, \theta_\star\rangle
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j \in \mathcal{N}_i \\ j>i}} \frac{2}{m} \Delta - \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{\substack{j \in \mathcal{N}_i \\ j>i}} \frac{2}{m} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \\
&= \sum_{t=1}^{T} \langle m_{1\to 2} \cdot z_{\tilde{x}_\star \tilde{x}'_\star} + m_{2\to 1} \cdot z_{\tilde{x}'_\star \tilde{x}_\star} + m_1 \cdot z_{\tilde{x}_\star \tilde{x}_\star} + m_2 \cdot z_{\tilde{x}'_\star \tilde{x}'_\star}, \theta_\star \rangle - \Delta \\
&\quad - \sum_{t=1}^{T} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \\
&\leq \sum_{t=1}^{T} \left\langle m_{1\to 2} \cdot z_{\tilde{x}_t \tilde{x}'_t} + m_{2\to 1} \cdot z_{\tilde{x}'_t \tilde{x}_t} + m_1 \cdot z_{\tilde{x}_t \tilde{x}_t} + m_2 \cdot z_{\tilde{x}'_t \tilde{x}'_t}, \tilde{\theta}_{t-1,m} \right\rangle - \Delta \\
&\quad - \sum_{t=1}^{T} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \qquad \text{(w.p } 1-\delta\text{)} \\
&= \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} \rangle - \sum_{t=1}^{T} \Delta - \sum_{t=1}^{T} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \ .
\end{aligned}
$$

By plugging the last upper bound in (a) and with probability $1-\delta$, we have,

$$
\begin{aligned}
(a) &\leq \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} \rangle - \sum_{t=1}^{T} \Delta - \sum_{t=1}^{T} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \\
&\quad + \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1 + m_2}{m} \langle z_\star^{(i,j)}, \theta_\star \rangle - \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \theta_\star \rangle \\
&= \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} - \theta_\star \rangle - \sum_{t=1}^{T} \Delta - \sum_{t=1}^{T} \langle m_1 \cdot z_{x_\star x_\star} + m_2 \cdot z_{x'_\star x'_\star}, \theta_\star \rangle \\
&\quad + \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1 + m_2}{m} \langle z_\star^{(i,j)}, \theta_\star \rangle \\
&= \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} - \theta_\star \rangle - \sum_{t=1}^{T} \Delta - \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1}{m} \xi_{x_\star} \langle z_\star^{(i,j)}, \theta_\star \rangle \\
&\quad + \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_2}{m} \xi_{x'_\star} \langle z_\star^{(i,j)}, \theta_\star \rangle + \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1 + m_2}{m} \langle z_\star^{(i,j)}, \theta_\star \rangle \\
&\leq \sum_{t=1}^{T} \sum_{(i,j)\in E} \langle z_t^{(i,j)}, \tilde{\theta}_{t-1,m} - \theta_\star \rangle - \sum_{t=1}^{T} \Delta - \sum_{t=1}^{T} \sum_{(i,j)\in E} \frac{m_1 + m_2}{m} \xi \langle z_\star^{(i,j)}, \theta_\star \rangle
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{t=1}^{T}\sum_{(i,j)\in E}\frac{m_1+m_2}{m}\langle z_\star^{(i,j)},\theta_\star\rangle \\
&=\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle-\sum_{t=1}^{T}\Delta+\sum_{t=1}^{T}\sum_{(i,j)\in E}\frac{m_1+m_2}{m}(1-\xi)\langle z_\star^{(i,j)},\theta_\star\rangle \\
&=\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle-\sum_{t=1}^{T}\sum_{(i,j)\in E}\epsilon\langle z_\star^{(i,j)},\theta_\star\rangle \\
&\quad+\sum_{t=1}^{T}\sum_{(i,j)\in E}\frac{m_1+m_2}{m}(1-\xi)\langle z_\star^{(i,j)},\theta_\star\rangle \\
&=\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle+\sum_{t=1}^{T}\sum_{(i,j)\in E}\left[\frac{m_1+m_2}{m}(1-\xi)-\epsilon\right]\langle z_\star^{(i,j)},\theta_\star\rangle\ .
\end{aligned}
$$

By plugging (a) in the regret and with probability $1-\delta$, we have,

$$
\begin{aligned}
R(T)\leq&\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle+\sum_{t=1}^{T}\sum_{(i,j)\in E}\left[\frac{m_1+m_2}{m}(1-\xi)-\epsilon\right]\langle z_\star^{(i,j)},\theta_\star\rangle \\
&+LSm\sum_{t=1}^{T}\mathbb{1}[D_t^c]\ ,
\end{aligned}
$$

which gives,

$$
\begin{aligned}
R(T)-\sum_{t=1}^{T}\sum_{(i,j)\in E}\left[\frac{m_1+m_2}{m}(1-\xi)-\epsilon\right]\langle z_\star^{(i,j)},\theta_\star\rangle\leq&\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle \\
&+LSm\sum_{t=1}^{T}\mathbb{1}[D_t^c]\ .
\end{aligned}
$$

Thus,

$$
R_{1-\left[\frac{m_1+m_2}{m}(1-\xi)-\epsilon\right]}(T)\leq\sum_{t=1}^{T}\sum_{(i,j)\in E}\langle z_t^{(i,j)},\tilde{\theta}_{t-1,m}-\theta_\star\rangle+LSm\sum_{t=1}^{T}\mathbb{1}[D_t^c]\ .
$$

The upper bound of the right hand terms follows exactly what we have already done for Theorem 5.1 by applying the upper bounds (5.12) and (5.16).

Moreover,

$$1 - \left[\frac{m_1 + m_2}{m}(1-\xi) - \epsilon\right] \geq 1 - \left[\frac{(1-\xi)}{2} - \epsilon\right] = \frac{(1+\xi)}{2} + \epsilon \ .$$

□

Here one can see that the improvement happens in the α of the α-regret. In the next section, we confirm this results through experiments.

5.2 Numerical experiments

In Chapter 3, we run experiments on α and how it can varying according to the graph and the problem parameters. In this subsection, we only focus on the impact on the regret. We design an experiment that compares in practice the performance of Algorithm 9 and Algorithm 10 with the Explore-Then-Commit (ETC) algorithm by using the exploration strategy designed in Chapter 4 during the exploration phase, and by allocating the nodes in V_1 and V_2 with the best estimated couple $(x, x') = \arg\max_{(x,x')} \langle z_{xx'} + z_{x'x}, \hat{\theta}_t \rangle$ during the commit phase. However, since the algorithms that we presented in this section have guarantees on α-regrets with different α, we plot the fraction of the optimal global reward for each iteration.

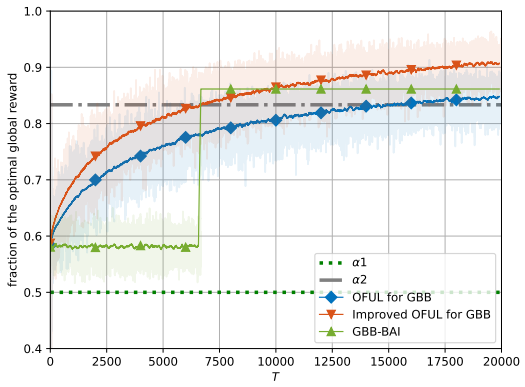


Figure 5.1: Fraction of the optimal global reward obtained at each round by applying the Algorithm 9, Algorithm 10 and the Explore-Then-commit algorithm (here named GBB-BAI) using the exploration strategy in Chapter 4. We use a complete graph of 5 nodes, we run the experiment on 5 different matrices as in Figure 3.1 with $\zeta = 0$ and run it 10 different times to plot the average fraction of the global reward. We set the confidence $\delta = 0.001$.

As in Chapter 3, we observe a clear improvement when choosing at each round t the couple of arms $(\tilde{x}_t, \tilde{x}'_t)$ instead of (x_t, x'_t).

5.3 Conclusion and perspectives

In this chapter, we presented the first regret-based algorithms for the stochastic graphical bilinear bandits problem with guarantees on the α-regret with $\alpha \geq 1/2$. We also showed experimentally that our algorithm achieves a better performance than Explore-Then-Exploit on our synthetic datasets. One could also study this problem in the adversarial setting, in particular adapting adversarial linear bandit algorithms to our case. Finally, our setting could be extended to the case where each agent has its own reward matrix.

6 Conclusion & perspectives

6.1 Summary of the results

In this thesis we introduced a new model that we named *Graphical Bilinear Bandits* that models centralized multi-agent problems where pairwise interactions exist between agents.

- In Chapter 3, we have highlighted the fact that the learner faced an underlying optimization problem that is NP-Hard no matter which goal the learner wanted to reach. Hence we have proposed an α-approximation algorithm with $\alpha \geq 1/2$ which only required to find the couple of arms $(x_\star, x'_\star)$ to return the α-approximate solution. Then we have refined this approximation-parameter with respect to problem dependent parameters based on the graph structure and a property of the parameter matrix $\mathbf{M}_\star$.

- In Chapter 4, given the α-approximation algorithm designed in Chapter 3, we have presented a pure-exploration algorithm that allowed the learner to construct an estimate $\hat{\mathbf{M}}$ that was statistically efficient in terms of optimal design. Indeed, the problem of finding within a minimum number of rounds the best couple $(x_\star, x'_\star)$ used in the α-approximation algorithm came down to finding the G-optimal design, also called G-allocation in the bandit literature. Solving this problem in the graphical bilinear bandits implied dealing with an additional constraint. That was why we have presented an algorithm that respected this constraint and that used randomized sampling to construct the estimate $\hat{\mathbf{M}}$. Our theoretical results have revealed a term that depended on the graph structure, so we showed the impact of the graph in our results.

- Finally, in Chapter 5, we have capitalized on the α-approximation algorithm given in chapter 3 and applied the principle of optimism in the face of uncertainty to design regret-based algorithms that achieved a sublinear α-regret in T where $\alpha \geq 1/2$. Furthermore, we have presented experimentally the performance of the proposed algorithms and used compared with an Explore-Then-Commit algorithm relying on the pure-exploration algorithm presented in chapter 4.

6.2 Perspective and future works

This thesis aimed to introduce the new graphical bilinear bandit setting and to provide the first solutions to common problems posed in the bandit literature. A lot of other approaches and modifications can be considered. We state two of them in the following.

Different parameter matrices $\mathbf{M}_\star^{(i,j)}$ for each edge $(i,j) \in E$. While dealing with a common parameter matrix $\mathbf{M}_\star$ for all the edges $(i,j) \in E$ was convenient for aggregating the rewards and constructing a common estimate $\hat{\mathbf{M}}$ for all the agents, when the rewards $y_t^{(i,j)}$ are defined with different matrices $\mathbf{M}_\star^{(i,j)}$, the problem becomes more complicated. Indeed consider the setting where the reward $y_t^{(i,j)}$ is defined as follows for each $(i,j) \in E$:

$$y_t^{(i,j)} = x_t^{(i)\top} \mathbf{M}_\star^{(i,j)} x_t^{(j)} + \eta_t^{(i,j)} \quad ,$$

where $\mathbf{M}_\star^{(i,j)}$ are unknown parameter matrices and $\eta_t^{(i,j)}$ σ-sub-gaussian random variables.

This setting is relevant when the agents do not have the same interactions between each of their neighbors, and thus not the same reward function.

> **Open questions:** In the context of pure exploration, how does the stopping condition change? Is there a sampling strategy for each agent such that estimates $\hat{\mathbf{M}}^{(i,j)}$ are constructed by satisfying an optimal design criterion?

Decentralized setting. When agents are controlled by a central entity, it is possible to aggregate the different rewards and to build a common estimate $\hat{\mathbf{M}}$ of $\mathbf{M}_\star$. Moreover, we have seen that the different objectives that appear are relative to the edge-arms and not directly to the node-arms selected by each agent. Indeed, this is due to the fact that we can express the graphical bilinear bandit as linear bandits at the edge level. This particular aspect makes the decentralized framework a bit tricky because coordinating two agents without communication to respectively draw the node-arms that will build the wanted edge-arm becomes even more complicated.

However other problem arise even if the coordination problem is solved. For example, in the best arm identification problem, we have already designed a sampling procedure that can be executed in parallel during a round, hence a decentralized choice for each agent. However, the stopping condition depends on the estimate $\hat{\mathbf{M}}$ constructed with the edge-arms during the learning procedure, but when the agents do not communicate, this estimate cannot be constructed. This is because an agent only knows which node arm it is pulling and observes the reward. However, the reward is linear with respect to the associated edge-arm and the agent does not have access to this edge-arm since it is constructed with its node arm but also with that of its neighbors (to which it does not have access).

> **Open questions:** In the fully decentralized setting (without communication), what kind of algorithms can we design to take advantage of the (bi-)linear bandit setting? If we allow communication, how can we adapt the proposed algorithms and what is the trade-off between the amount of communications and their performances?

A Refined bounds for randomized experimental design

This appendix contains the paper *"Refined bounds for randomized experimental design"*, Neurips 2019 Workshop "ML with Guarantees", G. Rizk, I. Colin, A. Thomas and Moez Draief.

Contents

Experimental design is an approach for selecting samples among a given set so as to obtain the best estimator for a given criterion. In the context of linear regression, several optimal designs have been derived, each associated with a different criterion: mean square error, robustness, *etc*. Computing such designs is generally an NP-hard problem and one can instead rely on a convex relaxation that considers probability distributions over the samples. Although greedy strategies and rounding procedures have received a lot of attention, straightforward sampling from the optimal distribution has hardly been investigated. In this paper, we propose theoretical guarantees for randomized strategies on E and G-optimal design. To this end, we develop a new concentration inequality for the eigenvalues of random matrices using a refined version of the intrinsic dimension that enables us to quantify the performance of such randomized strategies. Finally, we evidence the validity of our analysis through experiments, with particular attention on the G-optimal design applied to the best arm identification problem for linear bandits.

A.1 Introduction

Experimental designs consist in the selection of the best samples or *experiments* for the estimation of a given quantity. A well-known and extensively studied example is the one of the ordinary least squares (OLS) estimator in the linear regression setting. The OLS estimator being unbiased,

which experiments must be chosen in a fixed pool of experiments so as to minimize its variance? In the multi-dimensional case, this is done by minimizing a scalar function of its covariance matrix and several approaches have been considered such as the minimization of the determinant, the trace or the spectral norm, respectively denoted D, A and E-optimal design (see *e.g.*, [76, 83]). (See Appendix 1, for more details on experimental design)

E-optimal design has been exploited in practical settings such as for biological experiments [40] or for treatment versus control comparisons where useful statistical interpretations have been derived, see *e.g.*, [69, 80]. Another criterion, known as G-optimal design and which minimizes the worst predicted variance, has recently been investigated in the context of *best arm identification* in linear bandits [90, 92, 106] where one is interested in finding the experiment with maximum linear response.

The optimization problems associated to the aforementioned optimal designs (E, A, D, G) are known to be NP-hard [32, 104]. The two common approaches have been to resort to greedy strategies or convex relaxations. A greedy strategy iteratively finds the best experiment whereas solving a convex relaxation returns a discrete probability distribution over the samples. On the one hand, performance guarantees for greedy strategies have been obtained by exploiting supermodularity and approximate supermodularity properties of the different criteria [29, 82, 90]. On the other hand, for performance guarantees of randomized optimal designs, only the randomized A-optimal design has been thetoretically studied with bounds on the mean square error of the associated estimator ([103]).

We propose in this paper to fill the gap concerning randomized E and G-optimal designs. More precisely we study their theoretical validity by providing finite-sample confidence bounds and show with experiments that they are worth being considered in practice. The paper is organized as follows. Section A.2 defines the main notations and recalls the problem of experimental design as well as the different optimal criteria. Section A.3 presents the main results of this paper for the random strategies of E and G-optimal designs. Finally, the last section shows empirical results of the studied random strategies and an application to the best arm identification problem in linear bandits.

A.2 Preliminaries

Definitions and notations

Throughout the paper, we use small bold letters for vectors (*e.g.*, $\mathbf{x}$) and capital bold letters for matrices (*e.g.*, $\mathbf{X}$). For any $d > 0$ and any vector $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|$ will denote the usual ℓ_2-norm of $\mathbf{x}$. For any square matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$, we denote as $\|\mathbf{X}\|$ the spectral norm of $\mathbf{X}$, that is $\|\mathbf{X}\| \triangleq \sup_{\mathbf{y}:\|\mathbf{y}\|=1} \|\mathbf{X}\mathbf{y}\|$. We let $\lambda_{\min}(\mathbf{X})$ be the smallest eigenvalue of $\mathbf{X}$. For any $1 \leq i, j \leq d$, any $\mathbf{x} \in \mathbb{R}^d$ and any matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$, $[\mathbf{x}]_i$ denotes the i-th coordinate of vector $\mathbf{x}$, $[\mathbf{X}]_i$ the vector of the i-th row and $[\mathbf{X}]_{ij}$ the value at the i-th row and j-th column. Finally, we denote by $\mathbf{S}_d^+$ the cone of all $d \times d$ positive semi-definite matrices and by $\Delta_d \triangleq \{\mu \in [0,1]^d, \sum_{i=1}^d [\mu]_i = 1\}$ the simplex in $\mathbb{R}^d$.

Experimental design for linear regression

Given $\mathbf{X} \in \mathbb{R}^{K \times d}$ a matrix of K experiments[1] and $\mathbf{y} \in \mathbb{R}^K$ a vector of K measurements, it is assumed that there exists an unknown parameter $\theta_\star \in \mathbb{R}^d$ such that for all $k \in \{1, \dots, K\}$, $[\mathbf{y}]_k = \theta_\star^\top \mathbf{x}_k + \varepsilon_k$ where $\mathbf{x}_k = [\mathbf{X}]_k$ and $\varepsilon_1, \dots, \varepsilon_K$ are independent Gaussian random variables with zero mean and variance σ^2. The ordinary least squares (OLS) estimator of the parameter $\theta_\star$ is given by $\hat{\theta} = \arg\min_\theta \|\mathbf{y} - \mathbf{X}\theta\|_2^2 = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$.[2] This estimator is unbiased and has a covariance matrix $\mathbf{\Sigma}^{-1} = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$.

Experimental design [76] consists in estimating $\hat{\theta}$ by selecting only the experiments that are the most statistically efficient to reduce the variance.

More formally, let n be the total number of selected experiments and for all $k \in \{1, \dots, K\}$, let n_k be the number of times $\mathbf{x}_k$ is chosen. We have $n_k \geq 0$ and $\sum_{i=1}^K n_k = n$. The covariance matrix obtained with such a design can be written as $\mathbf{\Sigma}_D^{-1} = \sigma^2 (\sum_{k=1}^K n_k \mathbf{x}_k \mathbf{x}_k^\top)^{-1}$. The Loewner order on $\mathbf{S}_d^+$ being only a partial order, minimizing $\mathbf{\Sigma}_D^{-1}$ over the cone $\mathbf{S}_d^+$ is an ill-posed problem. An optimal design is thus defined thanks to scalar properties of a matrix in $\mathbf{S}_d^+$, *i.e.*, as a solution of $\min_{n_1, \dots, n_K} f(\mathbf{\Sigma}_D^{-1})$ where $f : \mathbf{S}_d^+ \to \mathbb{R}$. The two criterion f we study in this paper are:

- E-**optimality** : $f_E(\mathbf{\Sigma}_D^{-1}) = \|\mathbf{\Sigma}_D^{-1}\|$. The E-optimal design minimizes the maximum eigenvalue of $\mathbf{\Sigma}_D^{-1}$. Geometrically it minimizes the variance ellipsoid in the direction of its diameter only.
- G-**optimality** : $f_G(\mathbf{\Sigma}_D^{-1}) = \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top \mathbf{\Sigma}_D^{-1} \mathbf{x}$. The G-optimal design minimizes the worst possible predicted variance.

Those two optimality criteria are NP-hard optimization problems [32, 104]. However approximate solutions can be found in polynomial time by relying on greedy strategies or by relaxing the problem and looking for proportions $\mu_k \in [0, 1]$ instead of integers n_k. By letting $\mu_k = n_k / n$, the covariance matrix $\mathbf{\Sigma}_D^{-1}$ can be written as $\mathbf{\Sigma}_D^{-1} = \sigma^2 / n \cdot (\sum_{k=1}^K \mu_k \mathbf{x}_k \mathbf{x}_k^\top)^{-1}$ and it leads us to the convex optimiation problem $\min_{\mu_1, \dots, \mu_K \in [0,1]} f(\mathbf{\Sigma}_D^{-1})$ which returns a discrete probability distribution over the samples. For more details on experimental design and optimal design criteria see [20, 76].

A.3 Convergence analysis

In this section, we analyze the behavior of random sampling along the distribution associated to the convex relaxation discussed in Section A.2, for E and G-optimal designs. Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_K\} \subseteq \mathbb{R}^d$ be the set of experiments and let $\mu_E^\star$ and $\mu_G^\star$ be the optimal distributions in Δ_K associated to the convex relaxation of such designs. For any $\mu \in \Delta_K$, we denote as $\mathbf{M}(\mu)$ the matrix $\mathbf{M}(\mu) \triangleq \sum_{k=1}^K \mu_k \mathbf{x}_k \mathbf{x}_k^\top$ and $f_{G,n}^\star \triangleq f_G\big((n\mathbf{M}(\mu_G^\star))^{-1}\big)$ as the objective at the optimum $\mu_G^\star$ for a sample size n.

[1] In the remaining of this paper, we consider a finite set of experiments; some results could be easily transposed to a continuous setting.

[2] We assume that the experiments span $\mathbb{R}^d$ so that the matrix $\mathbf{X}^\top \mathbf{X}$ is non singular. If this is not the case then we may project the data onto a lower dimensional space.

Theorem A.1. *Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ be a set of experiments and let $\mu_E^\star$ be the solution of the relaxation of the E-optimal design. Let $0 \leq \delta \leq 1$ and let n such that*

$$n \geq 2L \left\| \mathbf{M}(\mu_E^\star)^{-1} \right\| \log(d/\delta),$$

where $L = \max_{x \in \mathcal{X}} \|x\|^2$. Then, with probability at least $1 - \delta$, one has

$$f_E\big(\mathbf{S}_{E,n}^{-1}\big) \leq \left(1 + \frac{1}{\sqrt{\dfrac{n}{2L\|\mathbf{M}(\mu_E^\star)^{-1}\| \log(d/\delta)}} - 1} \right) f_{E,n}^\star,$$

where $\mathbf{S}_{E,n}$ is the sum of n i.i.d. random matrices drawn from $\mu_E^\star$.

Similarly, let $\mu_G^\star$ be the solution of the relaxed G-optimal design and $\mathbf{S}_{G,n}$ the associated sample sum. One has, with probability at least $1 - 2\delta$,

$$f_G\big(\mathbf{S}_{G,n}^{-1}\big) \leq \left(1 + \frac{L}{d} \|\mathbf{M}(\mu_G^\star)^{-1}\|^2 \sqrt{\frac{2\sigma^2}{n} \log(d/\delta)} + o\left(\frac{1}{\sqrt{n}}\right) \right) f_{G,n}^\star,$$

with $\sigma^2 \triangleq L^2 \sum_{k=1}^K [\mu_G^\star]_k (1 - [\mu_G^\star]_k)$.

Theses results recover the $O(1/\sqrt{n})$ that one would expect. In addition, this confirms that the randomized approach asymptotically converges toward the true optimum, which is not the case—theoretically—for the greedy strategy. Finally, let us note that the $o(1/\sqrt{n})$ in the G-optimal design rate depends on the interaction between Hoeffding for $\lambda_{\min}(\mathbf{S}_n)$ and Bennett for $\|\mathbf{S}_n - \mathbb{E}\mathbf{S}_n\|$. We refer the reader to the supplementary material in Appendix A.6 for the full bound.

A refined approach of the dimension

In this section, we introduce two quantities, derived from the concept of *intrinsic dimension* [54, 68], that allow us to refine the convergence rate for G-optimal design. We recall the definition of intrinsic dimension.

Definition A.1 (Intrinsic dimension)**.** *Let $d > 0$ and $\mathbf{S} \in \mathbb{R}^{d \times d}$ be a positive semi-definite matrix. The intrinsic dimension of $\mathbf{S}$, denoted $\operatorname{intdim}(\mathbf{S})$, is defined as follows:*

$$\operatorname{intdim}(\mathbf{S}) \triangleq \frac{\operatorname{tr}(\mathbf{S})}{\|\mathbf{S}\|} \leq d.$$

Using this definition, one may alter the concentration results on the spectral norm, by replacing the dimension d by $2 \times \operatorname{intdim}(\mathbb{E}\mathbf{S}_n)$. For a matrix with eigenvalues decreasing fast enough, the improvement may be substantial—see [97, Ch. 7] and references therein for more details. The main drawback of this definition is that if eigenvalues are all of the same order of magnitude, one will not notice a sensible improvement; this is typically the case in G-optimal design as eigenvalues are designed to be large overall. We propose a refined version of the intrinsic dimension allowing improvements even with a narrow spectrum, in the form of 2 complementary quantities.

Definition A.2 (Upper and lower intrinsic dimension)**.** *Let $d > 0$ and $\mathbf{S} \in \mathbb{R}^{d\times d}$ be a positive semi-definite matrix. The upper and lower intrisic dimensions of $\mathbf{S}$, denoted $\mathrm{updim}(\mathbf{S})$ and $\mathrm{lowdim}(\mathbf{S})$ respectively, are defined as follows:*

$$\begin{cases} \mathrm{updim}(\mathbf{S}) & \triangleq & \dfrac{\mathrm{tr}\left(\mathbf{S} - \lambda_{\min}(\mathbf{S})\mathbf{I}\right)}{\|\mathbf{S}\| - \lambda_{\min}(\mathbf{S})} \\ \mathrm{lowdim}(\mathbf{S}) & \triangleq & \dfrac{\mathrm{tr}\left(\|\mathbf{S}\|\mathbf{I} - \mathbf{S}\right)}{\|\mathbf{S}\| - \lambda_{\min}(\mathbf{S})} = d - \mathrm{updim}(\mathbf{S}). \end{cases}$$

These new quantities use both the largest and the smallest eigenvalues to rescale the spectrum, which is of interest in our setting. Using this definition, one is able to formulate new concentration results on random matrices, including a concentration result on the lowest eigenvalue. In this particular case however, we are more interested in the potential speed up provided for the spectral norm, since it is the value controlling the slowest term in the G-optimal design error.

Theorem A.2. *Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ be a set of experiments and let $\mu^\star_G$ be the solution of the relaxation of the G-optimal design. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be n i.i.d. random matrices drawn according to $\mu^\star_G$ and $\mathbf{S}_n$ their sum. Let $\mathbf{V}$ be the covariance matrix of $\mathbf{X}_1$, that is $\mathbf{V} \triangleq \mathbb{E}\big[\mathbf{X}_1^2\big] - \mathbf{M}(\mu^\star_G)^2$ and let κ be its condition number.*

Let $0 \leq \delta \leq 1$ and let n such that

$$n \geq \left(\frac{4L^2}{9\|\mathbf{V}\|}\log\Big(\tilde{d}/\delta\Big)\right)$$

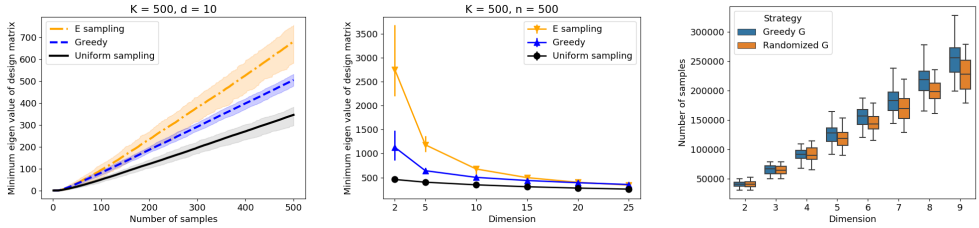
where $L = \max_{x\in\mathcal{X}} \|x\|^2$ and $\tilde{d}$ is defined by $\tilde{d} \triangleq \mathrm{updim}(\mathbf{V}) + \mathrm{lowdim}(\mathbf{V})e^{-n(1-\kappa^{-1})/16} < d$. Then, with probability at $1 - 2\delta$, one has

$$f_G\big(\mathbf{S}_{G,n}^{-1}\big) \leq \left(1 + \frac{L}{d}\|\mathbf{M}(\mu^\star_G)^{-1}\|^2\sqrt{\frac{4\sigma^2}{n}\log\Big(\tilde{d}/\delta\Big)} + o\Big(\frac{1}{\sqrt{n}}\Big)\right) f^\star_{G,n}.$$

We refer the reader to the supplementary material in Appendix A.6 for the proof of this result.

A.4 Experiments

In this section we compare the performances of randomized E and G-optimal designs against their greedy counterparts. We first show the behavior of the randomized E-optimal design on a synthetic data set. We then apply the randomized G-optimal design to the problem of best arm identification and compare it to the greedy approach used in [90]. We refer the reader to Appendix A.6 and A.6 for more details on best arm identification for linear bandits and on the experiments setting, respectively.



(a) Evolution of score with n (b) Evolution of score with d (c) Evolution of score with d

A.5 Conclusion

We have shown the convergence of randomized scheme for G and E-optimal criteria at a rate of $O(1/\sqrt{n})$. We also evidenced the dependence of the rate in a specific characteristic of the covariance matrix for the sampling. Empirically, the random sampling enjoys a favorable comparison with the greedy approach, even in the bandit application. One possible extension of this work could be to investigate the setting of batch or parallel bandits, using a random sampling to select a batch of arms before observing the rewards.

A.6 Proofs and details on experiments

Chernoff inequalities on matrices

Many concentration inequalities have been developped for bounding the deviation of a sum of i.i.d. random variables. In particular, Chernoff inequalities have been extensively studied and derived due to their exponential decay rate on tail distributions. Here we show how these bounds can be extended to random matrices (see *e.g.*, [96] for an introduction on that matter).

Additional notations

For any Hermitian matrices $\mathbf{X}, \mathbf{Y}$, we write $\mathbf{X} \preceq \mathbf{Y}$ if and only if the matrix $\mathbf{Y} - \mathbf{X}$ is positive semidefinite. Recall that for any Hermitian matrix $\mathbf{X}$, there exists a unitary matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$ such that $\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}^\top$. For such a matrix and for any function $f : \mathbb{R} \to \mathbb{R}$, we denote as $f(\mathbf{X})$ the extension of f to a Hermitian matrix, defined as follows:

$$f(\mathbf{X}) \triangleq \mathbf{P} \begin{pmatrix} f([\mathbf{D}]_{11}) & & \\ & \ddots & \\ & & f([\mathbf{D}]_{dd}) \end{pmatrix} \mathbf{P}^\top.$$

In particular, for any scalar $x \in \mathbb{R}$, we define $(x)_+ \triangleq \max(x, 0)$ so $(\mathbf{X})_+$ is the projection of $\mathbf{X}$ onto the positive semidefinite cone. We will use the exponential function for both scalars and matrices: for the sake of clarity, we denote as e^x the exponential of a scalar and $\exp(\mathbf{X})$ the exponential of a matrix. We will denote as $\mathrm{Sp}(\mathbf{X})$ the spectrum of $\mathbf{X}$, that is the set of all eigenvalues

associated to $\mathbf{X}$. The identity matrix and the zero matrix in dimension d are denoted $\mathbf{I}_d$ and $\mathbf{0}_d$, respectively; when clear from context, we drop the d index.

Useful lemmas

Before stating the concentration inequalities of interest, we need to state several useful lemmas. These lemmas are key for proving concentration of random matrices, as we need similar guarantees for matrix ordering ($\preceq$) than for scalar ordering ($\leq$). We first state two lemmas that will ensure order preserving under basic operations.

Lemma A.1 (Conjugation Rule)**.** *Let $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d\times d}$ be two Hermitian matrices, such that $\mathbf{M} \preceq \mathbf{N}$. Let $p > 0$ and let $\mathbf{Q} \in \mathbb{R}^{p\times d}$. Then, one has*

$$\mathbf{Q}\mathbf{M}\mathbf{Q}^\top \preceq \mathbf{Q}\mathbf{N}\mathbf{Q}^\top.$$

Proof. The proof is immediate when considering $\mathbf{Q}(\mathbf{N} - \mathbf{M})\mathbf{Q}^\top$ and using the definition of a positive semidefinite matrix. □

Lemma A.2 (Transfer Rule)**.** *Let $\mathbf{M} \in \mathbb{R}^{d\times d}$ be a Hermitian matrix and let $f, g : \mathbb{R} \to \mathbb{R}$ be such that, for any $x \in \mathrm{Sp}(\mathbf{M})$, $f(x) \leq g(x)$. Then, one has*

$$f(\mathbf{M}) \preceq g(\mathbf{M}).$$

Proof. Let $\mathbf{D}$ be the diagonal matrix in the spectral decomposition of $\mathbf{M}$. Since $f \leq g$ on $\mathrm{Sp}(\mathbf{M})$, one has $f(\mathbf{D}) \preceq g(\mathbf{D})$. The conjugation rule then allows us to conclude. □

Finally, we state two lemmas ensuring that two more complex operations ($\mathrm{tr}\exp$ and $\log$, respectively) preserve the order. Please note that this is usually not the case, even for operators that are monotone on $\mathbb{R}$—*e.g.*, the exponential does not preserve the order.

Lemma A.3 (Monotonicity of the trace of the exponential)**.** *Let $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d\times d}$ be two Hermitian matrices such that $\mathbf{M} \preceq \mathbf{N}$. Then for any non-decreasing function $\psi : \mathbb{R} \to \mathbb{R}$, one has:*

$$\mathrm{tr}\left(\psi(\mathbf{M})\right) \leq \mathrm{tr}\left(\psi(\mathbf{N})\right).$$

In particular,

$$\mathrm{tr}\exp(\mathbf{M}) \leq \mathrm{tr}\exp(\mathbf{N}).$$

Proof. Let $\lambda_1(\mathbf{M}) \geq \ldots \geq \lambda_d(\mathbf{M})$ and $\lambda_1(\mathbf{N}) \geq \ldots \geq \lambda_d(\mathbf{N})$ be the sorted eigenvalues of $\mathbf{M}$ and $\mathbf{N}$, respectively. Then, for $1 \leq i \leq d$, one can define an eigenvalue as follows:

$$\lambda_i(\mathbf{M}) = \max_{\mathbb{L}\subseteq\mathbb{R}^d:\dim\mathbb{L}=i} \min_{\mathbf{u}\in\mathbb{L}:\|\mathbf{u}\|=1} \mathbf{u}^\top\mathbf{M}\mathbf{u}.$$

Using the fact that $\mathbf{M} \preceq \mathbf{N}$, one can deduce that for any $1 \leq i \leq d$, $\lambda_i(\mathbf{M}) \leq \lambda_i(\mathbf{N})$.

Since ψ is a non-decreasing function on $\mathbb{R}$, one has that for any $1 \leq i \leq d$, $\psi\big(\lambda_i(\mathbf{M})\big) \leq \psi\big(\lambda_i(\mathbf{N})\big)$. Summing over the dimensions leads to the desired result. □

Lemma A.4 (Monotonicity of the logarithm)**.** *Let $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d\times d}$ be two positive definite matrices such that $\mathbf{M} \preceq \mathbf{N}$. Then one has:*

$$\log(\mathbf{M}) \preceq \log(\mathbf{N}).$$

Proof. We will first prove that for any $\gamma \in \mathbb{R}_+$, $(\mathbf{M} + \gamma\mathbf{I})^{-1} \succeq (\mathbf{N} + \gamma\mathbf{I})^{-1}$.

The facts that $\mathbf{M} \preceq \mathbf{N}$ and $\gamma \geq 0$ imply that $\mathbf{M} + \gamma\mathbf{I} \preceq \mathbf{N} + \gamma\mathbf{I}$. Using Lemma A.1, we obtain:

$$\mathbf{0} \prec (\mathbf{N} + \gamma\mathbf{I})^{-1/2}(\mathbf{M} + \gamma\mathbf{I})(\mathbf{N} + \gamma\mathbf{I})^{-1/2} \preceq \mathbf{I}.$$

Taking the inverse yields:

$$(\mathbf{N} + \gamma\mathbf{I})^{1/2}(\mathbf{M} + \gamma\mathbf{I})^{-1}(\mathbf{N} + \gamma\mathbf{I})^{1/2} \succeq \mathbf{I}.$$

Finally, applying again Lemma A.1 with $(\mathbf{N} + \gamma\mathbf{I})^{-1/2}$ yields:

$$(\mathbf{M} + \gamma\mathbf{I})^{-1} \succeq (\mathbf{N} + \gamma\mathbf{I})^{-1}.$$

Let us now focus on the main result. First recall that the logarithm of a positive scalar can be expressed using its integral representation, that is

$$\log x = \int_0^{+\infty} \left(\frac{1}{1+t} - \frac{1}{x+t} \right) \mathrm{d}t,$$

for any $x > 0$. Therefore, the logarithm of a matrix $\mathbf{X} \succ \mathbf{0}$ can be expressed similarly:

$$\log \mathbf{X} = \int_0^{+\infty} \left(\frac{1}{1+t}\mathbf{I} - (\mathbf{X} + t\mathbf{I})^{-1} \right) \mathrm{d}t.$$

In the beginning of the proof, we have shown that for any $\gamma \geq 0$, $(\mathbf{M}+t\mathbf{I})^{-1} \succeq (\mathbf{N}+t\mathbf{I})^{-1}$. Therefore, one has:

$$\frac{1}{1+\gamma}\mathbf{I} - (\mathbf{M} + \gamma\mathbf{I})^{-1} \preceq \frac{1}{1+\gamma}\mathbf{I} - (\mathbf{N} + \gamma\mathbf{I})^{-1},$$

and integrating over γ yields the final result. □

Chernoff inequalities

Let $n > 0$ and let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. positive semidefinite matrices, such that there exists $L > 0$ verifying:

$$\mathbf{0} \preceq \mathbf{X}_1 \preceq L\mathbf{I},$$

almost surely. Let us now consider the random matrix $\mathbf{S} = \sum_{i=1}^n \mathbf{X}_i$. In what follows, we will develop Chernoff bounds in order to control both $\|\mathbf{S}\|/\|\mathbb{E}\mathbf{S}\|$ and $\|\mathbf{S} - \mathbb{E}\mathbf{S}\|$.

In the scalar case, Chernoff's bounds for the sum of independent variables are based on the fact that the exponential converts a sum into a products, that is for n i.i.d. random variables $X_1, \dots, X_n$:

$$\mathbb{E}e^{\sum_{i=1}^n X_i} = \mathbb{E}\prod_{i=1}^n e^{X_i},$$

and then one uses the independence to pull the product out of the expectation. For two symmetric matrices $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d\times d}$ however, the relation $\exp(\mathbf{M}+\mathbf{N}) = \exp\mathbf{M}\exp\mathbf{N}$ does not hold in general—it holds if the matrices commute. Hopefully, the following theorem gives us a way to overcome this issue.

Theorem A.3. *Let $n, d > 0$ and let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be i.i.d. symmetric matrices in $\mathbb{R}^d$. Then, for any $t \in \mathbb{R}$, one has*

$$\mathbb{P}\left(\left\|\sum_{i=1}^n \mathbf{X}_i\right\| \geq t\right) \leq \inf_{\eta>0} e^{-\eta t}\operatorname{tr}\exp\left(\sum_{i=1}^n \log\mathbb{E}\exp(\eta\mathbf{X}_i)\right),$$

and similarly

$$\mathbb{P}\left(\lambda_{\min}\left(\sum_{i=1}^n \mathbf{X}_i\right) \leq t\right) \leq \inf_{\eta<0} e^{-\eta t}\operatorname{tr}\exp\left(\sum_{i=1}^n \log\mathbb{E}\exp(\eta\mathbf{X}_i)\right).$$

Proof. We start by the first inequality. Let $t \in \mathbb{R}$ and let $\eta > 0$. As in the scalar case, one has:

$$\mathbb{P}\left(\left\|\sum_{i=1}^n \mathbf{X}_i\right\| \geq t\right) = \mathbb{P}\left(e^{\eta\|\sum_{i=1}^n \mathbf{X}_i\|} \geq e^{\eta t}\right) \leq e^{-\eta t}\mathbb{E}e^{\eta\|\sum_{i=1}^n \mathbf{X}_i\|},$$

where the last inequality is an application of Markov's inequality. Using the fact that for a positive semidefinite matrix $\mathbf{X}$, $\|\mathbf{X}\| \leq \operatorname{tr}\mathbf{X}$, we obtain:

$$\mathbb{E}e^{\eta\|\sum_{i=1}^n \mathbf{X}_i\|} = \mathbb{E}\left\|\exp\left(\eta\sum_{i=1}^n \mathbf{X}_i\right)\right\| \leq \mathbb{E}\operatorname{tr}\exp\left(\eta\sum_{i=1}^n \mathbf{X}_i\right).$$

We will now use Lieb's Theorem [64], which states that for any symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, the mapping $\mathbf{M} \mapsto \operatorname{tr}\exp(\mathbf{A} + \log\mathbf{M})$ is concave on the cone of positive semidefinite matrices. This allows us to bound the above term as follows:

$$\mathbb{E}\operatorname{tr}\exp\left(\eta\sum_{i=1}^n \mathbf{X}_i\right) = \mathbb{E}\mathbb{E}\left[\operatorname{tr}\exp\left(\eta\sum_{i=1}^n \mathbf{X}_i\right)\middle|\mathcal{F}_{n-1}\right]$$

$$= \mathbb{E}\mathbb{E}\left[\operatorname{tr}\exp\left(\eta\sum_{i=1}^{n-1}\mathbf{X}_i + \log\exp(\eta\mathbf{X}_n)\right)\middle|\mathcal{F}_{n-1}\right]$$
$$\leq \mathbb{E}\operatorname{tr}\exp\left(\eta\sum_{i=1}^{n-1}\mathbf{X}_i + \log\mathbb{E}\exp(\eta\mathbf{X}_n)\right).$$

Iterating over n yields:

$$\mathbb{E}\operatorname{tr}\exp\left(\eta\sum_{i=1}^{n}\mathbf{X}_i\right) \leq \operatorname{tr}\exp\left(\sum_{i=1}^{n}\log\mathbb{E}\exp(\eta\mathbf{X}_i)\right),$$

hence the result.

The second inequality is a direct consequence from the fact that for any $\eta < 0$ and any matrix $\mathbf{X}$, $\eta\lambda_{\min}(\mathbf{X}) = \|\eta\mathbf{X}\|$. □

The formulation of Theorem A.3, although more complicated than is the scalar case, is very helpful for matrix concentration analysis. Indeed, since tr exp and log are both order-preserving operators on positive matrices, a bound on $\mathbb{E}\exp(\eta\mathbf{X}_1)$ will now be enough to provide an overall bound of the extreme eigenvalues.

Hoeffding's inequality

The bound we develop here ensures that $\|\mathbf{S}\|$ and $\lambda_{\min}(\mathbf{S})$ do not deviate too much from their counterpart on $\mathbb{E}\mathbf{S}$. We are now ready to state the first result.

Theorem A.4. *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. positive semidefinite random matrices, such that there exists $L > 0$ verifying $\mathbf{0} \preceq \mathbf{X}_1 \preceq L\mathbf{I}$. Let $\mathbf{S}$ be defined as:*

$$\mathbf{S} \triangleq \sum_{i=1}^{n}\mathbf{X}_i.$$

Then, for any $0 < \varepsilon < 1$, one can lowerbound $\lambda_{\min}(\mathbf{S})$ as follows:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{S})) \leq d\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\frac{n\lambda_{\min}(\mathbb{E}\mathbf{X}_1)}{L}}.$$

Similarly, one can upperbound $\|\mathbf{S}\|$ as follows:

$$\mathbb{P}(\|\mathbf{S}\| \geq (1+\varepsilon)\|\mathbb{E}\mathbf{S}\|) \leq d\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\frac{n\|\mathbb{E}\mathbf{X}_1\|}{L}}.$$

The following corollary shows an alternate (but slightly weaker) formmulation which is closer to usual concentration results.

Corollary A.1. *Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ and $\mathbf{S}$ be defined as above. For any $0 < \varepsilon < 1$, one can lower-bound $\lambda_{\min}(\mathbf{S})$ as follows:*

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{S})) \leq d \exp\left(-\frac{\varepsilon^2 \lambda_{\min}(\mathbb{E}\mathbf{X}_1)}{2L}\right).$$

Before proving Theorem A.4, we state a useful lemma for bouding moment generating function of random positive semi-definite matrices.

Lemma A.5. *Let $t \in \mathbb{R}$ and let $\mathbf{X}$ be a random matrix such that $\mathbf{0} \preceq \mathbf{X} \preceq L\mathbf{I}$ almost surely for some $L \geq 0$. Then, one has:*

$$\mathbb{E}\exp(t\mathbf{X}) \preceq \mathbf{I} + \frac{e^{tL}-1}{L}\mathbb{E}\mathbf{X} \preceq \exp\left(\frac{e^{tL}-1}{L}\mathbb{E}\mathbf{X}\right).$$

Proof. Both inequalities are derived from the convexity of the exponential. We will write scalar inequalities based on convexity and then extend them to matrices using the transfer rule in Lemma A.2. Let $t \in \mathbb{R}$, for any $0 \leq x \leq L$, the following holds:

$$e^{tx} \leq e^0 + \frac{x}{L}(e^{tL} - e^0) = 1 + \frac{e^{tL}-1}{L}x.$$

Since $\mathbf{0} \preceq \mathbf{X} \preceq L\mathbf{I}$ almost surely, this can be extented to the matrix exponential using the transfer rule in Lemma A.2:

$$\exp(t\mathbf{X}) \preceq \mathbf{I} + \frac{e^{tL}-1}{L}\mathbf{X}.$$

Taking the expectation yields the result:

$$\mathbb{E}\exp(t\mathbf{X}) \preceq \mathbf{I} + \frac{e^{tL}-1}{L}\mathbb{E}\mathbf{X}.$$

The second inequality is also an application of Lemma A.2 using the inequality $1 + x \leq e^x$ for any $x \in \mathbb{R}$. □

Proof of Theorem A.4. Let $t > 0$. Combining Lemma A.5 and Theorem A.3 yields:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}) \leq t) \leq \inf_{\eta<0} e^{-\eta t} \operatorname{tr}\exp\left(\frac{e^{\eta L}-1}{L}\mathbb{E}\mathbf{S}\right).$$

Reintegrating $\|\cdot\|$ into the RHS yields:

$$\inf_{\eta<0} e^{-t\eta} \operatorname{tr}\exp\left(\frac{e^{\eta L}-1}{L}\mathbb{E}\mathbf{S}\right) \leq \inf_{\eta<0} d e^{-t\eta} \left\|\exp\left(\frac{e^{\eta L}-1}{L}\mathbb{E}\mathbf{S}\right)\right\|$$

$$= \inf_{\eta<0} d e^{-t\eta + \frac{e^{\eta L}-1}{L}\lambda_{\min}(\mathbb{E}\mathbf{S})}.$$

Since the inequality holds for any $\eta < 0$, we can optimize over η. The optimal (lowest) value is reached for $\eta = (L)^{-1}\log(t/\lambda_{\min}(\mathbb{E}\mathbf{S}))$, which is negative if and only if $t < \lambda_{\min}(\mathbb{E}\mathbf{S})$. Let us make the change of variable $t = (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{S})$ for $0 < \varepsilon < 1$, so the condition holds. Substituting the value of η into **(??)** yields:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{S})) \leq d e^{\left((\varepsilon-1)\log(1-\varepsilon)-\varepsilon\right)\frac{n\lambda_{\min}(\mathbb{E}\mathbf{X}_1)}{L}},$$

and the result holds. □

Remark A.1. *Without additional characterization of the problem, the bound $\mathbb{E}\|\mathbf{X}\| \leq \operatorname{tr}\mathbb{E}\mathbf{X} \leq d\|\mathbb{E}\mathbf{X}\|$ is tight: consider a diagonal random matrix $\mathbf{X}$ such that for any $1 \leq i \leq d$, $\mathbb{P}(\mathbf{X} = \mathbf{E}_{ii}) = 1/d$, where $(\mathbf{E}_{ii})_{1\leq i\leq d}$ are the diagonal elements from the canonical base. Then, $\|\mathbf{X}\| = 1$ and $\|\mathbb{E}\mathbf{X}\| = 1/d$, so $\|\mathbf{X}\| = d\|\mathbb{E}\mathbf{X}\|$. If we consider the best arm identification application, this case essentially boils down to the MAB setting and would make the whole linear modeling irrelevant: maybe there is a more subtle way of characterizing linear bandits in order to avoid a brutal d factor in the bound.*

Bennett's and Bernstein's inequalities

Using the Chernoff's bound, we were able to prove, with high probability, the following:

$$\mathbf{x}^\top\left(n\sum_{k=1}^{K}[\mu_G^\star]_k\mathbf{x}_k\mathbf{x}_k^\top\right)^{-1}\mathbf{x} \leq \mathbf{x}^\top\left(\sum_{i=1}^{n}\mathbf{X}_i\right)^{-1}\mathbf{x} \leq \frac{\|\mathbf{x}\|^2}{(1-\varepsilon)}\lambda_{\max}\left(n\sum_{k=1}^{K}[\mu_G^\star]_k\mathbf{x}_k\mathbf{x}_k^\top\right)^{-1}.$$

This is not enough to ensure the convergence of the randomized sampling. Considering again the random matrix

$$\mathbf{S}_n = \sum_{i=1}^{n}\mathbf{X}_i,$$

our goal is to bound the following quantity:

$$\mathbf{x}^\top\left(\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1}\right)\mathbf{x}.$$

One way to bound the above quantity is to bound the maximum eigenvalue of $\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1}$. One has:

$$\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1} = \mathbf{S}_n^{-1}\left(\mathbf{I} - \mathbf{S}_n(\mathbb{E}\mathbf{S}_n)^{-1}\right) = \mathbf{S}_n^{-1}(\mathbb{E}\mathbf{S} - \mathbf{S}_n)(\mathbb{E}\mathbf{S}_n)^{-1}.$$

In Section A.6, we used Hoeffding's inequality to upperbound $\left\|\mathbf{S}_n^{-1}\right\|$ based on $\left\|(\mathbb{E}\mathbf{S}_n)^{-1}\right\|$ value. Therefore, we only need to care about the central term. Since the random matrix $\mathbb{E}\mathbf{S}_n - \mathbf{S}_n$ is not necessarily positive semi-definite anymore, we cannot use Hoeffding's inequality. We can use Bernstein's inequality however, as stated in the following theorem.

Theorem A.5. *Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be n i.i.d. random symmetric matrices such that $\mathbb{E}\mathbf{X}_1 = \mathbf{0}$ and there exists $L > 0$ such that $\|\mathbf{X}_1\| \leq L$, almost surely. Let $\mathbf{S}_n \triangleq \sum_{i=1}^n \mathbf{X}_i$. Then, for any $t > 0$, one has:*

$$\mathbb{P}(\|\mathbf{S}_n\| \geq t) \leq d e^{-\frac{t^2}{2Lt/3 + 2n\sigma^2}},$$

where $\sigma^2 \triangleq \left\|\mathbb{E}\left[\mathbf{X}_1^2\right]\right\|$.

As in the scalar case, Bernstein's inequality relies on using the Taylor expansion of the exponential to bound the moment generating function, so we will need the following lemma.

Lemma A.6. *Let $L > 0$ and let $\mathbf{X}$ be a random Hermitian matrix such that $\mathbb{E}\mathbf{X} = \mathbf{0}$ and $\mathbf{X} \preceq L\mathbf{I}$ almost surely. Then, for any $0 < t < 3/L$, one has:*

$$\mathbb{E}\exp(t\mathbf{X}) \preceq \exp\left(\frac{t^2/2}{1 - tL/3}\mathbb{E}\left[\mathbf{X}^2\right]\right).$$

Proof. Similarly to the Hoeffding's case, we will show a result for the exponential of a scalar and extend it to a Hermitian matrix. Let $L > 0, 0 < x < L$ and $0 < t < 3/L$. Let us define $f : [0, L] \to \mathbb{R}$ such that for any $0 < y < L$,

$$f(y) \triangleq \frac{e^{ty} - ty - 1}{y^2}.$$

In particular, one has $e^{tx} = 1 + tx + x^2 f(x)$. Notice that f is increasing, so $e^{tx} \leq 1 + tx + x^2 f(L)$. Now, using the Taylor expansion of the exponential, we can write:

$$f(L) = \frac{e^{tL} - tL - 1}{L^2} = \frac{1}{L^2}\sum_{k\geq 2}\frac{(tL)^k}{k!} \leq \frac{t^2}{2}\sum_{k\geq 2}\frac{(tL)^{k-2}}{3^{k-2}} = \frac{t^2/2}{1 - tL/3},$$

where the inequality comes from the fact that $k! \geq 2 \times 3^{k-2}$, for any $k \geq 2$.

Now, using the fact that $\mathbf{X} \preceq L\mathbf{I}$ almost surely, we can obtain the following bound:

$$\exp(t\mathbf{X}) \preceq \mathbf{I} + t\mathbf{X} + \mathbf{X}(f(L)\mathbf{I})\mathbf{X} = \mathbf{I} + t\mathbf{X} + f(L)\mathbf{X}^2.$$

Finally, taking the expectation and combining this result with a common bound of the exponential, we obtain:

$$\mathbb{E}\exp(t\mathbf{X}) \preceq \mathbf{I} + \frac{t^2/2}{1 - tL/3}\mathbb{E}\left[\mathbf{X}^2\right] \preceq \exp\left(\frac{t^2/2}{1 - tL/3}\mathbb{E}\left[\mathbf{X}^2\right]\right),$$

hence the result. □

We are now ready to prove Bernstein's inequality for matrices.

Proof of Theorem A.5. Let $0 < \eta < 3/L$, using Markov's inequality one has:

$$\mathbb{P}(\|\mathbf{S}_n\| \geq t) = \mathbb{P}\Big(e^{\eta\|\mathbf{S}_n\|} \geq e^{\eta t}\Big) \leq e^{-\eta t}\mathbb{E}e^{\eta\|\mathbf{S}_n\|}.$$

By Lemma A.6 and the subadditivity of the matrix CGF [96, Lemma 3.5.1, Ch. 3], we obtain

$$\mathbb{E}e^{\eta\|\mathbf{S}_n\|} = \mathbb{E}\|\exp(\eta\mathbf{S}_n)\| \leq \operatorname{tr}\mathbb{E}\exp(\eta\mathbf{S}_n) \leq \operatorname{tr}\exp\left(\frac{t^2/2}{1-tL/3}\mathbb{E}\big[\mathbf{S}_n^2\big]\right).$$

Plugin the trace back into the exponential yields:

$$\mathbb{E}e^{\eta\|\mathbf{S}_n\|} \leq \operatorname{tr}\exp\left(\frac{\eta^2/2}{1-\eta L/3}\mathbb{E}\big[\mathbf{S}_n^2\big]\right) \leq de^{\frac{\eta^2/2}{1-\eta L/3}\left\|\mathbb{E}\left[\mathbf{S}_n^2\right]\right\|}.$$

Optimizing on η would lead to a complicated result, so we use instead $\eta = t/(n\sigma^2 + tL/3)$, which verifies the condition $\eta < 3/L$ and yields the final result. □

The relationship between the precision (nt in Theorem A.5) and the confidence level δ_b (the RHS of the concentration inequality) is more complicated in Bernstein's inequality than in Hoeffding's. It requires solving a second order polynomial equation and leads to:

$$t = \frac{L}{3n}\log\frac{2d}{\delta_b} + \sqrt{\left(\frac{L}{3n}\log\frac{2d}{\delta_b}\right)^2 + \frac{2\sigma^2}{n}\log\frac{2d}{\delta_b}}.$$

In our case, we will use the bound provided by Bennett's inequality applied to random Hermitian matrices, as it is simpler to derive the precision associated to a confidence level.

Theorem A.6. *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be n i.i.d. random Hermitian matrices such that $\mathbb{E}\mathbf{X}_1 = \mathbf{0}$ and there exists $\sigma^2 > 0$ such that $\|\mathbb{E}\big[\mathbf{X}_1\big]^2\| \leq \sigma^2$. In addition, let us assume that there exists $c > 0$ such that for any $q \geq 3$:*

$$\left\|\mathbb{E}\big[(\mathbf{X}_1)_+^q\big]\right\| \leq \frac{q!}{2}\sigma^2 c^{q-2},$$

where for any symmetric matrix $\mathbf{X}$,$(\mathbf{X})_+$ is the orthogonal projection of $\mathbf{X}$ onto the semidefinite positive cone. Then, for any $t > 0$, one has:

$$\mathbb{P}\left(\left\|\sum_{i=1}^n \mathbf{X}_i\right\| \geq \sqrt{2n\sigma^2 t} + ct\right) \leq de^{-t}.$$

The proof is very similar to Bernstein's: we need an intermediary result on the moment generating function, as stated in the following lemma.

Lemma A.7. *Let $\sigma^2, c > 0$ and let $\mathbf{X}$ be a random Hermitian matrix such that $\mathbb{E}\mathbf{X} = \mathbf{0}$ and $\|\mathbb{E}\big[\mathbf{X}^2\big]\| \leq \sigma^2$. In addition, we assume that for any $q \geq 3$,*

$$\left\|\mathbb{E}\big[(\mathbf{X})_+^q\big]\right\| \leq \frac{q!}{2}\sigma^2 c^{q-2}.$$

Then, for any $0 < t < 1/c$*, one has:*

$$\mathbb{E}\exp(t\mathbf{X}) \preceq \exp\left(\frac{t^2/2}{1-ct}\mathbb{E}\left[\mathbf{X}^2\right]\right).$$

The proof of this Lemma is ommitted as it is very similar to Lemma A.6.

Proof of Theorem A.6. This proof can be directly adapted from Bernstein's using standard results on concentration (see *e.g.*, [19] for details). □

Proof of Theorem A.1. As mentionned in the beginning of this section, our goal is to bound $\|\mathbf{S}_n^{-1}(\mathbf{S}_n - \mathbb{E}\mathbf{S}_n)(\mathbb{E}\mathbf{S}_n)^{-1}\|$. Let us assume that the batch size n satisfies:

$$n > \frac{2L\log d}{\lambda_{\min}(\mathbb{E}\mathbf{X}_1)}.$$

Let $de^{-\frac{n\lambda_{\min}(\mathbb{E}\mathbf{X}_1)}{2L}} < \delta_h < 1$. Using the Chernoff's bound, we know that with probability at least $1-\delta_h$, the following holds true:

$$\|\mathbf{S}_n^{-1}\| \leq \frac{\|(\mathbb{E}\mathbf{S}_n)^{-1}\|}{1-\sqrt{\frac{2L}{n}\|(\mathbb{E}\mathbf{X}_1)^{-1}\|\log(d/\delta_h)}}.$$

Similarly, let $0 < \delta_b < 1$; using Bennett's inequality, with probability at least $1-\delta_b$, we have:

$$\|\mathbf{S}_n - \mathbb{E}\mathbf{S}_n\| \leq \frac{L}{3}\log\frac{d}{\delta_b} + \sqrt{2n\sigma^2\log\frac{d}{\delta_b}}.$$

Combining these two results with a union bound leads to the following bound, with probability $1-(\delta_b+\delta_h)$:

$$\begin{aligned}\|\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1}\| &\leq \|(\mathbb{E}\mathbf{S}_n)^{-1}\|^2\frac{(L/3)\log(d/\delta_b) + \sqrt{2n\sigma^2\log(d/\delta_b)}}{1-\sqrt{(2L/n)\|(\mathbb{E}\mathbf{X}_1)^{-1}\|\log(d/\delta_h)}} \\ &\leq \frac{1}{n^2}\|(\mathbb{E}\mathbf{X}_1)^{-1}\|^2\frac{(L/3)\log(d/\delta_b) + \sqrt{2n\sigma^2\log(d/\delta_b)}}{1-\sqrt{(2L/n)\|(\mathbb{E}\mathbf{X}_1)^{-1}\|\log(d/\delta_h)}} \qquad \text{(A.1)}\end{aligned}$$

In order to obtain a unified bound depending on one confidence parameter $1-\delta$, one could optimize over δ_b and δ_h, subject to $\delta_b + \delta_h = \delta$. This leads to a messy result and a negligible improvement. One can use simple values $\delta_b = \delta_h = \delta/2$, so the overall bound becomes, with probability $1-\delta$, $\|\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1}\|$ is upper bounded by

$$\frac{1}{n}\|(\mathbb{E}\mathbf{X}_1)^{-1}\|^2\sqrt{\frac{2\sigma^2}{n}\log\left(\frac{2d}{\delta}\right)}\left(\frac{1+\sqrt{(L^2/18\sigma^2 n)\log(2d/\delta)}}{1-\sqrt{(2L/n)\|(\mathbb{E}\mathbf{X}_1)^{-1}\|\log(2d/\delta)}}\right).$$

This can finally be formulated as follows:

$$\|\mathbf{S}_n^{-1} - (\mathbb{E}\mathbf{S}_n)^{-1}\| \leq \frac{1}{n}\|(\mathbb{E}\mathbf{X}_1)^{-1}\|^2\sqrt{\frac{2\sigma^2}{n}\log\left(\frac{2d}{\delta}\right)} + o\left(\frac{1}{n\sqrt{n}}\right).$$

The final result yields using the fact that $\max_{\mathbf{x}\in\mathcal{X}}\|\mathbf{x}\|^2 = L$ and $f^\star_{G,n} = f^\star_{D,n} = \frac{d}{n}$.

The bound on $f_E(\mathbf{S}_n^{-1})$ is obtained similarly, only using the Hoeffding's result on minimum eigenvalue. □

A refined approach of the dimension

Intrinsic dimension

Definition A.3 (Intrinsic dimension)**.** *Let $d > 0$ and $\mathbf{S}_n \in \mathbb{R}^{d\times d}$ be a positive semi-definite matrix. The intrisic dimension of $\mathbf{S}_n$, denoted $\mathrm{intdim}(\mathbf{S}_n)$, is defined as follows:*

$$\mathrm{intdim}(\mathbf{S}_n) \triangleq \frac{\mathrm{tr}(\mathbf{S}_n)}{\|\mathbf{S}_n\|}.$$

One always has $1 \leq \mathrm{intdim}(\mathbf{S}_n) \leq d$.

As for the regular concentration proofs, we will need two useful lemmas: one for deriving a nicer upperbound from Markov's inequality and the other for forcing the intrinsic dimension into the bound.

Lemma A.8. *Let $\mathbf{Z} \in \mathbb{R}^d$ be a random Hermitian matrix and let $\psi : \mathbb{R} \to \mathbb{R}_+$ be non-decreasing and non-negative. Then, for any $t \in \mathbb{R}$ such that $\psi(t) > 0$, one has:*

$$\mathbb{P}(\|\mathbf{Z}\| \geq t) \leq \frac{1}{\psi(t)}\mathbb{E}\,\mathrm{tr}(\psi(\mathbf{Z})).$$

Proof. Let $t \in \mathbb{R}$. Since ψ is non-decreasing, the event $\{\|\mathbf{Z}\| \geq t\}$ contains $\{\psi(\|\mathbf{Z}\|) \geq \psi(t)\}$. In addition, using the definition of $\psi(\mathbf{Z})$, one can easily notice that $\psi(\mathbf{Z}) \succeq \mathbf{0}$ and $\|\psi(\mathbf{Z})\| \geq \psi(\|\mathbf{Z}\|)$. Therefore, one can write:

$$\mathbb{P}(\|\mathbf{Z}\| \geq t) \leq \mathbb{P}(\|\psi(\mathbf{Z})\| \geq \psi(t)) \leq \mathbb{P}(\mathrm{tr}(\psi(\mathbf{Z})) \geq \psi(t)),$$

where we used the fact that $\psi(\mathbf{Z}) \succeq \mathbf{0}$ in the rightmost inequality. Finally, one can conclude using Markov's inequality. □

Lemma A.9. *Let $\varphi : \mathbb{R} \mapsto \mathbb{R}$ be a convex function and let $\mathbf{Z}$ be a positive semi-definite matrix. Then, one has:*

$$\mathrm{tr}\left(\varphi(\mathbf{Z})\right) \leq \mathrm{intdim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big) + (d - \mathrm{intdim}(\mathbf{Z}))\varphi(0).$$

In particular, if $\varphi(0) = 0$*, one has:*

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq \operatorname{intdim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big).$$

Proof. Let $0 \leq x \leq \|\mathbf{Z}\|$. By convexity of φ, we can write:

$$\varphi(x) \leq \varphi(0) + \big(\varphi(\|\mathbf{Z}\|) - \varphi(0)\big)\frac{x}{\|\mathbf{Z}\|}.$$

Using Lemma A.3, we can extend the above inequality to $\mathbf{Z}$:

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq \operatorname{tr}\big(\varphi(0)\mathbf{I}\big) + \frac{\varphi(\|\mathbf{Z}\|) - \varphi(0)}{\|\mathbf{Z}\|}\operatorname{tr}\big(\mathbf{Z}\big),$$

which can be rearranged as follows:

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq \operatorname{intdim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big) + (d - \operatorname{intdim}(\mathbf{Z}))\varphi(0),$$

and the result holds. □

Using the two previous lemmas, we can adapt the proof in Hoeffding in order to obtain an improved bound with the intrinsic dimension.

Theorem A.7. *Let* $L > 0$ *and let* $\mathbf{X}_1, \ldots, \mathbf{X}_n$ *be i.i.d. random matrices such that* $\mathbf{0} \preceq \mathbf{X}_1 \preceq L\mathbf{I}$*. Let* $\mathbf{S}_n = \sum_{i=1}^n \mathbf{X}_i$*. For any* $0 < \varepsilon < 1$*, one can upperbound* $\|\mathbf{S}_n\|$ *as follows:*

$$\mathbb{P}(\|\mathbf{S}_n\| \geq (1+\varepsilon)\|\mathbb{E}\mathbf{S}_n\|) \leq 2 \times \operatorname{intdim}(\mathbf{Z})\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\frac{\|\mathbb{E}\mathbf{S}_n\|}{L}}.$$

Proof. Let $t, \eta > 0$. Using Lemma A.8 with $\psi : x \in \mathbb{R} \mapsto (e^{\eta x} - 1)_+$ yields:

$$\mathbb{P}(\|\mathbf{S}_n\| \geq t) \leq \frac{1}{e^{\eta t} - 1}\mathbb{E}\operatorname{tr}\big((\exp(\eta\mathbf{S}_n) - \mathbf{I})_+\big) = \frac{1}{e^{\eta t} - 1}\operatorname{tr}(\exp(\eta\mathbf{S}_n) - \mathbf{I}), \quad \text{(A.2)}$$

where we used the fact that $\mathbf{S}_n \succeq \mathbf{0}$ implies $\exp(\eta\mathbf{S}_n) \succeq \mathbf{I}$. Let $0 \leq x \leq L$, by convexity of the exponential, one has:

$$e^{\eta x} - 1 \leq \big(e^{\eta L} - 1\big)\frac{x}{L}.$$

Once again, we can extend this result to $\mathbf{S}_n$ and obtain:

$$\operatorname{tr}\big(\exp(\eta\mathbf{S}_n) - \mathbf{I}\big) \leq \operatorname{tr}\left(\frac{e^{\eta L} - 1}{L}\mathbf{S}_n\right).$$

Taking the expectation and using the inequality $x \leq e^x - 1$ yields:

$$\mathbb{E}\,\mathrm{tr}\left(\exp(\eta \mathbf{S}_n) - \mathbf{I}\right) \leq \mathrm{tr}\left(\frac{e^{\eta nL} - 1}{nL}\mathbb{E}\mathbf{S}_n\right) \leq \mathrm{tr}\left(\exp\left(\frac{e^{\eta nL} - 1}{nL}\mathbb{E}\mathbf{S}_n\right) - \mathbf{I}\right)$$

We can now use Lemma A.9 with $\varphi : x \in \mathbb{R} \mapsto e^x - 1$ to obtain a bound depending on $\|\mathbb{E}\mathbf{S}_n\|$:

$$\begin{aligned}\mathbb{E}\,\mathrm{tr}\left(\exp(\eta \mathbf{S}_n) - \mathbf{I}\right) &\leq \mathrm{tr}\left(\exp\left(\frac{e^{\eta nL} - 1}{nL}\mathbb{E}\mathbf{S}_n\right) - \mathbf{I}\right) \\ &\leq \mathrm{intdim}(\mathbb{E}\mathbf{S}_n)\left(e^{\frac{e^{\eta nL}-1}{nL}\|\mathbb{E}\mathbf{S}_n\|} - 1\right).\end{aligned}$$

Combining the previous inequality with (A.2) yields:

$$\begin{aligned}\mathbb{P}(\|\mathbf{S}_n\| \geq t) &\leq \mathrm{intdim}(\mathbb{E}\mathbf{S}_n) \times \frac{e^{\frac{e^{\eta nL}-1}{nL}\|\mathbb{E}\mathbf{S}_n\|} - 1}{e^{\eta t} - 1} \\ &\leq \mathrm{intdim}(\mathbb{E}\mathbf{S}_n) \times \frac{e^{\eta t}}{e^{\eta t} - 1} \cdot e^{-\eta t + \frac{e^{\eta nL}-1}{nL}\|\mathbb{E}\mathbf{S}_n\|}.\end{aligned}$$

The remainder of the proof consists in bounding $e^{\eta t}/(e^{\eta t} - 1)$ by 2 and the rightmost term as in the regular Hoeffding's proof (see [96] for additional details). □

There are two main differences between this version of the Hoeffding's bound and the regular one. First, there is a factor 2 with the intrinsic dimension. This a not necessarily a big deal as we can win on other aspects. Then, we have obtained a bound on the highest eigenvalue, but not on the lowest. This is due to the current definition of the intrinsic dimension: we use this limitation as a motivation for the refinement we propose in the next section.

A refined approach of the intrinsic dimension

Definition A.4 (Upper and lower intrinsic dimension)**.** *Let $d > 0$ and $\mathbf{S}_n \in \mathbb{R}^{d\times d}$ be a positive semi-definite matrix. The upper and lower intrisic dimensions of $\mathbf{S}_n$, denoted $\mathrm{updim}(\mathbf{S}_n)$ and $\mathrm{lowdim}(\mathbf{S}_n)$ respectively, are defined as follows:*

$$\begin{cases} \mathrm{updim}(\mathbf{S}_n) & \triangleq & \dfrac{\mathrm{tr}\left(\mathbf{S}_n - \lambda_{\min}(\mathbf{S}_n)\mathbf{I}\right)}{\|\mathbf{S}_n\| - \lambda_{\min}(\mathbf{S}_n)} \\ \mathrm{lowdim}(\mathbf{S}_n) & \triangleq & \dfrac{\mathrm{tr}\left(\|\mathbf{S}_n\|\mathbf{I} - \mathbf{S}_n\right)}{\|\mathbf{S}_n\| - \lambda_{\min}(\mathbf{S}_n)} = d - \mathrm{updim}(\mathbf{S}_n). \end{cases}$$

One always has $1 \leq \mathrm{updim}(\mathbf{S}_n), \mathrm{lowdim}(\mathbf{S}_n) \leq d - 1$.

This definition brings a different information about the matrix at stake: instead of renormalizing the trace using the spectral norm, we also shift it using the lowest eigenvalue. With these new quantities, we are able to formulate a refined version of Lemma A.9.

Lemma A.10. *Let $\varphi : \mathbb{R} \mapsto \mathbb{R}$ be a convex function and let $\mathbf{Z}$ be a positive semi-definite matrix. Then, one has:*

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq \operatorname{updim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big) + \operatorname{lowdim}(\mathbf{Z})\varphi(\lambda_{\min}(\mathbf{Z})).$$

This bound is always tighter than the one with the intrinsic dimension, that is:

$$\operatorname{updim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big)+\operatorname{lowdim}(\mathbf{Z})\varphi(\lambda_{\min}(\mathbf{Z})) \leq \operatorname{intdim}(\mathbf{Z})\varphi\big(\|\mathbf{Z}\|\big)+(d-\operatorname{intdim}(\mathbf{Z}))\varphi(0).$$

Proof. To prove both assertions, we will show a more general bound, of the form:

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq f(l),$$

for $0 \leq l \leq \lambda_{\min}(\mathbf{Z})$ and then we will show that f is non-increasing. Let $0 \leq l \leq \lambda_{\min}(\mathbf{Z})$ and let $l \leq x \leq \|\mathbf{Z}\|$. Using the convexity of φ, one can write:

$$\varphi(x) \leq \varphi(l) + (\varphi(\|\mathbf{Z}\|) - \varphi(l))\frac{x-l}{\|\mathbf{Z}\|-l} = \frac{x-l}{\|\mathbf{Z}\|-l}\varphi(\|\mathbf{Z}\|) + \frac{\|\mathbf{Z}\|-x}{\|\mathbf{Z}\|-l}\varphi(l).$$

Using Lemma A.3 again, we can extend the above inequality to $\mathbf{Z}$:

$$\operatorname{tr}\big(\varphi(\mathbf{Z})\big) \leq \frac{\operatorname{tr}(\mathbf{Z}-l\mathbf{I})}{\|\mathbf{Z}\|-l}\varphi(\|\mathbf{Z}\|) + \frac{\operatorname{tr}(\|\mathbf{Z}\|\mathbf{I}-\mathbf{Z})}{\|\mathbf{Z}\|-l}\varphi(l).$$

It is immediate to see that taking $l = 0$ leads to Lemma A.9 and $l = \lambda_{\min}(\mathbf{Z})$ shows the first assertion of this lemma. The last assertion of the theorem just comes from the convexity of φ: when applying the convexity bound on two segments $\mathcal{I} \subseteq \mathcal{J}$, the bound on $\mathcal{I}$ is necessarily tighter than the bound on $\mathcal{J}$. □

Theorem A.8. *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. positive semidefinite random matrices, such that there exists $L > 0$ verifying $\mathbf{0} \preceq \mathbf{X}_1 \preceq L\mathbf{I}$. Let $\mathbf{S}_n$ be defined as:*

$$\mathbf{S}_n \triangleq \sum_{i=1}^{n} \mathbf{X}_i.$$

In addition, let κ be the condition number of $\mathbb{E}\mathbf{S}_n$, that is

$$\kappa \triangleq \frac{\|\mathbb{E}\mathbf{S}_n\|}{\lambda_{\min}(\mathbb{E}\mathbf{S}_n)}.$$

Then, for any $0 < \varepsilon < 1$, one can lowerbound $\lambda_{\min}(\mathbf{S}_n)$ as follows:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}_n) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}\mathbf{S}_n)) \leq \tilde{d}_{\min}\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\frac{n\lambda_{\min}(\mathbb{E}\mathbf{X}_1)}{L}},$$

where

$$\tilde{d}_{\min} = \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) + \operatorname{updim}(\mathbb{E}\mathbf{S}_n) e^{-n\varepsilon\lambda_{\min}(\mathbb{E}\mathbf{X}_1)(\kappa-1)/L}.$$

Similarly, one can upperbound $\|\mathbf{S}_n\|$ *as follows:*

$$\mathbb{P}(\|\mathbf{S}_n\| \geq (1+\varepsilon)\|\mathbb{E}\mathbf{S}_n\|) \leq \tilde{d}_{\max}\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\frac{n\|\mathbb{E}\mathbf{X}_1\|}{L}},$$

where

$$\tilde{d}_{\max} = \operatorname{updim}(\mathbb{E}\mathbf{S}_n) + \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) e^{-n\varepsilon\|\mathbb{E}\mathbf{X}_1\|(1-\kappa^{-1})/L}.$$

Proof. The beggining of the proof is similar to the regular Hoeffding's proof so we can directly write, for $t, \eta > 0$:

$$\mathbb{P}(\|\mathbf{S}_n\| \geq t) \leq e^{-\eta t} \operatorname{tr}\exp\left(\frac{e^{\eta L}-1}{L}\mathbb{E}\mathbf{S}_n\right)$$

Using Lemma A.10 with $\varphi : x \in \mathbb{R} \mapsto e^{g(\eta)x}$ and $g : \eta \mapsto L^{-1}(e^{\eta L}-1)$ yields:

$$\begin{aligned}\operatorname{tr}\exp(g(\eta)\mathbb{E}\mathbf{S}_n) &\leq \operatorname{updim}(\mathbb{E}\mathbf{S}_n) e^{g(\eta)\|\mathbb{E}\mathbf{S}_n\|} + \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) e^{g(\eta)\lambda_{\min}(\mathbb{E}\mathbf{S}_n)} \\ &= \left(\operatorname{updim}(\mathbb{E}\mathbf{S}_n) + \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) e^{g(\eta)(\lambda_{\min}(\mathbb{E}\mathbf{S}_n)-\|\mathbb{E}\mathbf{S}_n\|)}\right) e^{g(\eta)\|\mathbb{E}\mathbf{S}_n\|}.\end{aligned}$$

Using the same value for η than in regular Hoeffding's proof thus yields:

$$\mathbb{P}(\|\mathbf{S}_n\| \geq (1+\varepsilon)\|\mathbb{E}\mathbf{S}_n\|) \leq \tilde{d}\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\|\mathbb{E}\mathbf{S}_n\|/L}$$

where

$$\tilde{d} = \operatorname{updim}(\mathbb{E}\mathbf{S}_n) + \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) e^{g(\eta)(\lambda_{\min}(\mathbb{E}\mathbf{S}_n)-\|\mathbb{E}\mathbf{S}_n\|)}.$$

Pluging the value $\eta = L^{-1}\log(1+\varepsilon)$ into $\tilde{d}$'s expression yields the result. The result on $\lambda_{\min}(\mathbf{S}_n)$ is very similar:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}_n) \leq t) = \mathbb{P}(\|-\mathbf{S}_n)\| \geq -t) \leq e^{\eta t} \operatorname{tr}\exp\left(\frac{e^{-\eta L}-1}{L}\mathbb{E}\mathbf{S}_n\right).$$

Using the same reasoning, one obtains:

$$\mathbb{P}(\lambda_{\min}(\mathbf{S}_n) \leq (1-\varepsilon)\|\mathbb{E}\mathbf{S}_n\|) \leq \tilde{d}\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\lambda_{\min}(\mathbb{E}\mathbf{S}_n)/L}$$

where

$$\tilde{d} = \operatorname{lowdim}(\mathbb{E}\mathbf{S}_n) + \operatorname{updim}(\mathbb{E}\mathbf{S}_n) e^{g(-\eta)(\|\mathbb{E}\mathbf{S}_n\|-\lambda_{\min}(\mathbb{E}\mathbf{S}_n))},$$

which proves the result since $\eta = -L^{-1}\log(1-\varepsilon)$. □

Theorem A.9. *Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be n i.i.d. random symmetric matrices such that $\mathbb{E}\mathbf{X}_1 = \mathbf{0}$ and there exists $L > 0$ such that $\|\mathbf{X}_1\| \leq L$, almost surely. Let $\mathbf{S}_n \triangleq \sum_{i=1}^n \mathbf{X}_i$. Let $\mathbf{V}$ be the covariance matrix of $\mathbf{X}_1$, that is $\mathbf{V} \triangleq \mathbb{E}\big[\mathbf{X}_1^2\big] - \mathbf{M}(\mu_G^\star)^2$ and let κ be its condition number. Let t verifying*

$$\frac{3n\|\mathbf{V}\|^2}{L} > t > \sqrt{n}\|\mathbf{V}\|^2 + \frac{L}{3\sqrt{n}}.$$

Then, one has:

$$\mathbb{P}\Big(\|\mathbf{S}_n\| \geq \sqrt{t}\Big) \leq \tilde{d} e^{-\dfrac{t^2}{4\|\mathbf{V}\|^2}},$$

where

$$\tilde{d} = \mathrm{updim}(\mathbf{V}) + \mathrm{lowdim}(\mathbf{V}) e^{-\frac{n}{16}(1-\kappa^{-1})}.$$

Proof. One can combine reasonings of Bernstein's regular proof with the proof of Hoeffding's with updim and lowdim to obtain:

$$\mathbb{P}(\|\mathbf{S}_n\| \geq \sqrt{n}t) \leq \mathrm{updim}(\mathbf{V}) e^{-\frac{t^2/2}{\sigma^2 + Lt/3n}} + \mathrm{lowdim}(\mathbf{V}) e^{-\frac{t^2/2}{\sigma^2+Lt/3n}(2-\kappa^{-1})}.$$

Let us assume that

$$\frac{3n\sigma^2}{L} > t > \sqrt{n}\sigma^2 + \frac{L}{3\sqrt{n}}.$$

Then, the previous result can be bounded as follows:

$$\begin{aligned}\mathbb{P}(\|\mathbf{S}_n\| \geq \sqrt{n}t) &\leq \left(\mathrm{updim}(\mathbf{V}) + \mathrm{lowdim}(\mathbf{V}) e^{-\frac{t^2}{4\sigma^2}(1-\kappa^{-1})}\right) e^{-\frac{t^2}{4\sigma^2}} \\ &\leq \left(\mathrm{updim}(\mathbf{V}) + \mathrm{lowdim}(\mathbf{V}) e^{-\frac{n}{16}(1-\kappa^{-1})}\right) e^{-\frac{t^2}{4\sigma^2}},\end{aligned}$$

and the result holds. □

Link with the best arm identification in linear bandits

Let $d > 0$ and $\mathcal{X} \subseteq \mathbb{R}^d$ a subset of $\mathbb{R}^d$, corresponding to the bandit arms. The linear bandit setting assumes that the conditional distribution of the rewards given the arm follows a linear model: there exists an unknown parameter $\theta_\star \in \mathbb{R}^d$ such that the reward $r(\mathbf{x})$ associated to any action $\mathbf{x} \in \mathcal{X}$ is of the form

$$r(\mathbf{x}) = \theta_\star^\top \mathbf{x} + \epsilon,$$

where ϵ is a R-subgaussian noise independent from $\mathbf{x}$. This linear structure implies that some information is shared between arms through the parameter $\theta_\star$: an action-reward pair $(\mathbf{x}, r(\mathbf{x}))$ gives information about $\theta_\star$ and thus about the reward distributions of the other actions. This makes this setting very different to the classical multi-armed bandit setting where the reward distributions of each action are assumed to be independent. Whereas multi-armed bandit algorithms

mainly focus on the estimation of the mean reward of each action, linear bandit algorithms are mainly interested in the estimation of the parameter $\theta_\star$.

Depending on the context, the goal of a bandit algorithm can either be to maximize the cumulated reward (the sum of the rewards collected over several iterations) or to find the arm maximizing the reward, referred to as *best arm identification* or *pure exploration*.

We here focus on best arm identification in linear bandits whose objective is to find the arm $\mathbf{x}_\star$ maximizing the average reward:

$$\mathbf{x}_\star = \arg\max_{\mathbf{x}\in\mathcal{X}} \theta_\star^\top \mathbf{x}\,.$$

As the parameter $\theta_\star$ is unknown the aim is to design a strategy that will sequentially choose t actions $\mathbf{x}_1, \dots, \mathbf{x}_t \in \mathcal{X}$ and collect their associated rewards $r_i = \theta_\star^\top \mathbf{x}_i + \epsilon_i$, $1 \leq i \leq t$, where $\epsilon_1, \dots, \epsilon_t$ are independent realizations of ϵ, to obtain an estimate $\hat{\theta}_t$ of $\theta_\star$. To find the best arm, the estimated prediction $\hat{\theta}_t^\top \mathbf{x}$ should be close to the real prediction $\theta_\star^\top \mathbf{x}$ for all $\mathbf{x} \in \mathcal{X}$. More precisely, rather than the reward prediction of an action itself, we are interested in comparing the predictions of each pair of arms. We thus want $|(\hat{\theta}_t - \theta_\star)^\top(\mathbf{x} - \mathbf{x}')|$ to be small.

Remark A.2. *Note that compared to the multi-armed bandit case where known suboptimal arms are no longer played, the situation is different for the best arm identification case as playing suboptimal arms might give information about the parameter $\theta_\star$ and improve the discrimination of the unknown $\mathbf{x}_\star$ with other arms.*

Most of the designed strategies for best arm identification in linear bandits [90, 92, 106] have relied on two concentration inequalities giving high probability bounds on the prediction error $|(\hat{\theta}_t - \theta_\star)^\top \mathbf{x}|$ of the regression estimator $\hat{\theta}_t$ obtained from a sequence of action-reward pairs. The first concentration inequality is only valid when the sequence of actions is fixed and hence cannot depend on the observed random rewards. The authors of [2] derived a concentration inequality which holds when the sequence of actions is adaptive to the observed random rewards. However this concentration inequality offers a looser bound than the one given for fixed sequences. In [90] strategies relying on the fixed sequence bound are developed whereas [106] designed a fully adaptive algorithm based on the adaptive bound. These two concentration inequalities are detailed below.

Let $\hat{\theta}_t(\lambda)$ denote the ridge estimate of $\theta_\star$ with a ℓ_2-penalty λ:

$$\hat{\theta}_t(\lambda) \triangleq \arg\min_{\theta\in\mathbb{R}^d} \sum_{s=1}^{t} (\theta^\top \mathbf{x}_s - r_s)^2 + \frac{\lambda}{2}\|\theta\|^2.$$

The ridge estimate $\hat{\theta}_t(\lambda)$ can be expressed in closed form:

$$\hat{\theta}_t(\lambda) = \hat{\mathbf{A}}_t(\lambda)^{-1} \mathbf{X}_t^\top \mathbf{r}_t,$$

where $\mathbf{X}_t^\top \triangleq (\mathbf{x}_1, \dots, \mathbf{x}_t)$, $\mathbf{r}_t^\top \triangleq (r_1, \dots, r_t)$ and $\hat{\mathbf{A}}_t(\lambda) \triangleq \mathbf{X}_t^\top \mathbf{X}_t + \lambda \mathbf{I}_d$.

Fixed design concentration inequality. We assume that there is a finite number of arms $|\mathcal{X}| = K$. If $\lambda = 0$ and $(\mathbf{x}_i)_{1\leq i\leq\infty}$ is a fixed sequence of actions (independent of the random rewards $(\mathbf{r}_i)_{1\leq i\leq\infty}$) we have the following concentration inequality [90]: for all $\delta \in (0,1)$,

$$\mathbb{P}\left(\forall t \in \mathbb{N}, \forall \mathbf{x} \in \mathcal{X}, |\theta_\star^\top \mathbf{x} - \hat{\theta}_t^\top \mathbf{x}| \leq 2R\|\mathbf{x}\|_{\hat{\mathbf{A}}_t^{-1}} \sqrt{2\log\left(\frac{6t^2K}{\pi^2\delta}\right)}\right) \geq 1-\delta\,. \tag{A.3}$$

One may notice that this result holds over these directions by replacing K by K^2 in the logarithmic term, as there are of the order of K^2 such directions.[3]

Adaptive design concentration inequality. When the sequence of actions is chosen adaptively of the history, *i.e.*, for all $i \in \mathbb{N}$, $\mathbf{x}_i$ is allowed to depend on $(\mathbf{x}_1, r_1, \ldots, \mathbf{x}_{t-1}, r_{t-1})$, we need to rely on a result established by [2]: if $\lambda > 0$ and $\|\mathbf{x}_i\| \leq L$ for all i then for all $\delta \in (0,1)$ and all $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbb{P}\left(|\theta_\star^\top \mathbf{x} - \hat{\theta}_t(\lambda)^\top \mathbf{x}| \leq \|\mathbf{x}\|_{\hat{\mathbf{A}}_t(\lambda)^{-1}} \left(R\sqrt{d\log\left(\frac{1+tL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}\|\theta_\star\|\right)\right) \geq 1-\delta\,. \tag{A.4}$$

The reader can refer to [2, Appendix B] for the proof of this result. The main difference with (A.3) is the presence of an extra $\sqrt{d}$ factor which cannot be removed and which makes adaptive algorithms suffer more from the dimension than fixed design strategies (see [59, Chapter 20] for a more complete discussion on this aspect). We now omit the dependence of $\hat{\mathbf{A}}_t(\lambda)^{-1}$ in λ when it is not relevant for the purpose of the discussion.

Whichever concentration inequality is used, the bound on the prediction error in a direction $\mathbf{y} = \mathbf{x} - \mathbf{x}'$, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ depends on the matrix norm $\|\mathbf{y}\|_{\hat{\mathbf{A}}_t^{-1}}$. The goal of a strategy for the problem of best arm identification in linear bandits as formulated in [90] is to choose a sequence of actions that reduces this matrix norm as fast as possible for all directions $\mathbf{y}$ so as to reduce the prediction error and be able to identify the best arm. This approach thus leads to the following optimization problem:

$$(\mathbf{x}_1, \ldots, \mathbf{x}_B) \in \operatorname*{arg\,min}_{\mathbf{x}_1,\ldots,\mathbf{x}_B \in \mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \mathbf{y}^\top \left(\sum_{i=1}^t \mathbf{x}_i\mathbf{x}_i^\top\right)^{-1} \mathbf{y}. \tag{A.5}$$

If one upper bounds $\|\mathbf{y}\|_{\hat{\mathbf{A}}_t^{-1}}$ by $2\|\mathbf{x}\|_{\hat{\mathbf{A}}_t^{-1}}$ we finally obtain the G-optimal design.

Details on experiment setting and comments

For the randomized strategies we use the *cvxopt* python package [8] to compute the solution of the semi-definite program associated to the E-optimal design and compute the solution of the convex relaxation of the D-optimal design problem. We recall that as the relaxed G-optimal design problem is equivalent to the relaxed D-optimal design problem we can use the solution of the latter

[3] It suffices to consider exactly $K(K-1)/2$ directions as the result is the same for $\mathbf{x} - \mathbf{x}'$ and $\mathbf{x}' - \mathbf{x}$.

for the former. Finally, for the greedy implementations of E and G-optimal design, when there are ties between several samples at a given iteration we uniformly select one at random.

Randomized strategy versus greedy strategy for E-optimal design

We recall here that the goal of E-optimal design is to choose experiments maximizing $\lambda_{\min}(\sum_{k=1}^{K} n_k \mathbf{x}_k \mathbf{x}_k^\top)$. We generate a pool of experiments in $\mathbb{R}^d$ made of K independent and identically distributed realizations of a standard Gaussian random variable. Figure A.1a shows the performance of the randomized and greedy strategies against the number of selected samples n when $K = 500$ and $d = 10$. For very small numbers of selected experiments the performances of the different strategies are equivalent but as the number of experiments increases the randomized E-optimal design outperforms the greedy strategy. Figure A.1b shows the performance of the strategies against the dimension d when $K = 500$ and the number of selected experiments n is fixed to 500. For small dimensions the randomized E-optimal design achieves a better performance but its superiority decreases when the dimension increases. For both settings we also plot the performance of the random strategy that selects experiments uniformly at random. Furthermore, the results are averaged over 100 random seeds controlling the generation of the dataset as well as the random sampling of the experiments.

Application of randomized G-optimal design to best arm identification in linear bandits

We now compare the randomized G-optimal design with the greedy implementation that has been used for the problem of best arm identification in linear bandits. We note that the objective of this experiment is not to achieve state-of-the-art results for best arm identification in linear bandits but rather to show that the randomized strategy while being easy to implement achieves comparable results as the ones obtained with the greedy strategy.

The underlying model of a linear bandit is the same as the one presented in Section A.2: the relationship between the experiments $\mathbf{x}$, referred to as *arms* in the bandit literature, and their associated measurements y is assumed to be linear. The goal of best arm identification (see *e.g.*, [90, 92, 106]) is to find the arm with maximum linear response among a finite set of arms. We focus on the case where one wants to solve this task with a minimum number of trials for a given confidence level. The core idea of most of the developed strategies is to sequentially choose arms so as to minimize a confidence bound on the prediction error of the linear response. Indeed, the sooner we become confident about the predicted response of each arm the sooner we can identify the best one with high probability.

One would like to take advantage of the past responses y when choosing future arms. However the confidence bound that is available for this adaptive setting has a worse dependence on the dimension d than the confidence bound available for fixed sequences of arm [2]. The confidence

bound for fixed sequences can be stated as follows: for all $\delta \in (0,1)$, with probability at least $1-\delta$, for all $n \in \mathbb{N}$ and for all arms $\mathbf{x} \in \mathcal{X}$,

$$|\theta_\star^\top \mathbf{x} - \hat{\theta}_n^\top \mathbf{x}| \leq 2c\|\mathbf{x}\|_{\mathbf{\Sigma}_D^{-1}} \sqrt{\log\left(\frac{6t^2|\mathcal{X}|}{\pi^2\delta}\right)}, \tag{A.6}$$

where $\hat{\theta}_n$ is the OLS estimator obtained with n samples, c is a constant depending on the variance of the Gaussian noise and $\mathbf{\Sigma}_D = \sum_{k=1}^K n_k \mathbf{x}_k \mathbf{x}_k^\top$. It can be observed that designing a strategy that minimizes this confidence bound for all arms naturally leads to the G-optimal design optimization problem. The reader can refer to the supplementary material or [90] for more details.

To compare the randomized G-optimal design with the greedy implementation used for best arm identification in [90] we use the same setting as the one of the experiment presented in Section 6 of [90]. More specifically we consider a set of $d+1$ arms in $\mathbb{R}^d$ where $d \geq 2$. This set is made of the d vectors $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ forming the canonical basis of $\mathbb{R}^d$ and one additional arm $\mathbf{x}_{d+1} = (\cos(\omega), \sin(\omega), 0, \ldots, 0)^\top$ with $\omega = 0.1$. The true parameter $\theta_\star$ has all its coordinates equal to 0 except the first one which is set to 2. In this setting, the best arm, *i.e.*, the one with maximum linear response, is $\mathbf{e}_1$. One can also note that it is much harder to differentiate this arm from $\mathbf{x}_{d+1}$ than from the other arms. The noise of the linear model is a standard Gaussian random variable $\mathcal{N}(0,1)$ and the confidence level in (A.6) is chosen equal to $\delta = 0.05$. We also use the same condition as in [90] (equation (13) therein) to check when enough arms have been pulled to be able to identify the best arm with high probability. This condition naturally derives from the confidence bound (A.6). As explained in Section A.2 the greedy implementation does not work for the first iterations because the design matrix is singular. As in [90] we thus initialize the procedure by choosing once each arm of the canonical basis. Although this would not be required for the randomized strategy as we could start by sampling a given number of experiments, we use the same initialization for the sake of fairness.

The number of samples required to find the best arm are shown in Figure A.1c which summarizes the results obtained over 100 random seeds controlling the Gaussian noise of the linear model and the random selection of the experiments. One can see that the randomized G-optimal design, while being simple to use, achieves similar performances for low dimensions and even better performances on average than the greedy implementation of the G-optimal design as the dimension increases. We note that for all the random repetitions the best arm returned by both strategies is always $\mathbf{e}_1$.

B Randomization matters. How to defend against strong adversarial attacks.

This appendix contains the paper *"Randomization matters, how to defend against strong adversarial attacks"*, International Conference on Machine Learning (ICML) 2020, R. Pinot, R. Ettedgui, G. Rizk, Y. Chevaleyre, J. Atif

Contents

Is there a classifier that ensures optimal robustness against all adversarial attacks? This paper tackles this question by adopting a game-theoretic point of view. We present the adversarial attacks and defenses problem as an *infinite* zero-sum game where classical results (*e.g.* Nash or Sion theorems) do not apply. We demonstrate the non-existence of a Nash equilibrium in our game when the classifier and the Adversary are both deterministic, hence giving a negative answer to the above question in the deterministic regime. Nonetheless, the question remains open in the randomized regime. We tackle this problem by showing that any deterministic classifier can be outperformed by a randomized one. This gives arguments for using randomization, and leads us to a simple method for building randomized classifiers that are robust to state-or-the-art adversarial attacks. Empirical results validate our theoretical analysis, and show that our defense method considerably outperforms Adversarial Training against strong adaptive attacks, by achieving 0.55 accuracy under adaptive PGD-attack on CIFAR10, compared to 0.42 for Adversarial training.

B.1 Introduction

Adversarial example attacks recently became a major concern in the machine learning community. An adversarial attack refers to a small, imperceptible change of an input that is maliciously designed to fool a machine learning algorithm. Since the seminal work of [17] and [91] it became increasingly important to understand the very nature of this phenomenon [23, 37, 38, 46, 50]. Furthermore, a large body of work has been published on designing attacks [9, 27, 45, 66, 71] and defenses [33, 45, 66, 72].

Besides, in real-life scenarios such as for an autonomous car, errors can be very costly. It is not enough to just defend against new attacks as they are published. We would need an algorithm that behaves optimally against every single attack. However, it remains unknown whether such a defense exists. This leads to the following questions, for which we provide principled and theoretically-grounded answers.

Q1: Is there a deterministic classifier that ensures optimal robustness against any adversarial attack?

A1: To answer this question, in Section B.3, we cast the adversarial examples problem as an *infinite* zero-sum game between a Defender (the classifier) and an Adversary that produces adversarial examples. Then we demonstrate, in Section B.4, the non-existence of a Nash equilibrium in the deterministic setting of this game. This entails that no deterministic classifier can claim to be more robust than all other classifiers against any possible adversarial attack. Another consequence of our analysis is that there is no free lunch for transferable attacks: an attack that works on all classifiers will never be optimal against any of them.

Q2: Would randomized defense strategies be a suitable alternative to defend against strong adversarial attacks?

A2: We tackle this problem both theoretically and empirically. In Section B.5, we demonstrate that for any deterministic defense there exists a mixture of classifiers that offers better worst-case theoretical guarantees. Building upon this, we devise a method that generates a robust randomized classifier with a one step boosting method. We evaluate this method, in Section B.6, against strong adaptive attacks on CIFAR10 and CIFAR100 datasets. It outperforms Adversarial Training against both ℓ_∞-**PGD** [66], and ℓ_2-**C&W** [27] attacks. More precisely, on CIFAR10, our algorithm achieves 0.55 (resp. 0.53) accuracy under attack against these attacks, which is an improvement of 0.13 (resp. 0.18) over Adversarial Training.

B.2 Related Work

Many works have studied adversarial examples, in several different settings. We discuss hereafter the different frameworks that we believe to be related to our work, and discuss the aspects on which our contribution differs from them.

Distributionally robust optimization. The work in [88] addresses the problem of adversarial examples through the lens of distributionally robust optimization. They study a min-max problem where the Adversary manipulates the test distribution while being constrained in a Wasserstein distance ball (they impose a global constraint on distributions for the Adversary, while we study a local, pointwise constraint, leading to different attack policies). A similar analysis was presented in [61] in a more general setting that does not focus on adversarial examples. Even though

our work studies a close problem, our reasoning is very different. We adopt a game theoretic standpoint, which allows us to investigate randomized defenses and endow them with strong theoretical evidences.

Game Theory. Some works have tackled the problem of adversarial examples as a two player game. For example [22] views adversarial example attacks and defenses as a Stackelberg game. More recently, [81] and [73] investigated zero-sum games. They consider restricted versions of the game where classical theorems apply, such as when the players only have a finite set of possible strategies. We study a more general setting. Finally, [35] motivates the use of noise injection as a defense mechanism by game theoretic arguments but only present empirical results.

Randomization. Following the work of [35] and [105], several recent works studied noise injection as a defense mechanism. In particular, [60], followed by [33, 62, 75, 102] demonstrated that noise injection can, in some cases, give provable defense against adversarial attacks. The analysis and defense method we propose in this paper are not based on noise injection. However, a link could be made between these works and the mixture we propose, by noting that a classifier in which noise is being injected can be seen as an infinite mixture of perturbed classifiers.

Optimal transport. Our work considers a distributionnal setting, in which the Adversary manipulating the dataset is formalized by a push-forward measure. This kind of setting is close to optimal transport settings recently developed by [16] and [77]. Specifically, these works investigate classifier-agnostic lower bounds on the risk for binary classification under attack, with some hypothesis on the data distribution. The main differences are that we focus on studying equilibria and not deriving bounds. Moreover, these works do not study the influence of randomization. Finally they express the optimal risk of the Defender in terms of transportation costs between two distributions, whereas we explicitly study the Adversary's behaviour as a transport from one distribution to another. Even though they do not treat the problem from the same prism, we believe that these works are profoundly related and complementary to ours.

Ensemble of classifiers. Some works have been done to improve the robustness of a model by constructing ensemble of classifiers [1, 70, 86, 100, 107]. However all the defense methods proposed in those papers subsequently proved to be ineffective against adaptive attacks introduced in [48, 94]. The main difference with our method is that it is not an ensemble method since it uses sampling instead of voting to aggregate the classifiers' output. Hence in terms of volatility, in voting methods, whenever a majority agrees on an opinion, all others votes will be ignored, whereas here each classifier always contributes according to its probability weights, which do not depend on the others.

B.3 A Game Theoretic point of view.

Initial problem statement

Notations. For any set $\mathcal{Z}$ with σ-algebra $\sigma(\mathcal{Z})$, if there is no ambiguity on the considered σ-algebra, we denote $\mathcal{P}(\mathcal{Z})$ the set of all probability measures over $(\mathcal{Z}, \sigma(\mathcal{Z}))$, and $\mathcal{F}_{\mathcal{Z}}$ the set of all measurable functions from $(\mathcal{Z}, \sigma(\mathcal{Z}))$ to $(\mathcal{Z}, \sigma(\mathcal{Z}))$. For $\mu \in \mathcal{P}(\mathcal{Z})$ and $\phi \in \mathcal{F}_{\mathcal{Z}}$, the *pushforward measure* of μ by ϕ is the measure $\phi\#\mu$ such that $\phi\#\mu(B) = \mu(\phi^{-1}(B))$ for any $B \in \sigma(\mathcal{Z})$.

Binary classification task. Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{\text{-}1, 1\}$. We consider a distribution $\mathcal{D} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ that we assume to be of support $\mathcal{X} \times \mathcal{Y}$. The Defender is looking for a hypothesis (classifier) h in a class of functions $\mathcal{H}$, minimizing the risk of h w.r.t. $\mathcal{D}$:

$$\begin{aligned} \mathcal{R}(h) :&= \mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}}[\mathbb{1}\{h(X) \neq Y\}] \\ &= \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\mu_Y}[\mathbb{1}\{h(X) \neq Y\}]\right]. \end{aligned} \tag{B.1}$$

Where $\mathcal{H} := \{h : x \mapsto \operatorname{sgn} g(x) \mid g : \mathcal{X} \to \mathbb{R} \text{ continuous}\}$, $\nu \in \mathcal{P}(\mathcal{Y})$ is the probability measure that defines the law of the random variable Y, and for any $y \in \mathcal{Y}$, $\mu_y \in \mathcal{P}(\mathcal{X})$ is the conditional law of $X|(Y = y)$.

Adversarial example attack (point-wise). Given a classifier $h : \mathcal{X} \to \mathcal{Y}$ and a data sample $(x, y) \sim \mathcal{D}$, the Adversary seeks a perturbation $\tau \in \mathcal{X}$ that is visually imperceptible, but modifies x enough to change its class, *i.e.* $h(x + \tau) \neq y$. Such a perturbation is called an *adversarial example attack*. In practice, it is hard to evaluate the set of visually imperceptible modifications of an image. However, a sufficient condition to ensure that the attack is undetectable is to constrain the perturbation τ to have a small norm, be it for the ℓ_∞ or the ℓ_2 norm. Hence, one should always ensure that $\|\tau\|_\infty \leq \epsilon_\infty$, or $\|\tau\|_2 \leq \epsilon_2$, depending on the norm used to measure visual imperceptibility. The choice of the threshold depends on the application at hand. For example, on CIFAR datasets, typical values for ϵ_∞ and ϵ_2 are respectively, 0.031 and $0.4/0.6/0.8$. In the remaining of this work, we will define our constraint using an ℓ_2 norm, but all our results are valid for an ℓ_∞ based constraint.

Adversarial example attack (distributional). The Adversary chooses, for every $x \in \mathcal{X}$, a perturbation that depends on its true label y. This amounts to construct, for each label $y \in \mathcal{Y}$, a measurable function ϕ_y such that $\phi_y(x)$ is the perturbation associated with the labeled example (x, y). This function naturally induces a probability distribution over adversarial examples, which is simply the push-forward measure $\phi_y \# \mu_y$. The goal of the Adversary is thus to find $\boldsymbol{\phi} = (\phi_{\text{-}1}, \phi_1) \in (\mathcal{F}_{\mathcal{X}|\epsilon_2})^2$ that maximizes the adversarial risk $\mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi})$ defined as follows:

$$\mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi}) := \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\phi_Y\#\mu_Y}[\mathbb{1}\{h(X) \neq Y\}]\right]. \tag{B.2}$$

Where for any $\epsilon_2 \in (0, 1)$, $\mathcal{F}_{\mathcal{X}|\epsilon_2}$ is the set of functions that imperceptibly modifies a distribution:

$$\mathcal{F}_{\mathcal{X}|\epsilon_2} := \left\{\psi \in \mathcal{F}_{\mathcal{X}} \mid \operatorname*{essup}_{x\in\mathcal{X}} \|\psi(x) - x\|_2 \leq \epsilon_2\right\}.$$

Adversarial defense, a two-player zero-sum game. With the setting defined above, the adversarial examples problem can be seen as a two-player zero-sum game, where the Defender tries to find the best possible hypothesis h, while a strong Adversary is manipulating the dataset distribution:

$$\inf_{h\in\mathcal{H}} \sup_{\boldsymbol{\phi}\in\left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}(h, \boldsymbol{\phi}). \tag{B.3}$$

This means that the Defender tries to design the classifier with the best performance under attack, whereas the Adversary will each time design the optimal attack on this specific classifier. In the game theoretical terminology, the choice of a classifier h (resp. an attack ϕ) for the Defender (resp. the Adversary) is called a *strategy*. It is crucial to note that the sup-inf and inf-sup problems do not necessarily coincide. In this paper, we mainly focus on the Defender's point of view which corresponds to the inf-sup problem. We will be interested in understanding the behaviour of players in this game, *i.e.* the best responses they have to a given strategy, and whether some equilibria may arise. This motivates the following definitions.

Definition B.1 (Best Response)**.** *Let $h \in \mathcal{H}$, and $\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. A* best response *from the Defender to ϕ is a classifier $h^* \in \mathcal{H}$ such that $\mathcal{R}_{\text{adv}}(h^*, \phi) = \min_{h \in \mathcal{H}} \mathcal{R}_{\text{adv}}(h, \phi)$. Similarly, a* best response *from the Adversary to h is an attack $\phi^* \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$ such that $\mathcal{R}_{\text{adv}}(h, \phi^*) = \max_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}(h, \phi)$.*

In the remaining, we denote $\mathfrak{BR}(h)$ the set of all best responses of the Adversary to a classifier h. Similarly $\mathfrak{BR}(\phi)$ denotes the set of best responses to an attack ϕ.

Definition B.2 (Pure Nash Equilibrium)**.** *In the zero-sum game (Eq. B.3), a* Pure Nash Equilibrium *is a couple of strategies $(h, \phi) \in \mathcal{H} \times \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$ such that*

$$\begin{cases} h & \in \mathfrak{BR}(\phi), \text{ and,} \\ \phi & \in \mathfrak{BR}(h). \end{cases}$$

When it exists, a Pure Nash Equilibrium is a state of the game in which no player has any incentive to modify its strategy. In our setting, this simultaneously means that no attack could better fool the current classifier, and that the classifier is optimal for the current attack.

Remark. All the definitions in this section assume a deterministic regime, *i.e.* that neither the Defender nor the Adversary use randomization, hence the notion of *Pure* Nash Equilibrium in the game theory terminology. The randomized regime will be studied in Section B.5.

Trivial solution and Regularized Adversary

Trivial Nash equilibrium. Our current definition of the problem implies that the Adversary has perfect information on the dataset distribution and the classifier. It also has unlimited computational power and no constraint on the attack except on the size of the perturbation. Going back to the example of the autonomous car, this would mean that the Adversary can modify every single image that the camera *may* receive during *any* trip, which is highly unrealistic. The Adversary has no downside to attacking, even when the attack is unnecessary, *e.g.* if the attack cannot work or if the point is already misclassified.

This type of behavior for the Adversary can lead to the existence of a pathological (and trivial) Nash Equilibrium as demonstrated in Figure B.1 for the uni-dimensional setting with Gaussian distributions. The unbounded Adversary moves every point toward the decision boundary (each time maximizing the perturbation budget), and the Defender cannot do anything to mitigate the damage. In this case the decision boundary for the Optimal Bayes Classifier remains unchanged,

even though both curves have been moved toward the center, hence a trivial equilibrium. In the remaining of this work, we show that such an equilibrium does not exist as soon as there is a small restraint on the Adversary's strength, *i.e.* as soon as it is not perfectly indifferent to produce unnecessary perturbations.

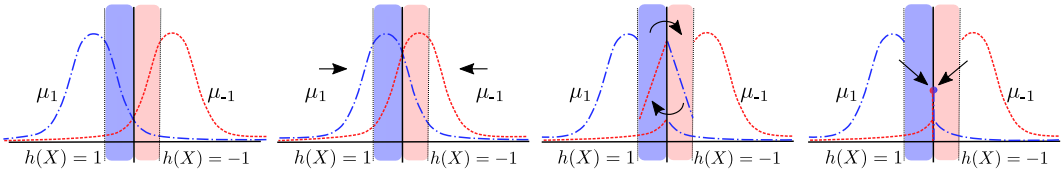


Figure B.1: Representation of the μ_{-1} (blue dotted line) and μ_1 (red plain line) distributions, without attack (left) and with three different attacks: no penalty (second drawing), with mass penalty (third) and with norm penalty (fourth). On all figures blue area on the left of the axis is $P_h(\epsilon_2)$ and red area on the right is $N_h(\epsilon_2)$.

Regularized Adversary. To mitigate the Adversary strength, we introduce a penalization term:

$$\inf_{h\in\mathcal{H}} \sup_{\phi\in\left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \underbrace{\left[\mathcal{R}_{\text{adv}}(h,\phi) - \lambda\,\Omega(\phi)\right]}_{\mathcal{R}^{\Omega}_{\text{adv}}(h,\phi)}. \tag{B.4}$$

The penalty function Ω represents the limitations on the Adversary's budget, be it because of computational resources or to avoid being detected. $\lambda \in (0,1)$ is some regularization weight. In this paper, we study two types of penalties: the *mass penalty* Ω_{mass}, and the *norm penalty* Ω_{norm}.

From a computer-security point of view, the first limitation that comes to mind is to limit the number of queries the Adversary can send to the classifier. In our distributional setting, this boils down to penalizing the mass of points that the function ϕ moves. Hence we define the mass penalty as:

$$\Omega_{\text{mass}}(\phi) := \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\mu_Y}\left[\mathbb{1}\{X \neq \phi_Y(X)\}\right]\right]. \tag{B.5}$$

The mass penalty discourages the Adversary from attacking too many points by penalizing the overall mass of transported points. The second limitation we consider penalizes the expected norm under ϕ:

$$\Omega_{\text{norm}}(\phi) := \mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\mu_Y}\left[\|X - \phi_Y(X)\|_2\right]\right]. \tag{B.6}$$

This regularization is very common in both the optimization and adversarial example communities. In particular, it is used by Carlini & Wagner [27] to compute the eponymous attack[1]. In the following, we denote $\mathfrak{BR}_{\Omega_{\text{mass}}}$ (resp. $\mathfrak{BR}_{\Omega_{\text{norm}}}$) the best responses for the Adversary w.r.t the mass (resp. norm) penalty. Section B.4 shows that whatever penalty the Adversary has, no Pure

[1] Ω_{norm} is not limited to ℓ_2 norm. The results we present hold as long as the norm used to compare X and $\phi_Y(X)$ comes from a scalar product on $\mathcal{X}$.

Nash Equilibrium exists. We characterize the best responses for each player, and show that they can never satisfy Definition B.2.

B.4 Deterministic regime

Notations. Let $h \in \mathcal{H}$, we denote $P_h := \{x \in \mathcal{X} \mid h(x) = 1\}$, and $N_h := \{x \in \mathcal{X} \mid h(x) = \text{-}1\}$ respectively the set of positive and negative outputs of h. We also denote the set of attackable points from the positive outputs $P_h(\delta) := \{x \in P_h \mid \exists z \in N_h \text{ and } \|z - x\|_2 \leq \delta\}$, and $N_h(\delta)$ likewise.

Adversary's best response. Let us first present the best responses of the Adversary under respectively the mass penalty and the norm penalty. Both best responses share a fundamental behavior: the optimal attack will only change points that are close enough to the decision boundary. This means that, when the Adversary has no chance of making the classifier change its decision about a given point, it will not attack it. However, for the norm penalty all attacked points are projected on the decision boundary, whereas with the mass penalty the attack moves the points across the border.

Lemma B.1. *Let $h \in \mathcal{H}$ and $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{mass}}(h)$. Then the following assertion holds:*

$$\begin{cases} \phi_1(x) \in (P_h)^{\complement} & \text{if } x \in P_h(\epsilon_2) \\ \phi_1(x) = x & \text{otherwise.} \end{cases}$$

Where $(P_h)^{\complement}$, the complement of P_h in $\mathcal{X}$. $\phi_{\text{-}1}$ is characterized symmetrically.

Lemma B.2. *Let $h \in \mathcal{H}$ and $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{norm}}(h)$. Then the following assertion holds:*

$$\phi_1(x) = \begin{cases} \pi(x) & \text{if } x \in P_h(\epsilon_2) \\ x & \text{otherwise.} \end{cases}$$

Where π is the orthogonal projection on $(P_h)^{\complement}$. $\phi_{\text{-}1}$ is characterized symmetrically.

These best responses are illustrated in Figure B.1 with two uni-dimensional Gaussian distributions. For the mass penalty, μ_1 is set to 0 in $P_h(\epsilon_2)$, and this mass is transported into $N_h(\epsilon_2)$. The symmetric holds for $\mu_{\text{-}1}$. After attack, we now have $\mu_1(P_h(\epsilon_2)) = 0$, so a small value of $\mu_{\text{-}1}$ in $P_h(\epsilon_2)$ suffices to make it dominant, and that zone will now be classified -1 by the Optimal Bayes Classifier. For the norm penalty, the part of μ_1 that was in $P_h(\epsilon_2)$ is transported on a Dirac distribution at the decision boundary. Similarly to the mass penalty, the best response now predicts -1 for the zone $P_h(\epsilon_2)$.

Remark. In practice, it might be computationally hard to generate the exact best response for the norm penalty, *i.e.* the projection on the decision boundary. That will happen for example if this boundary is very complex (*e.g.* highly non-smooth), or when $\mathcal{X}$ is in a high dimensional space. To keep the attack tractable, the Adversary will have to compute an approximated best response by allowing the projection to reach the point within a small ball around the boundary. This means that the best responses of the norm penalty and the mass penalty problems will often match.

Defender's best response. At a first glance, one would suspect that the best response for the Defender ought to be the Optimal Bayes Classifier for the transported distribution. However, it is only well defined if the conditional distributions admit a probability density function. This might not always hold here for the transported distribution. Nevertheless, we show that there is a property, shared by the Optimal Bayes Classifier when defined, that always holds for the Defender's best response.

Lemma B.3. *Let us consider* $\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. *If we take* $h \in \mathfrak{BR}(\phi)$, *then for* $y = 1$ *(resp.* $y = -1$*), and for any* $B \subset P_h$ *(resp.* $B \subset N_h$*) one has*

$$\mathbb{P}(Y = y | X \in B) \geq \mathbb{P}(Y = -y | X \in B)$$

with $Y \sim \nu$ *and for all* $y \in \mathcal{Y}$, $X|(Y = y) \sim \phi_y \# \mu_y$.

In particular, when $\phi_1 \# \mu_1$ and $\phi_{-1} \# \mu_{-1}$ admit probability density functions, Lemma B.3 simply means that h is the Optimal Bayes Classifier for the distribution $(\nu, \phi_1 \# \mu_1, \phi_{-1} \# \mu_{-1})$[2]. We can now state our main theorem, as well as two of its important consequences.

Theorem B.1 (Non-existence of a pure Nash equilibrium)**.** *In the zero-sum game (Eq. B.4) with* $\lambda \in (0, 1)$ *and penalty* $\Omega \in \{\Omega_{mass}, \Omega_{norm}\}$, *there is no Pure Nash Equilibrium.*

Consequence 1. (*No free lunch for transferable attacks*) To understand this statement, remark that, thanks to weak duality, the following inequality always holds:

$$\sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \inf_{h \in \mathcal{H}} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi) \leq \inf_{h \in \mathcal{H}} \sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi).$$

On the left side problem (sup-inf), the Adversary looks for the best strategy ϕ against any *unknown* classifier. This is tightly related to the notion of *transferable attacks* (see *e.g.* [95]), which refers to attacks successful against a wide range of classifiers. On the right side (our) problem (inf-sup), the Defender tries to find the best classifier under any possible attack, whereas the Adversary plays in second and specifically attacks this classifier. As a consequence of Theorem B.3, the inequality is always strict:

$$\sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \inf_{h \in \mathcal{H}} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi) < \inf_{h \in \mathcal{H}} \sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}^{\Omega}_{\text{adv}}(h, \phi).$$

This means that both problems are not equivalent. In particular, an attack designed to succeed against *any* classifier (*i.e.* a transferable attack) will not be as good as an attack tailored for a given classifier. Hence she has to trade-off between effectiveness and transferability of the attack.

Consequence 2. (*No deterministic defense may be proof against every attack*) Let us consider the state-of-the-art defense which is Adversarial Training [45, 66]. The idea is to compute an efficient attack ϕ, and train the classifier on created adversarial examples, in order to move the decision boundary and make the classifier more robust to new perturbations by ϕ.

[2] We prove this result in the supplementary material.

To be fully efficient, this method requires that ϕ remains an optimal attack on h even after training. Our theorem shows that it is never the case: after training our classifier h to become (h') robust against ϕ, there will always be a different optimal attack ϕ' that is efficient against h'. Hence Adversarial Training will never achieve a perfect defense.

B.5 Randomization matters

As we showed that there is no Pure Nash Equilibrium, no deterministic classifier may be proof against every attack. We would therefore need to allow for a wider class of strategies. A natural extension of the game would thus be to allow randomization for both players, who would now choose a distribution over pure strategies, leading to this game:

$$\inf_{\eta \in \mathcal{P}(\mathcal{H})} \sup_{\varphi \in \mathcal{P}\left(\left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2\right)} \mathop{\mathbb{E}}_{\substack{h \sim \eta \\ \phi \sim \varphi}} \left[\mathcal{R}^{\Omega}_{\text{adv}}(h, \phi)\right]. \tag{B.7}$$

Without making further assumptions on this game (e.g. compactness), we cannot apply known results from game theory (e.g. Sion theorem) to prove the existence of an equilibrium. These assumptions would however make the problem loose much generality, and do not hold here.

Randomization matters. Even without knowing if an equilibrium exists in the randomized setting, we can prove that *randomization matters*. More precisely we show that any deterministic classifier can be outperformed by a randomized one in terms of the worst case adversarial risk. To do so we simplify Equation B.7 in two ways:

1. We do not consider the Adversary to be randomized, *i.e.* we restrict the search space of the Adversary to $(\mathcal{F}_{\mathcal{X}})^2$ instead of $\mathcal{P}\left((\mathcal{F}_{\mathcal{X}})^2\right)$. This condition corresponds to the current state-of-the-art in the domain: to the best of our knowledge, no efficient randomized adversarial example attack has been designed (and so is used) yet.

2. We only consider a subclass of randomized classifiers, called mixtures, which are discrete probability measures on a finite set of classifiers. We show that this kind of randomization is enough to strictly outperform any deterministic classifier. We will discuss later the use of more general randomization (such as noise injection) for the Defender. Let us now define a mixture of classifiers.

Definition B.3 (Mixture of classifier)**.** *Let $n \in \mathbb{N}$, $\mathbf{h} = (h_1, ..., h_n) \in \mathcal{H}^n$, and $\mathbf{q} \in \mathcal{P}(\{1, ..., n\})$. A* mixed classifier of $\mathbf{h}$ by $\mathbf{q}$ *is a mapping $m^{\mathbf{q}}_{\mathbf{h}}$ from $\mathcal{X}$ to $\mathcal{P}(\mathcal{Y})$ such that for all $x \in \mathcal{X}$, $m^{\mathbf{q}}_{\mathbf{h}}(x)$ is the discrete probability distribution that is defined for all $y \in \mathcal{Y}$ as follows:*

$$m^{\mathbf{q}}_{\mathbf{h}}(x)(y) := \mathop{\mathbb{E}}_{i \sim \mathbf{q}} [\mathbb{1}\{h_i(x) = y\}].$$

We call such a mixture a *mixed strategy* of the Defender. Given some $x \in \mathcal{X}$, this amounts to picking a classifier h_i from $\mathbf{h}$ at random following the distribution $\mathbf{q}$, and use it to output the predicted class for x, *i.e.* $h_i(x)$. Note that a mixed strategy for the Defender is a non deterministic algorithm, since it depends on the sampling one makes on $\mathbf{q}$. Hence, even if the attacks are defined

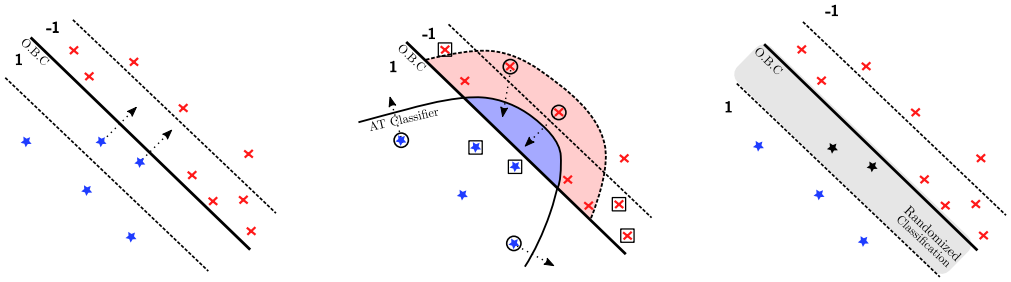


Figure B.2: Illustration of adversarial examples (only on class 1 for more readability) crossing the decision boundary (left), adversarially trained classifier for the class 1 (middle), and a randomized classifier that defends class 1. Stars are natural examples for class 1, and crosses are natural examples for class -1. The straight line is the optimal Bayes classifier, and dashed lines delimit the points close enough to the boundary to be attacked resp. for class 1 and -1. We focus the drawing on the star points. Crosses can be treated symmetrically.

in the same way as before, the Adversary now needs to maximize a new objective function which is the expectation of the adversarial risk under the distribution $m_{\mathbf{h}}^{\mathbf{q}}$. It writes as follows:

$$\mathop{\mathbb{E}}_{Y\sim\nu}\left[\mathop{\mathbb{E}}_{X\sim\phi_Y\#\mu_Y}\left[\mathop{\mathbb{E}}_{\hat{Y}\sim m_{\mathbf{h}}^{\mathbf{q}}(X)}\left[\mathbb{1}\left\{\hat{Y}\neq Y\right\}\right]\right]\right]-\lambda\,\Omega(\phi). \tag{B.8}$$

We also write $\mathcal{R}_{\text{adv}}^{\Omega}$ to mean the left part of Equation (B.8), when it is clear from context that the Defender uses a mixed classifier. Using this new set of strategies for the Defender, we can study whether mixed classifiers outperform deterministic ones, and how to efficiently design them.

Mixed strategy. We demonstrate that the efficiency of any deterministic defense can be improved using a simple mixed strategy. This method presents similarities with the notions of fictitious play [21] in game theory, and boosting in machine learning [42]. Given a deterministic classifier h_1, we combine it (via randomization) with the best response h_2 to its optimal attack.

The rational behind this idea is that, by construction, efficient attacks on one of these two classifiers will not work on the other. Mixing h_1 with h_2 has two opposite consequences on the adversarial risk. On one hand, where we only had to defend against attack on h_1, we are now also vulnerable to attacks on h_2, so the total set of possible attacks is now bigger. On the other hand, each attack will only work part of the time, depending on the probability distribution $\mathbf{q}$. If we can calibrate the weights so that attacks on important zones have a low probability of succeeding, then the average risk under attack on the mixture will be low.

Toy example where a mixture outperforms AT. To better understand how randomization can work, let us look at a simple toy example. Figure B.2 illustrates a binary classification setting between two set of points. Attacking the Optimal Bayes Classifier (bold straight line) consists in moving all the points that lie between the dotted lines to the opposite side of the decision boundary (Figure B.2, left). The general tactic to defend against an attack is to change the classifier's output for points that are too close to the boundary. This can be done all the time, as in Adversar-

ial Training (where we move the decision boundary to incorporate adversarial examples), or part of the time as in a randomized algorithm (so that the attack only works with a given probability).

When we use Adversarial Training for the star points (Figure B.2, middle), we change the output on the blue zone, so that 2 of the star (squared) points cannot be successfully attacked anymore. But in exchange, the dilation of the new boundary can now be attacked. For Adversarial Training to work, we need the number of new potential attacks (*i.e.* the points that are circled, 2 crosses in the dilation and 2 stars that are close to the new boundary) to be smaller than the number of attacks we prevent (the squared points, 2 blue ones that an attack would send in the blue zone, and 3 red points that are far from the new decision boundary). Here we prevent 5 attacks at the cost of 4 new ones, so the Adversarial Training improves the total score from 8 to 7.

Similarly, we observe what happens for the randomized defense (Figure B.2, right). We mix the Optimal Bayes Classifier with the best response to attacking all the points. We get a classifier that is determinsitic outside the gray area, and random inside it[3]. If the first classifier has a weight $\alpha = 0.5$, 6 of the old attacks now succeed only with probability 0.5 (crosses between the dotted lines), whereas 3 new attacks are created (stars outside of the gray area) that succeed with probability 0.5 also. At the end, the average rate of successful attacks is 6.5, where adversarial training previously achieved 7.

More formally, Theorem B.4 shows that whatever penalty we consider, a deterministic classifier can always be outperformed by a randomized algorithm. We now can state our second main result: randomization matters.

Theorem B.2. *(Randomization matters) Let us consider $h_1 \in \mathcal{H}$, $\lambda \in (0,1)$, $\Omega = \Omega_{mass}$, $\phi \in \mathfrak{BR}_\Omega(h_1)$ and $h_2 \in \mathfrak{BR}(\phi)$. Then for any $\alpha \in (\max(\lambda, 1-\lambda), 1)$ and for any $\phi' \in \mathfrak{BR}_\Omega(m^{\mathbf{q}}_{\mathbf{h}})$ one has*

$$\mathcal{R}^{\Omega}_{\text{adv}}(m^{\mathbf{q}}_{\mathbf{h}}, \phi') < \mathcal{R}^{\Omega}_{\text{adv}}(h_1, \phi).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1-\alpha)$, and $m^{\mathbf{q}}_{\mathbf{h}}$ is the mixture of $\mathbf{h}$ by $\mathbf{q}$. A similar result holds when $\Omega = \Omega_{norm}$ (see supplementary materials).

Remark Note that depending on the initial hypothesis h_1 and the conditional distributions μ_1 and μ_{-1}, the gap between $\mathcal{R}^{\Omega}_{\text{adv}}(m^{\boldsymbol{q}}_{\boldsymbol{h}}, \phi')$and $\mathcal{R}^{\Omega}_{\text{adv}}(h_1, \phi)$ could vary. Hence, with additional conditions on h_1, μ_1 and μ_{-1}, we could make the gap appear more explicitly. We keep the formulation general to emphasize that for *any* deterministic classifier, there exists a randomized one that outperforms it in terms of worst-case adversarial score.

Based on Theorem B.4 we devise a new procedure called Boosted Adversarial Training (BAT) to construct a robust mixture of two classifiers. It is based on three core principles: Adversarial Training, Boosting and Randomization.

B.6 Experiments: How to build the mixture

Simple mixture procedure (BAT). Given a dataset D and a weight parameter $\alpha \in [0,1]$, we construct h_1 the first classifier of the mixture using Adversarial Training[4] on D. Then, we train

[3] The grey area should actually be bigger since the best response to the attack would also change the decision on the upper part between the OBC and the doted line. We focus on what happens on the star points for simplicity.

[4] We use ℓ_∞-**PGD** with 20 iterations and $\epsilon_\infty = 0.031$ to train the first classifier and to build $\tilde{D}$.

Dataset	Method	Natural Accuracy	Adaptive-l_∞-PGD $\epsilon_\infty = 0.031$	Adaptive-ℓ_2-C&W $\epsilon_2 = 0.4$	$\epsilon_2 = 0.6$	$\epsilon_2 = 0.8$
CIFAR10	Natural	0.88	0.00	0.00	0.00	0.00
	AT [66]	0.83	0.42	**0.60**	0.47	0.35
	Ours	0.80	**0.55**	**0.60**	**0.57**	**0.53**
CIFAR100	Natural	0.62	0.00	0.00	0.00	0.00
	AT [66]	0.58	0.26	0.38	0.29	0.22
	Ours	0.56	**0.40**	**0.45**	**0.41**	**0.38**

Table B.1: Evaluation on CIFAR10 and CIFAR100 without *data augmentation*. Accuracy under attack of a single adversarially trained classifier (AT) and the mixture formed with our method (Ours). The evaluation is made with **Adaptive-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W** attacks both computed with 100 iterations. For **Adaptive-ℓ_∞-PGD** we use an epsilon equal to $8/255$ (≈ 0.031), a step size equal to $2/255$ (≈ 0.008) and we allow random initialization. For **Adaptive-ℓ_2-C&W** we use a learning rate equal to 0.01, 9 binary search steps, the initial constant to 0.001, we allow the abortion when it has already converged and we give the results for the different values of rejection threshold $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. As for EOT, we don't need to estimate the expected accuracy of the mixture through Monte Carlo sampling since we have the exact weight of each classifier of the mixture. Thus we give the exact expected accuracy.

the second classifier h_2 on a data set $\tilde{D}$ that contains adversarial examples against h_1 created from examples of D. At the end we return the mixture constructed with those two classifiers where the first one has a weight of $1 - \alpha$ and the second one a weight of α. The parameter α is found by conducting a grid-search. In Table B.1 we present results for $\alpha = 0.2$ under strong state-of-the-art attacks. The procedure is summarized in Algorithm 12[5]

Algorithm 11: Boosted Adversarial Training

Input : D the training data set and α the weight parameter.

Create and adversarially train h_1 on D
Generate the adversarial data set $\tilde{D}$ against h_1.
Create and naturally train h_2 on $\tilde{D}$
$\mathbf{q} \leftarrow (1 - \alpha, \alpha)$
$\mathbf{q} \leftarrow (h_1, h_2)$

return $m_{\mathbf{h}}^{\mathbf{q}}$

Comparison to fictitious play. Contrary to classical algorithms such as *Fictitious play* that also generates mixtures of classifiers, and whose theoretical guarantees rely on the existence of a Mixed Nash Equilibrium, the performance of our method is ensured by Theorem B.4 to be at least as good as the classifier it uses as a basis. Moreover, the implementation of Fictitious Play would be impractical on the high dimensional datasets we consider, due to its computational costs.

[5] More algorithmic and implementation details can be found in the supplementary materials.

Evaluating against strong adversarial attacks. When evaluating a defense against adversarial examples, it is crucial to test the robustness of the method against the best possible attack. Accordingly, the defense method should be evaluated against attacks that were specifically tailored to it (a.k.a. adaptive attacks). In particular, when evaluating randomized algorithms, one should use Expectation over Transformation (EOT) to avoid gradient masking as pointed out by [9] and [26]. More recently, [94] emphasized that one should also make sure that EOT is computed properly[6]. Previous works such as [35] and [75] estimate the EOT through a Monte Carlo sampling which can introduce a bias in the attack if the sample size is too small. Since we assume perfect information for the Adversary, it knows the exact distribution of the mixture. Hence it can directly compute the expectation without using a sampling method, which avoid any bias. Table B.1 evaluates our method against strong adaptive attacks namely **Adaptive-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W**.

Hard constraint parameter. The typical value of ϵ in the hard constraint depends on the norm we consider in the problem setting. In this paper, we use an ℓ_2 norm, however, the constraint parameter for ℓ_∞-**PGD** attack was initially set to be an ℓ_∞ constraint. In order to compare attacks of similar strength, we choose different threshold (ϵ_2 or ϵ_∞) values which result in balls of equivalent volumes. For CIFAR10 an CIFAR100 datasets [55], which are $3 \times 32 \times 32$ dimensional spaces, this gives $\epsilon_\infty = 0.03$ and $\epsilon_2 = 0.8$ (we also give results for ϵ_2 equal to 0.6 and 0.4 as this values are sometimes used in the literature). Since **Adaptive-ℓ_2-C&W** attack creates an unbounded perturbation on the examples, we implemented the constraint from Equation B.6 by checking at test time whether the ℓ_2-norm of the perturbation exceeds a certain threshold $\epsilon_2 \in \{0.4, 0.6, 0.8\}$. If it does, the adversarial example is disregarded, and we keep the natural example instead.

Experimental results. In Table B.1 we compare the accuracy, on CIFAR10 and CIFAR100, of our method and classical Adversarial Training under attack with **Adaptive-ℓ_∞-PGD** and **Adaptive-ℓ_2-C&W**, both run for 100 iterations. We used 5 times more iterations for the evaluation as we used during training, and carefully check for convergence. the rational behind this is that, for a classifier to be fully robust, its loss of accuracy should be controlled when the attacks are stronger than the ones it was trained on. For both attacks, both datasets and all thresholds (i.e. *the budget for a perturbation*), the accuracy under attack of our mixture is higher than the single classifier with Adversarial Training. Our defense is especially more robust than Adversarial Training when the threshold is high.

Extension to more than two classifiers. In this paper we focus our experiments on a mixture of two classifiers to present a proof of concept of Theorem B.4. Nevertheless, a mixture of more than two classifiers can be constructed by adding at each step t a new classifier trained naturally on the dataset $\tilde{D}$ that contains adversarial examples against the mixture at step $t-1$. Since $\tilde{D}$ has to be constructed from a mixture, one would have to use an adaptive attack as **Adaptive-ℓ_∞-PGD**. We refer the reader to the supplementary material for this extended version of the algorithm and for all the implementation details related to our experiments (architecture of models, optimization settings, hyper-parameters, etc.).

[6] In order for the attack to succeed, it it more efficient to compute the expected transformation of the logits instead of taking the expectation over the loss. More details on this in the supplementary materials.

B.7 Discussion & Conclusion

Finally, is there a classifier that ensures optimal robustness against all adversarial attacks? We gave a negative answer to this question in the deterministic regime, but part of the question remains open when considering randomized algorithms. We demonstrated that randomized defenses are more efficient than deterministic ones, and devised a simple method to implement them.

Game theoretical point of view. There remains to study whether an Equilibrium exists in the Randomized regime. This question is appealing from a theoretical point of view, and requires to investigate the space of randomized Adversaries $\mathcal{P}((\mathcal{F}_{\mathcal{X}})^2)$. The characterization of this space is not straightforward, and would require strong results in the theory of optimal transport. A possible research direction is to quotient the space $(\mathcal{F}_{\mathcal{X}})^2$ so as to simplify the search in $\mathcal{P}((\mathcal{F}_{\mathcal{X}})^2)$ and the characterization of the Adversary's best responses. The study of this equilibrium is tightly related to that of the value of the game, which would be interesting for obtaining min-max bounds on the accuracy under attack, as well as certificates of robustness for a set of classifiers.

Advocating for more provable defenses. Although the experimental results show that our mixture of classifiers outperforms Adversarial Training, our algorithm does not provide guarantees in terms of certified accuracy. As the literature on adversarial attacks and defenses demonstrated, better attacks always exist. This is why, more theoretical works need to be done to prove the robustness of a mixture created from this particular algorithm. More generally, our work advocates for the study of mixtures as a provable defense against adversarial attacks. One could, for example, build upon the connection between mixtures and noise injection to investigate a broader range of randomized strategies for the Defender, and devise certificates accordingly.

Improving Boosted Adversarial Training. From an algorithmic point of view, BAT can be improved in several ways. For instance, the weights can be learned while choosing the new classifier for the mixture. This could lead to an improved accuracy under attack, but would lack some theoretical justifications that still need to be set up. Finally, tighter connections with standard boosting algorithms could be established to improve the analysis of BAT.

B.8 Omitted proofs and Additional results

Notations. Let us suppose that $(\mathcal{X},\|.\|)$ is a normed vector space. $B_{\|.\|}(x,\epsilon) = \{z \in \mathcal{X} \mid \|x-z\| \leq \epsilon\}$ is the closed ball of center x and radius ϵ for the norm $\|.\|$. Note that $\mathcal{H} := \{h : x \mapsto \operatorname{sgn} g(x) \mid g : \mathcal{X} \to \mathbb{R} \text{ continuous}\}$, with sgn the function that outputs 1 if $g(x) > 0$, -1 if $g(x) < 0$, and 0 otherwise. Hence for any $(x,y) \sim D$, and $h \in \mathcal{H}$ one has $\mathbb{1}\{h(x) \neq y\} = \mathbb{1}\{g(x)y \leq 0\}$. Finally, we denote ν_1 and $\nu_{\text{-}1}$ respectively the probabilities of class 1 and -1.

Introducing remarks. Let us first note that in the paper, the penalties are defined with an ℓ_2 norm. However, Lemma B.1 and B.2 hold as long as $\mathcal{X}$ is an Hilbert space with dot product $<|>$ and associated norm $||.|| = \sqrt{<.\mid.>}$. We first demonstrate Lemma B.2 with these general notations. Then we present the proof of Lemma B.1 that follows the same schema. Note that, for Lemma B.1, we do not even need the norm to be Hilbertian, since the core argument rely on separation property of the norm, *i.e.* on the property $\|x-y\| = 0 \iff x = y$.

Lemma B.2. *Let $h \in \mathcal{H}$ and $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{norm}}(h)$. Then the following assertion holds:*

$$\phi_1(x) = \begin{cases} \pi(x) & \text{if } x \in P_h(\epsilon_2) \\ x & \text{otherwise.} \end{cases}$$

Where π is the orthogonal projection on $(P_h)^{\complement}$. ϕ_{-1} is characterized symmetrically.

Proof. Let us first simplify the worst case adversarial risk for h. Recall that $h = \operatorname{sgn}(g)$ with g continuous. From the definition of adversarial risk we have:

$$\begin{aligned}
&\sup_{\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(h, \boldsymbol{\phi}) \\
&= \sup_{\boldsymbol{\phi} \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathop{\mathbb{E}}_{X \sim \mu_y} \left[\mathbb{1}\{h(\phi_y(X)) \neq y\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\} \right] \\
&= \sup_{\boldsymbol{\phi} \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathop{\mathbb{E}}_{X \sim \mu_y} \left[\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\} \right] \\
&= \sum_{y=\pm 1} \nu_y \sup_{\phi_y \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_y} \left[\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \|X - \phi_y(X)\| - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\} \right]
\end{aligned}$$

Finding ϕ_1 and ϕ_1 are two independent optimization problems, hence, we focus on characterizing ϕ_1 (*i.e.* $y = 1$).

$$\begin{aligned}
&\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_1} \left[\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \|X - \phi_1(X)\| - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\} \right] \\
&= \mathop{\mathbb{E}}_{X \sim \mu_1} \left[\operatorname*{essup}_{z \in B_{\|.\|}(X, \epsilon_2)} \mathbb{1}(g(z) \leq 0) - \lambda \|X - z\| \right] \\
&= \int_{\mathcal{X}} \operatorname*{essup}_{z \in B_{\|.\|}(x, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| \; d\mu_1(x).
\end{aligned}$$

Let us now consider $(H_j)_{j \in J}$ a partition of $\mathcal{X}$, we can write.

$$\begin{aligned}
&\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_1} \left[\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \|X - \phi_1(X)\| - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\} \right] \\
&= \sum_{j \in J} \int_{H_j} \operatorname*{essup}_{z \in B_{\|.\|}(x, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| \; d\mu_1(x)
\end{aligned}$$

In particular, we consider here $H_0 = P_h^{\complement}$, $H_1 = P_h \setminus P_h(\epsilon_2)$, and $H_2 = P_h(\epsilon_2)$.

For $x \in H_0 = P_h^{\complement}$. Taking $z = x$ we get $\mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| = 1$. Since for any $z \in \mathcal{X}$ we have $\mathbb{1}\{g(z) \leq 0\} - \lambda \|x - z\| \leq 1$, this strategy is optimal. Furthermore, for

any other optimal strategy z', we would have $\|x - z'\| = 0$, hence $z' = x$, and an optimal attack will never move the points of $H_0 = P_h^{\complement}$.

For $x \in H_1 = P_h \setminus P_h(\epsilon_2)$. We have $B_{\|.\|}(x, \epsilon_2) \subset P_h$ by definition of $P_h(\epsilon_2)$. Hence, for any $z \in B_{\|.\|}(x, \epsilon_2)$, one gets $g(z) > 0$. Then $\mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| \leq 0$. The only optimal z will thus be $z = x$, giving value 0.

Let us now consider $x \in H_2 = P_h(\epsilon_2)$ which is the interesting case where an attack is possible. We know that $B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement} \neq \emptyset$, and for any z in this intersection, $\mathbb{1}(g(z) \leq 0) = 1$. Hence :

$$\operatorname*{essup}_{z \in B_{\|.\|}(x,\epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda\|x - z\| = \max(1 - \lambda \operatorname*{essinf}_{z \in B_{\|.\|}(x,\epsilon_2) \cap P_h^{\complement}} \|x - z\|, 0) \quad \text{(B.9)}$$

$$= \max(1 - \lambda \pi_{B_{\|.\|}(x,\epsilon_2) \cap P_h^{\complement}}(x), 0) \quad \text{(B.10)}$$

Where $\pi_{B_{\|.\|}(x,\epsilon_2) \cap P_h^{\complement}}$ is the projection on the closure of $B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement}$. Note that $\pi_{B_{\|.\|}(x,\epsilon_2) \cap P_h^{\complement}}$ exists: g is continuous, so $B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement}$ is a closed set, bounded, and thus compact, since we are in finite dimension. The projection is however not guaranteed to be unique since we have no evidence on the convexity of the set. Finally, let us remark that, since $\lambda \in (0, 1)$, and $\epsilon_2 \leq 1$, one has $1 - \lambda \pi_{B_{\|.\|}(x,\epsilon_2) \cap P_h^{\complement}}(x) \geq 0$ for any $x \in H_2$. Hence, on $P_h(\epsilon_2)$, the optimal attack projects all the points on the decision boundary. For simplicity, and since there is no ambiguity, we write the projection π.

Finally. Since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma B.2 holds. Furthermore, the score for this optimal attack is:

$$\sup_{\phi \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(h, \phi)$$

$$= \sum_{y=\pm 1} \nu_y \sum_{j \in J} \int_{H_j} \operatorname*{essup}_{z \in B_{\|.\|}(x,\epsilon_2)} \mathbb{1}\{g(z)y \leq 0\} - \lambda\|x - z\| \; d\mu_y(x)$$

Since the value is 0 on $P_h \setminus P_h(\epsilon_2)$ (resp. on $N_h \setminus N_h(\epsilon_2)$) for ϕ_1 (resp. $\phi_{\text{-}1}$), one gets:

$$= \nu_1 \left[\int_{P_h(\epsilon_2)} \left(1 - \lambda\|x - \pi(x)\|\right) d\mu_1(x) + \int_{P_h^{\complement}} 1 d\mu_1(x) \right]$$

$$+ \nu_{\text{-}1} \left[\int_{N_h(\epsilon_2)} \left(1 - \lambda\|x - \pi(x)\|\right) d\mu_{\text{-}1}(x) + \int_{N_h^{\complement}} 1 d\mu_{\text{-}1}(x) \right]$$

$$
\begin{aligned}
=&\nu_1 \left[\int_{P_h(\epsilon_2)} \big(1 - \lambda \|x - \pi(x)\|\big) d\mu_1(x) + \mu_1(P_h^{\complement}) \right] \\
&+ \nu_{\text{-}1} \left[\int_{N_h(\epsilon_2)} \big(1 - \lambda \|x - \pi(x)\|\big) d\mu_{\text{-}1}(x) + \mu_{\text{-}1}(N_h^{\complement}) \right] \\
=&\, \mathcal{R}(h) + \nu_1 \int_{P_h(\epsilon_2)} \big(1 - \lambda \|x - \pi(x)\|\big) d\mu_1(x) + \nu_{\text{-}1} \int_{N_h(\epsilon_2)} \big(1 - \lambda \|x - \pi(x)\|\big) d\mu_{\text{-}1}(x)
\end{aligned}
$$

(16) holds since $\mathcal{R}(h) = \mathbb{P}(h(X) \neq Y)\mathbb{P}(g(X)Y \leq 0) = \nu_1 \mu_1(P_h^{\complement}) + \nu_{\text{-}1}\mu_{\text{-}1}(N_h^{\complement})$. This provides an interesting decomposition of the adversarial risk into the risk without attack and the loss on the attack zone.

□

Lemma B.1. *Let $h \in \mathcal{H}$ and $\boldsymbol{\phi} \in \mathfrak{BR}_{\Omega_{mass}}(h)$. Then the following assertion holds:*

$$
\begin{cases}
\phi_1(x) \in (P_h)^{\complement} & \text{if } x \in P_h(\epsilon_2) \\
\phi_1(x) = x & \text{otherwise.}
\end{cases}
$$

Where $(P_h)^{\complement}$, the complement of P_h in $\mathcal{X}$. $\phi_{\text{-}1}$ is characterized symmetrically.

Proof. Following the same proof schema as before the adversarial risk writes as follows:

$$
\begin{aligned}
&\sup_{\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2} \mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h, \phi) \\
=&\sup_{\boldsymbol{\phi} \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathop{\mathbb{E}}_{X \sim \mu_y} [\mathbb{1}\{h(\phi_y(X)) \neq y\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}] \\
=&\sup_{\boldsymbol{\phi} \in (\mathcal{F}_{\mathcal{X}})^2} \sum_{y=\pm 1} \nu_y \mathop{\mathbb{E}}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}] \\
=&\sum_{y=\pm 1} \nu_y \sup_{\phi_y \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_y} [\mathbb{1}\{g(\phi_y(X))y \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_y(X)\} - \infty \mathbb{1}\{\|X - \phi_y(X)\| > \epsilon_2\}]
\end{aligned}
$$

Finding ϕ_1 and ϕ_1 are two independent optimization problem, hence we focus on characterizing ϕ_1 (*i.e.* $y = 1$).

$$
\begin{aligned}
&\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_1} \left[\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_1(X)\} - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\}\right] \\
=&\mathop{\mathbb{E}}_{X \sim \mu_1} \left[\operatorname*{essup}_{z \in B_{\|.\|}(X, \epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{X \neq z\} \right]
\end{aligned}
$$

$$= \int_{\mathcal{X}} \operatorname*{essup}_{z \in B_{\|.\|}(x,\epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{x \neq z\} \; d\mu_1(x).$$

Let us now consider $(H_j)_{j \in J}$ a partition of $\mathcal{X}$, we can write.

$$\sup_{\phi_1 \in \mathcal{F}_{\mathcal{X}}} \mathop{\mathbb{E}}_{X \sim \mu_1} \left[\mathbb{1}\{g(\phi_1(X)) \leq 0\} - \lambda \mathbb{1}\{X \neq \phi_1(X)\} - \infty \mathbb{1}\{\|X - \phi_1(X)\| > \epsilon_2\} \right]$$
$$= \sum_{j \in J} \int_{H_j} \operatorname*{essup}_{z \in B_{\|.\|}(x,\epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{x \neq z\} \; d\mu_1(x)$$

In particular, we can take $H_0 = P_h^{\complement}$, $H_1 = P_h \setminus P_h(\epsilon_2)$, and $H_2 = P_h(\epsilon_2)$.

For $x \in H_0 = P_h^{\complement}$ or $x \in H_1 = P_h \setminus P_h(\epsilon_2)$. With the same reasoning as before, any optimal attack will choose $\phi_1(x) = x$.

Let $x \in H_2 = P_h(\epsilon_2)$. We know that $B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement} \neq \emptyset$, and for any z in this intersection, one has $g(z) \leq 0$ and $z \neq x$. Hence $\operatorname*{essup}_{z \in B_{\|.\|}(x,\epsilon_2)} \mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{z \neq x\} = \max(1 - \lambda, 0)$. Since $\lambda \in (0, 1)$ one has $\mathbb{1}\{g(z) \leq 0\} - \lambda \mathbb{1}\{z \neq x\} = 1 - \lambda$ for any $z \in B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement}$. Then any function that given a $x \in \mathcal{X}$ outputs $\phi_1(x) \in B_{\|.\|}(x, \epsilon_2) \cap P_h^{\complement}$ is optimal on H_2.

Finally. Since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma B.1 holds.

□

Lemma B.3. *Let us consider $\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. If we take $h \in \mathfrak{BR}(\boldsymbol{\phi})$, then for $y = 1$ (resp. $y =$ -1), and for any $B \subset P_h$ (resp. $B \subset N_h$) one has*

$$\mathbb{P}(Y = y | X \in B) \geq \mathbb{P}(Y = -y | X \in B)$$

with $Y \sim \nu$ and for all $y \in \mathcal{Y}$, $X|(Y = y) \sim \phi_y \# \mu_y$.

Proof. We reason ad absurdum. Let us consider $y = 1$, the proof for $y = -1$ is symmetrical. Let us suppose that there exists $C \subset P_h$ such that $\nu_{\text{-}1}\phi_{\text{-}1}\#\mu_{\text{-}1}(C) > \nu_1\phi_1\#\mu_1(C)$. We can then construct h_1 as follows:

$$h_1(x) = \begin{cases} h(x) & \text{if } x \notin C \\ -1 & \text{otherwise.} \end{cases}$$

Since h and h_1 are identical outside C, the difference between the adversarial risks of h and h_1 writes as follows:

$$\mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h, \phi) - \mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h_1, \phi)$$

$$
\begin{aligned}
&= \sum_{y=\pm 1} \nu_y \int_C \left(\mathbb{1}\{h(x) \neq y\} - \mathbb{1}\{h_1(x) \neq y\} \right) \, d(\phi_y \# \mu_y)(x) \\
&= \nu_{-1} \mathbb{1}\{h(x) = 1\} \phi_{-1} \# \mu_{\text{-}1}(C) - \nu_1 \mathbb{1}\{h_1(x) \neq 1\} \phi_1 \# \mu_1(C) \\
&= \nu_{-1} \phi_{-1} \# \mu_{\text{-}1}(C) - \nu_1 \phi_1 \# \mu_1(C)
\end{aligned}
$$

Since by hypothesis $\nu_{-1}\phi_{-1}\#\mu_{\text{-}1}(C) > \nu_1\phi_1\#\mu_1(C)$ the difference between the adversarial risks of h and h_1 is strictly positive. This means that h_1 gives strictly better adversarial risk than the best response h. Since, by definition h is supposed to be optimal, this leads to a contradiction. Hence Lemma B.3 holds. □

Additional Result. *Let us assume that there is a probability measure ζ that dominates both $\phi_1\#\mu_1$ and $\phi_{\text{-}1}\#\mu_{\text{-}1}$. Let us consider $\boldsymbol{\phi} \in \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$. If we take $h \in \mathfrak{BR}(\boldsymbol{\phi})$, then h is the Bayes Optimal Classifier for the distribution characterized by $(\nu, \phi_1\#\mu_1, \phi_{\text{-}1}\#\mu_{\text{-}1})$.*

Proof. For simplicity, we denote $f_1 = \frac{(d\phi_1\#\mu_1)}{d\zeta}$ and $f_{-1} = \frac{d(\phi_{-1}\#\mu_{\text{-}1})}{d\zeta}$ the Radon-Nikodym derivatives of $\phi_1\#\mu_1$ and $\phi_{-1}\#\mu_{\text{-}1}$ w.r.t. ζ. The best response h minimizes adversarial risk under attack $\boldsymbol{\phi}$. This minimal risk writes:

$$
\begin{aligned}
&\inf_{h\in\mathcal{H}} \mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h, \phi) \\
&= \inf_{h\in\mathcal{H}} \sum_{y=\pm 1} \nu_y \mathop{\mathbb{E}}_{x\sim\mu_y} [\mathbb{1}\{h(\phi_y(x)) \neq y\}] - \lambda\, \Omega(\boldsymbol{\phi}).
\end{aligned}
$$

Since the the penalty function does not depend on h, it suffices to seek $\inf_{h\in\mathcal{H}} \sum_{y=\pm 1} \nu_y \int_{\mathcal{X}} \mathbb{1}\{h(x) \neq y\} \, d(\phi_y\#\mu_y)(x)$. Moreover thanks to the transfer theorem, one gets the following:

$$
\begin{aligned}
&\inf_{h\in\mathcal{H}} \sum_{y=\pm 1} \nu_y \int_{\mathcal{X}} \mathbb{1}\{h(x) \neq y\} \; d(\phi_y\#\mu_y)(x) \\
&= \inf_{h\in\mathcal{H}} \sum_{y=\pm 1} \nu_y \int_{\mathcal{X}} \mathbb{1}\{h(x) \neq y\} f_y(x) \; d\zeta(x) \\
&= \inf_{h\in\mathcal{H}} \int_{\mathcal{X}} \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) \; d\zeta(x).
\end{aligned}
$$

Finally, since the integral is bounded we get:

$$
\inf_{h\in\mathcal{H}} \int_{\mathcal{X}} \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) \; d\zeta(x)
$$

$$= \int_{\mathcal{X}} \left[\inf_{h \in \mathcal{H}} \sum_{y=\pm 1} \nu_y \mathbb{1}\{h(x) \neq y\} f_y(x) \right] d\zeta(x).$$

Hence, the best response h is such that for every $x \in \mathcal{X}$, and $y \in \mathcal{Y}$, one has $h(x) = y$ if and only if $f_y(x) \leq f_{-y}(x)$. Thus, h is the optimal Bayes classifier for the distribution $(\nu, \phi_1 \# \mu_1, \phi_{\text{-}1} \# \mu_{\text{-}1})$. Furthermore, for $y = 1$ (resp. $y =$ -1), and for any $B \subset P_h$ (resp. $B \subset N_h$) one has:

$$\mathbb{P}(Y = y | X \in B) \geq \mathbb{P}(Y = -y | X \in B)$$

with $Y \sim \nu$ and for all $y \in \mathcal{Y}$, $X|(Y = y) \sim \phi_y \# \mu_y$.

□

Theorem B.3 (Non-existence of a pure Nash equilibrium)**.** *In our zero-sum game with $\lambda \in (0, 1)$ and penalty $\Omega \in \{\Omega_{mass}, \Omega_{norm}\}$, there is no Pure Nash Equilibrium.*

Proof. Let h be a classifier, $\phi \in \mathfrak{BR}_\Omega(h)$ an optimal attack against h. We will show that $h \notin \mathfrak{BR}(\boldsymbol{\phi})$, i.e. that h does not satisfy the condition from Lemma B.3. This suffices for Theorem B.3 to hold since it implies that there is no $(h, \boldsymbol{\phi}) \in \mathcal{H} \times \left(\mathcal{F}_{\mathcal{X}|\epsilon_2}\right)^2$ such that $h \in \mathfrak{BR}(\boldsymbol{\phi})$ *and* $\boldsymbol{\phi} \in \mathfrak{BR}_\Omega(h)$.

According to Lemmas B.1 and B.2, whatever penalty we use, there exists $\delta > 0$ such that $\phi_1 \# \mu_1(P_h(\delta)) = 0$ or $\phi_{-1} \# \mu_{\text{-}1}(N_h(\delta)) = 0$. Both cases are symmetrical, so let us assume that $P_h(\delta)$ is of null measure for the transported distribution conditioned by $y = 1$. Furthermore we have $\phi_{-1} \# \mu_{\text{-}1}(P_h(\delta)) = \mu_{\text{-}1}(P_h(\delta)) > 0$ since ϕ_{-1} is the identity function on $P_h(\delta)$, and since $\mu_{\text{-}1}$ is of full support on $\mathcal{X}$. Hence we get the following:

$$\phi_{-1} \# \mu_{\text{-}1}(P_h(\delta)) > \phi_1 \# \mu_1(P_h(\delta)).$$

Since the right side of the inequality is null, we also get:

$$\phi_{-1} \# \mu_{\text{-}1}(P_h(\delta)) \nu_{\text{-}1} > \phi_1 \# \mu_1(P_h(\delta)) \nu_1.$$

This inequality is incompatible with the characterization of best response for the Defender of Lemma B.3. Hence $h \notin \mathfrak{BR}(\boldsymbol{\phi})$.

□

Theorem B.4. *(Randomization matters) Let us consider $h_1 \in \mathcal{H}$, $\lambda \in (0, 1)$, $\Omega = \Omega_{mass}$, $\boldsymbol{\phi} \in \mathfrak{BR}_\Omega(h_1)$ and $h_2 \in \mathfrak{BR}(\boldsymbol{\phi})$. Then for any $\alpha \in (\max(\lambda, 1 - \lambda), 1)$ and for any $\boldsymbol{\phi}' \in \mathfrak{BR}_\Omega(m_{\mathbf{h}}^{\mathbf{q}})$ one has*

$$\mathcal{R}_{\text{adv}}^{\Omega_{mass}}(m_{\mathbf{h}}^{\mathbf{q}}, \boldsymbol{\phi}') < \mathcal{R}_{\text{adv}}^{\Omega_{mass}}(h_1, \boldsymbol{\phi}).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1 - \alpha)$, and $m_{\mathbf{h}}^{\mathbf{q}}$ is the mixture of $\mathbf{h}$ by $\mathbf{q}$.

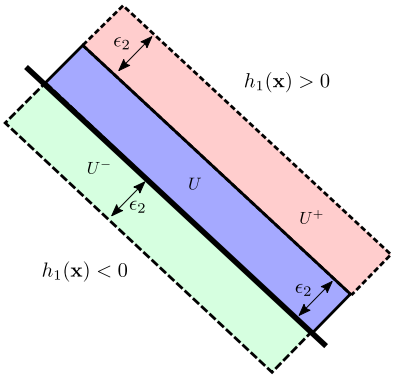


Figure B.3: Illustration of the notations U, U^+, and U^- for proof of Theorem B.4.

Proof. To demonstrate Theorem B.4, let us denote $U = P_{h_1}(\epsilon_2)$ and define the ϵ_2-*dilation of* U as $U \oplus \epsilon_2 := \left\{u + v \mid (u, v) \in U \times \mathcal{X} \text{ and } \|v\|_p \leq \epsilon_2\right\}$. We can construct h_2 as follows

$$h_2(x) = \begin{cases} -h_1(x) & \text{if } x \in U \\ h_1(x) & \text{otherwise.} \end{cases}$$

This means that h_2 changes the class of all points in U, and do not change the rest, compared to h_1. Then taking $\alpha \in (0, 1)$, we can define $m_{\boldsymbol{h}}^{\boldsymbol{q}}$, and $\boldsymbol{\phi}' \in \mathfrak{BR}_\Omega(m_{\boldsymbol{h}}^{\boldsymbol{q}})$. We aim to find a condition on α so that the score of $m_{\boldsymbol{h}}^{\boldsymbol{q}}$ is lower than the score of h_1. Finally, let us recall that

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(m_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\phi}') \\
&= \nu_1 \int_{\mathcal{X}} \operatorname*{essup}_{z \in B_{\|.\|}(x, \epsilon_2)} \alpha \mathbb{1}\{h_1(z) = \text{-}1\} + (1-\alpha)\mathbb{1}\{h_2(z) = \text{-}1\} - \lambda \mathbb{1}\{x \neq z\}\, d\mu_1(x) \\
&+ \nu_{\text{-}1} \int_{\mathcal{X}} \operatorname*{essup}_{z \in B_{\|.\|}(x, \epsilon_2)} \alpha \mathbb{1}\{h_1(z) = 10\} + (1-\alpha)\mathbb{1}\{h_2(z) = 1\} - \lambda \mathbb{1}\{x \neq z\}\, d\mu_{\text{-}1}(x).
\end{aligned}$$

The only terms that may vary between the score of h_1 and the score of $m_{\boldsymbol{h}}^{\boldsymbol{q}}$ are the integrals on $U, U \oplus \epsilon_2 \cap P_{h_1}$ and $\phi_{\text{-}1}^{-1}(U)$ – inverse image of U by $\phi_{\text{-}1}$. These sets represent respectively the points we mix on, the points that may become attacked – when changing from h_1 to $m_{\boldsymbol{h}}^{\boldsymbol{q}}$ – by moving them on U, and the ones that were – for h_1 – attacked before by moving them on U. Hence, for simplicity, we only write those terms. Furthermore, we denote

$$U^+ := U \oplus \epsilon_2 \cap P_{h_1} \setminus U,\ U^- := \phi_{\text{-}1}^{-1}(U) \text{ and recall } U := P_{h_1}(\epsilon_2).$$

One can refer to Figure B.3 for visual interpretation of this sets. We can now evaluate the worst case adversarial score for h_1 restricted to the above sets. Thanks to Lemma B.1 that characterizes $\boldsymbol{\phi}$, we can write

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h_1, \boldsymbol{\phi})_{|U,\, U^+,\, U^-} \\
&= (1-\lambda) \times \nu_1\mu_1(U) + \nu_{\text{-}1}\mu_{\text{-}1}(U) \\
&+ 0 \times \nu_1\mu_1\big(U^+\big) + \nu_{\text{-}1}\mu_{\text{-}1}\big(U^+\big) \\
&+ \nu_1\mu_1\big(U^-\big) + (1-\lambda) \times \nu_{\text{-}1}\mu_{\text{-}1}\big(U^-\big).
\end{aligned}$$

Similarly, we can write the worst case adversarial score of the mixture on the sets we consider. Note that the max operator comes from the fact that the adversary has to make a choice between attacking the zone or just take advantage of the error due to randomization.

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(m_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\phi}')_{|U,\, U^+,\, U^-} \\
&= \max(1-\alpha, 1-\lambda) \times \nu_1\mu_1(U) + \max(\alpha, 1-\lambda) \times \nu_{\text{-}1}\mu_{\text{-}1}(U) \\
&+ \max(0, 1-\alpha-\lambda) \times \nu_1\mu_1\big(U^+\big) + \nu_{\text{-}1}\mu_{\text{-}1}\big(U^+\big) \\
&+ \nu_1\mu_1\big(U^-\big) + \max(0, \alpha-\lambda) \times \nu_{\text{-}1}\mu_{\text{-}1}\big(U^-\big).
\end{aligned}$$

Computing the difference between these two terms, we get the following

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(h_1, \boldsymbol{\phi}) - \mathcal{R}_{\text{adv}}^{\Omega_{\text{mass}}}(m_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\phi}') && \text{(B.11)} \\
&= (1-\lambda-\max(1-\alpha, 1-\lambda)) \times \nu_1\mu_1(U) && \text{(B.12)} \\
&+ (1-\max(\alpha, 1-\lambda)) \times \nu_{\text{-}1}\mu_{\text{-}1}(U) && \text{(B.13)} \\
&- \max(0, 1-\alpha-\lambda) \times \nu_1\mu_1\big(U^+\big) && \text{(B.14)} \\
&+(1-\lambda-\max(0, \alpha-\lambda)) \times \nu_{\text{-}1}\mu_{\text{-}1}\big(U^-\big) && \text{(B.15)}
\end{aligned}$$

Let us now simplify Equation (B.11) using additional assumptions.

- First, we have that Equation (B.13) is equal to

  $$\min(1-\alpha, \lambda)\mu_{\text{-}1}(U)\nu_{\text{-}1} > 0.$$

  Thus, a sufficient condition for the difference between the adversarial scores to be positive is to have the other terms greater or equal to 0.

- To have Equation (B.12) ≥ 0 we can always set $\max(1-\alpha, 1-\lambda) = 1-\lambda$. This gives us $\alpha \geq \lambda$.

- Also note that to get (B.14) ≥ 0, we can force $\max(1-\alpha-\lambda, 0) = 0$. This gives us $\alpha \geq 1-\lambda$.

- Finally, since $\alpha \geq \lambda$, we have that $1 - \lambda - \max(0, \alpha - \lambda) = 1 - \alpha$ thus Equations (B.15) > 0.

With the above simplifications, we have (B.11) > 0 for any $\alpha > \max(\lambda, 1 - \lambda)$ which concludes the proof. □

Theorem B.5. *(Randomization matters) Let us consider $h_1 \in \mathcal{H}$, $\lambda \in (0, 1)$, $\Omega = \Omega_{norm}$, $\phi \in \mathfrak{BR}_\Omega(h_1)$ and $h_2 \in \mathfrak{BR}(\phi)$. Let us take $\delta \in (0, \epsilon_2)$, then for any $\alpha \in (\max(1 - \lambda\delta, \lambda(\epsilon_2 - \delta)), 1)$ and for any $\phi' \in \mathfrak{BR}_\Omega(m^{\mathbf{q}}_{\mathbf{h}})$ one has*

$$\mathcal{R}^{\Omega_{norm}}_{\text{adv}}(m^{\mathbf{q}}_{\mathbf{h}}, \phi') < \mathcal{R}^{\Omega_{norm}}_{\text{adv}}(h_1, \phi).$$

Where $\mathbf{h} = (h_1, h_2)$, $\mathbf{q} = (\alpha, 1 - \alpha)$, and $m^{\mathbf{q}}_{\mathbf{h}}$ is the mixture of $\mathbf{h}$ by $\mathbf{q}$.

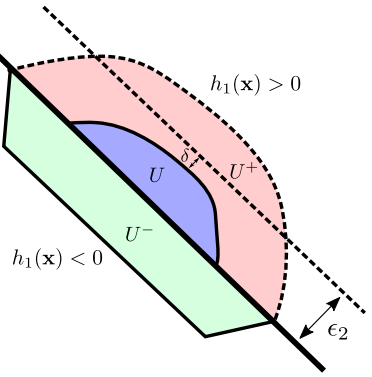


Figure B.4: Illustration of the notations U, U^+, U^- and δ for proof of Theorem B.5.

Proof. Let us take $U \subset P_{h_1}(\epsilon_2)$ such that

$$\min_{x \in U} \|x - \pi_{P_h \setminus P_h(\epsilon_2)}(x)\| = \delta \in (0, \epsilon_2)$$

. We construct h_2 as follows.

$$h_2(x) = \begin{cases} -h_1(x) & \text{if } x \in U \\ h_1(x) & \text{otherwise.} \end{cases}$$

This means that h_2 changes the class of all points in U, and do not change the rest. Let $\alpha \in (0, 1)$, the corresponding mixture $m^{\boldsymbol{q}}_{\boldsymbol{h}}$, and $\boldsymbol{\phi}' \in \mathfrak{BR}_\Omega(m^{\boldsymbol{q}}_{\boldsymbol{h}})$. We will find a condition on α so that the score of $m^{\boldsymbol{q}}_{\boldsymbol{h}}$ is lower than the score of h_1. Recall that

$$\begin{aligned} &\mathcal{R}^{\Omega_{\text{norm}}}_{\text{adv}}(m^{\boldsymbol{q}}_{\boldsymbol{h}}, \boldsymbol{\phi}') \\ &= \nu_1 \int_{\mathcal{X}} \operatorname*{essup}_{z \in B_{\|.\|}(x, \epsilon_2)} \alpha \mathbb{1}\{h_1(z) = \text{-}1\} + (1 - \alpha)\mathbb{1}\{h_2(z) = \text{-}1\} - \lambda\|x - z\| \, d\mu_1(x) \end{aligned}$$

$$+\nu_{-1}\int_{\mathcal{X}} \operatorname*{essup}_{z\in B_{\|\cdot\|}(x,\epsilon_2)} \alpha\mathbb{1}\{h_1(z)=1\}+(1-\alpha)\mathbb{1}\{h_2(z)=1\}-\lambda\|x-z\|\, d\mu_{-1}(x).$$

As we discussed in proof of Theorem B.4, the only terms that may vary between the score of h_1 and the score of $m_{\boldsymbol{h}}^{\boldsymbol{q}}$ are the integrals on U, $U\oplus\epsilon_2\cap P_{h_1}$ and $\phi_{-1}^{-1}(U)$. Hence, for simplicity, we only write those terms. Furthermore, we denote

$$U^+ := U\oplus\epsilon_2\cap P_{h_1}\setminus U,\ U^- := \phi_{-1}^{-1}(U) \text{ and } P_{\epsilon_2} := P_{h_1}(\epsilon_2).$$

One can refer to Figure B.4 for a visual interpretation of this ensembles. We can now evaluate the worst case adversarial score for h_1 restricted to the above sets. Thanks to Lemma B.2 that characterizes $\boldsymbol{\phi}$, we can write

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(h_1,\boldsymbol{\phi})\\
&=\nu_1\int_U\left(1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\right)d\mu_1(x)+\nu_{-1}\mu_{-1}(U)\\
&+\nu_1\int_{U^+\setminus P_{\epsilon_2}} 0\, d\mu_1(x)+\nu_{-1}\mu_{-1}\big(U^+\setminus P_{\epsilon_2}\big)\\
&+\nu_1\int_{U^+\cap P_{\epsilon_2}}\left(1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\right)d\mu_1(x)+\nu_{-1}\mu_{-1}\big(U^+\cap P_{\epsilon_2}\big)\\
&+\nu_1\mu_1\big(U^-\big)+\nu_{-1}\int_{U^-}\Big(1-\lambda\|x-\pi_U(x)\|\Big)d\mu_{-1}(x).
\end{aligned}$$

Similarly we can evaluate the worst case adversarial score for the mixture,

$$\begin{aligned}
&\mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(m_{\boldsymbol{h}}^{\boldsymbol{q}},\boldsymbol{\phi}')\\
&=\nu_1\int_U\max\left(1-\alpha,1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\right)\,d\mu_1(x)\\
&+\nu_{-1}\int_U\max(\alpha,1-\lambda\|x-\pi_{U^+}(x)\|)\,d\mu_{-1}(x)\\
&+\nu_1\int_{U^+\setminus P_{\epsilon_2}}\max(0,1-\alpha-\lambda\|x-\pi_U(x)\|)\,d\mu_1(x)+\nu_{-1}\mu_{-1}\big(U^+\setminus P_{\epsilon_2}\big)\\
&+\nu_1\int_{U^+\cap P_{\epsilon_2}}\max\left(1-\alpha-\lambda\|x-\pi_U(x)\|,1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\right)\,d\mu_1(x)\\
&+\nu_{-1}\mu_{-1}\big(U^+\cap P_{\epsilon_2}\big)+\nu_1\mu_1\big(U^-\big)
\end{aligned}$$

$$+\nu_{-1}\int_{U^-}\max\Big(0,1-\lambda\|x-\pi_{N_{h_1}^{\complement}\setminus U}(x)\|,\alpha-\lambda\|x-\pi_U(x)\|\Big)d\mu_{-1}(x).$$

Note that we need to take into account the special case of the points in the dilation that were already in the attacked zone before, and that can now be attacked in two ways, either by projecting on U – but that works with probability α, since the classification on U is now randomized – or by projecting on $P_{h_1}^{\complement}$, which works with probability 1 but may use more distance and so pay more penalty. We can now compute the difference between both scores.

$$\mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(h_1,\boldsymbol{\phi})-\mathcal{R}_{\text{adv}}^{\Omega_{\text{norm}}}(m_{\boldsymbol{h}}^{\boldsymbol{q}},\boldsymbol{\phi}') \tag{B.16}$$

$$=\nu_1\int_U 1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|-\max\Big(1-\alpha,1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\Big)d\mu_1(x) \tag{B.17}$$

$$+\nu_{-1}\int_U 1-\max(\alpha,1-\lambda\|x-\pi_{U^+}(x)\|)d\mu_{-1}(x) \tag{B.18}$$

$$-\nu_1\int_{U^+\setminus P_{\epsilon_2}}\max(1-\alpha-\lambda\|x-\pi_U(x)\|,0)d\mu_1(x) \tag{B.19}$$

$$+\nu_1\int_{U^+\cap P_{\epsilon_2}}1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|$$

$$-\max\Big(1-\alpha-\lambda\|x-\pi_U(x)\|,1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\Big)d\mu_1(x) \tag{B.20}$$

$$+\nu_{-1}\int_{U^-}1-\lambda\|x-\pi_U(x)\|$$

$$-\max\Big(0,1-\lambda\|x-\pi_{N_{h_1}^{\complement}\setminus U}(x)\|,\alpha-\lambda\|x-\pi_U(x)\|\Big)d\mu_{-1}(x). \tag{B.21}$$

Let us simplify Equation (B.16) using using additional hypothesis:

- First, note that Equation (B.18)> 0. Then a sufficient condition for the difference to be strictly positive is to ensure that other lines are ≥ 0.
- In particular to have (B.17) ≥ 0 it is sufficient to have for all $x \in U$

$$\max\Big(1-\alpha,1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|\Big)=1-\lambda\|x-\pi_{P_{h_1}^{\complement}}(x)\|.$$

This gives us $\alpha \geq \lambda(\epsilon_2-\delta) \geq \lambda\max_{x\in U}\|x-\pi_{P_{h_1}^{\complement}}(x)\|$.

- Similarly, to have (B.19) ≥ 0, we should set for all $x \in U^+ \setminus P_{\epsilon_2}$

$$\alpha \geq 1 - \lambda \|x - \pi_U(x)\|.$$

  Since $\min_{x \in U^+ \setminus P_{\epsilon_2}} \|x - \pi_U(x)\| = \delta$, we get the condition $\alpha \geq 1 - \lambda\delta$.

- Finally (B.21) ≥ 0, since by definition of U^-, for any $x \in U^-$ we have

$$\|x - \pi_{N_{h_1}^{\mathfrak{C}} \setminus U}(x)\| \geq \|x - \pi_U(x)\|.$$

Finally, by summing all these simplifications, we have (B.16) > 0. Hence the result hold for any $\alpha > \max(1 - \lambda\delta, \lambda(\epsilon_2 - \delta))$ □

B.9 Experimental results

In the experimental section, we consider $\mathcal{X} = [0,1]^{3\times32\times32}$ to be the set of images, and $\mathcal{Y} = \{1, ..., 10\}$ or $\mathcal{Y} = \{1, ..., 100\}$ according to the dataset at hand.

Adversarial attacks

Let $(x, y) \sim D$ and $h \in \mathcal{H}$. We consider the following attacks:

(i) ℓ_∞-PGD attack. In this scenario, the Adversary maximizes the loss objective function, under the constraint that the ℓ_∞ norm of the perturbation remains bounded by some value ϵ_∞. To do so, it recursively computes:

$$x^{t+1} = \Pi_{B_{\|.\|}(x,\epsilon_\infty)}\left[x^t + \beta\,\mathrm{sgn}\left(\nabla_x \mathcal{L}\left(h\left(x^t\right), y\right)\right)\right] \tag{B.22}$$

where $\mathcal{L}$ is some differentiable loss (such as the cross-entropy), β is a gradient step size, and Π_S is the projection operator on S. One can refer to [66] for implementation details.

(ii) ℓ_2-C&W attack. In this attack, the Adversary optimizes the following objective:

$$\arg\min_{\tau \in \mathcal{X}} \|\tau\|_2 + \lambda \times \mathrm{cost}(x + \tau) \tag{B.23}$$

where $\mathrm{cost}(x + \tau) < 0$ if and only if $h(x + \tau) \neq y$. The authors use a change of variable $\tau = \frac{1}{2}(\tanh(w) - x + 1)$ to ensure that $x + \tau \in \mathcal{X}$, a binary search to optimize the constant λ, and Adam or SGD to compute an approximated solution. One should refer to [27] for implementation details.

Experimental setup

Datasets. To illustrate our theoretical results we did experiments on the **CIFAR10** and **CIFAR100** datasets. See [56] for more details.

Classifiers. All the classifiers we use are WideResNets (see [108]) with 28 layers, a widen factor of 10, a dropout factor of 0.3 and LeakyRelu activations with a 0.1 slope.

Natural Training. To train an undefended classifier we use the following hyperparameters.

- **Number of Epochs:** 200
- **Batch size:** 128
- **Loss function:** Cross Entropy Loss
- **Optimizer :** SGD algorithm with momentum 0.9, weight decay of 2×10^{-4} and a learning rate that decreases during the training as follows:

$$lr = \begin{cases} 0.1 & \text{if} \quad 0 \leq \text{epoch} < 60 \\ 0.02 & \text{if} \quad 60 \leq \text{epoch} < 120 \\ 0.004 & \text{if} \quad 120 \leq \text{epoch} < 160 \\ 0.0008 & \text{if} \quad 160 \leq \text{epoch} < 200 \end{cases}$$

Adversarial Training. To adversarially train a classifier we use the same hyperparameters as above, and generate adversarial examples using the ℓ_∞**-PGD** attack with 20 iterations. When considering that the input space is $[0, 255]^{3\times32\times32}$, on **CIFAR10** and **CIFAR100**, a perturbation is considered to be imperceptible for $\epsilon_\infty = 8$. Here, we consider $\mathcal{X} = [0, 1]^{3\times32\times32}$ which is the normalization of the pixel space $[0.255]^{3\times32\times32}$. Hence, we choose $\epsilon_2 = 0.031$ ($\approx 8/255$) for each attack. Moreover, the step size we use for ℓ_∞**-PGD** is 0.008 ($\approx 2/255$), we use a random initialization for the gradient descent and we repeat the procedure three times to take the best perturbation over all the iterations *i.e* the one that maximises the loss. For the ℓ_∞**-PGD** attack against the mixture $m^{\mathbf{q}}_{\mathbf{h}}$, we use the same parameters as above, but compute the gradient over the loss of the expected logits (as explained in the main paper).

Evaluation Under Attack. At evaluation time, we use 100 iterations instead of 20 for **Adaptive-**ℓ_∞**-PGD**, and the same remaining hyperparameters as before. For the **Adaptive-**ℓ_2**-C&W** attack, we use 100 iterations, a learning rate equal to 0.01, 9 binary search steps, and an initial constant of 0.001. We give results for several different values of the rejection threshold: $\epsilon_2 \in \{0.4, 0.6, 0.8\}$.

Computing Adaptive-ℓ_2-C&W on a mixture To attack a randomized model, it is advised in the literature [94] to compute the expected logits returned by this model. However this advice holds for randomized models that return logits in the same range for a same example (*e.g.* classifier with noise injection). Our randomized model is a mixture and returns logits that depend on selected classifier. Hence, for a same example, the logits can be very different. This phenomenon made us notice that for some example in the dataset, computing the expected loss over the classifier (instead of the expected logits) performs better to find a good perturbation (it can be seen as computing the expectation of the logits normalized thanks to the loss). To ensure a fair evaluation of our model, in addition of using EOT with the expected logits, we compute in parallel EOT with the expected loss and take the perturbation that maximizes the expected error of the mixture. See the submitted code for more details.

Library used. We used the Pytorch and Advertorch libraries for all implementations.

Machine used. 6 Tesla V100-SXM2-32GB GPUs

Experimental details

Sanity checks for Adaptive attacks In [94], the authors give a lot of sanity checks and good practices to design an Adaptive attacks. We follow them and here are the information for **Adaptive-ℓ_∞-PGD** :

- We compute the gradient of the loss by doing the expected logits over the mixture.
- The attack is repeated 3 times with random start and we take the best perturbation over all the iterations.
- When adding a constant to the logits, it doesn't change anything to the attack
- When doing 200 iterations instead of 100 iterations, it doesn't change the performance of the attack
- When increasing the budget ϵ_∞, the accuracy goes to 0, which ensures that there is no gradient masking. Here are some values to back this statement:

Epsilon	0.015	0.031	0.125	0.250
Accuracy	0.638	0.546	0.027	0.000

Table B.5: Evolution of the accuracy under **Adaptive-ℓ_∞-PGD** attack depending on the budget ϵ_∞

- The loss doesn't fluctuate at the end of the optimization process.

Selecting the first element of the mixture. Our algorithm creates classifiers in a boosting fashion, starting with an adversarially trained classifier. There are several ways of selecting this first element of the mixture: use the classifier with the best accuracy under attack (option 1, called bestAUA), or rather the one with the best natural accuracy (option 2). Table B.6 compares both options.

Beside the fact that any of the two mixtures outperforms the first classifier, we see that the fisrt option always outperforms the second. In fact, when taking option 1 (bestAUA = True) the accuracy under ℓ_∞-**PGD** attack of the mixture is 3% better than with option 2 (bestAUA = False). One can also note that both mixtures have the same natural accuracy (0.80), which makes the choice of option 1 natural.

Extension to more than two classifiers

As we mention in the main part of the paper, a mixture of more than two classifiers can be constructed by adding at each step t a new classifier trained naturally on the dataset $\tilde{D}$ that contains adversarial examples against the mixture at step $t-1$. Since $\tilde{D}$ has to be constructed from a mixture, one would have to use an adaptive attack as **Adaptive-ℓ_∞-PGD**. Here is the algorithm for the extented version :

Training method	NA of the 1^{st} clf	AUA of the 1^{st} clf	NA of the mixture	AUA of the mixture
BAT (bestAUA=True)	0.77	**0.46**	**0.80**	**0.55**
BAT (bestAUA=False)	**0.83**	0.42	**0.80**	0.52

Table B.6: Comparison of the mixture that has as first classifier the best one in term of natural accuracy and the mixture that has as first classifier the best one in term of Accuracy under attack. The accuracy under attack is computed with the ℓ_∞-**PGD** attack. NA means matural accuracy, and AUA means accuracy under attack.

Algorithm 12: Boosted Adversarial Training

Input : n the number of classifiers, D the training data set and α the weight update parameter.

Create and adversarially train h_1 on D

$\mathbf{h} = (h_1)$; $\mathbf{q} = (1)$

for $i = 2, \ldots, n$ **do**

  Generate the adversarial data set $\tilde{D}$ against $m^{\mathbf{q}}_{\mathbf{h}}$.

  Create and naturally train h_i on $\tilde{D}$

  $q_k \leftarrow (1-\alpha)q_k \quad \forall k \in [i-1]$

  $q_i \leftarrow \alpha$

  $\mathbf{q} \leftarrow (\alpha, \ldots, q_i)$

  $\mathbf{h} \leftarrow (h_1, \ldots, h_i)$

end

return $m^{\mathbf{q}}_{\mathbf{h}}$

Here to find the parameter α, the grid search is more costly. In fact in the two-classifier version we only need to train the first and second classifier without taking care of α, and then test all the values of α using the same two classifier we trained. For the extended version, the third classifier (and all the other ones added after) depends on the first classifier, the second one and their weights $1-\alpha$ and α. Hence the third classifier for a certain value of α can't be use for another one and, to conduct the grid search, one have to retrain all the classifiers from the third one. Naturally the parameters α depends on the number of classifiers n in the mixtures.

C Résumé en français de la thèse

Contents

C.1 Contexte & motivations

Cette thèse vise à résoudre des problèmes multi-agents centralisés qui impliquent des interactions par paire entre agents. La configuration d'antennes dans un réseau cellulaire sans fil [89] est un exemple de ces problèmes : le choix d'un paramètre pour une antenne a un impact à la fois sur sa propre qualité de signal et sur celle de chacune de ses antennes voisines en raison de l'interférence du signal. De même, dans un parc éolien, le réglage d'une éolienne a un impact non seulement sur sa propre efficacité de collecte d'énergie mais aussi sur celle de ses voisines en raison des turbulences du vent [13, 36]. En considérant chaque antenne ou éolienne comme un agent, ces problèmes peuvent être modélisés comme un problème de bandit multi-agents (MA-MAB) [13] avec la connaissance d'un graphe de coordination [47] où chaque nœud représente un agent et chaque arête représente une interaction entre deux agents. Un problème de bandit à bras multiples (MAB) est

un problème de décision séquentiel où un apprenant doit choisir une action (aussi appelée bras) à chaque itération et obtient une récompense associée (éventuellement perturbée) qui informe sur la qualité de l'action choisie. Naturellement, l'apprenant ne connaît pas la distribution de la récompense pour chaque action possible. L'apprenant peut avoir des objectifs très différents, tels que maximiser les récompenses accumulées au cours du processus, ou bien en un nombre minimum d'essais et indépendamment des récompenses accumulées, déduire quelle est la meilleure action à choisir - c'est-à-dire la plus gratifiante. Par conséquent, un problème de bandit multi-agents à plusieurs bras est le cadre dans lequel plusieurs agents sont confrontés à un problème de bandit à plusieurs bras. Dans la littérature sur les bandits, on peut distinguer les bandits non structurés et les bandits structurés. Alors que le bandit non structuré considère que le fait de jouer une action et d'obtenir la récompense associée ne permet pas de déduire quoi que ce soit sur la distribution des récompenses des autres actions, le bandit structuré inclut les modèle de bandit où les récompenses des différentes actions partagent un paramètre commun [59]. Par exemple, une configuration de bandit structuré deja très etudiée dans la litterature est le bandit linéaire [11] où la récompense associée à toute action dépend linéairement d'un vecteur paramètre inconnu θ. Par conséquent, à un moment donné, le fait de choisir une action et de recevoir la récompense qui lui est associée donne des informations sur θ et, par définition, également sur les récompenses de toutes les autres actions. Nous nous intéressons ici à de tels environnements structurés et nous présentons à ce sujet un nouveau bandit structuré multi-agents appelé *Bandits Bilinéaires Graphiques*. La spécificité de cet environnement réside dans l'interdépendance des récompenses obtenues par les agents voisins dans le graphe et dans l'hypothèse que ces récompenses sont bilinéaires, ce qui nous apparaît comme l'extension naturelle des récompenses linéaires lorsque les agents sont dépendants par paire. En effet, si les problèmes MA-MAB ont été étudiés dans le cadre de bandits non structurés avec des agents indépendants et dépendants (voir *e.g.*, [3, 7, 13, 15, 18, 49, 58, 85, 87, 101]), seul le cadre des bandits structurés avec des agents indépendants a été exploré (voir *e.g.*, [6, 28, 30]). A travers cette thèse et les articles auxquels elle fait référence, nous voulons poser une première pierre à l'édifice.

C.2 Définition du problème

Bandits Bilinéaires Graphiques Stochastiques

Soit $\mathcal{G} = (V, E)$ le graphe dirigé défini par V l'ensemble fini de nœuds représentant les agents et E l'ensemble d'arêtes représentant les interactions entre les agents. Nous supposons que si $(i, j) \in E$ alors $(j, i) \in E$. Le graphe pourrait être considéré comme non dirigé mais nous supposons que les interactions entre deux voisins ne sont pas nécessairement symétriques par rapport aux récompenses obtenues, nous choisissons donc de conserver le graphe dirigé pour mettre en évidence cette asymétrie potentielle. Pour tout agent $i \in V$, nous désignons $\mathcal{N}_i$ l'ensemble de ses agents voisins. Soit $n = |V|$ le nombre de nœuds, $m = |E|$ le nombre d'arêtes et $\mathcal{X} \subset \mathbb{R}^d$ un ensemble de bras fini où $K = |\mathcal{X}|$ désigne le nombre de bras. Le bandit bilinéaire graphique avec un graphe $\mathcal{G}$ et un ensemble de bras $\mathcal{X}$ consiste en le problème de décision séquentiel suivant :

Bandits Bilinéaires Graphiques Stochastiques

Pour chaque tour $t > 0$,

1. Chaque agent $i \in V$ choisit un bras $x_t^{(i)}$ dans $\mathcal{X}$.
2. Ensuite, chaque agent $i \in V$ reçoit une récompense bilinéaire bruitée pour chacun de ses voisins $j \in \mathcal{N}_i$:

$$y_t^{(i,j)} = x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} + \eta_t^{(i,j)} \quad , \tag{C.1}$$

où $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ est une matrice inconnue, et $\eta_t^{(i,j)}$ une variable aléatoire σ-sous-gaussienne de moyenne nulle.

La récompense $y_t^{(i,j)}$ reflète la qualité de l'interaction entre les nœuds voisins i et j lorsqu'ils tirent respectivement les bras $x_t^{(i)}$ et $x_t^{(j)}$ à l'itération t. Le cadre bilinéaire apparaît comme une extension naturelle du cadre linéaire pour modéliser l'interaction entre deux agents.

Notons que ce model peut être considéré soit dans un cadre décentralisé où les agents prennent des actions sans consulter les autres agents, soit dans un cadre centralisé où une entité centrale choisit les bras de tous les agents, agrège les récompenses obtenues et conçoit une stratégie globale pour les agents du graphe.

Dans cette thèse, nous ne considérons que le cas centralisé où une entité centrale gère tous les agents, choisit à chaque instant t l'allocation $(x_t^{(1)}, \dots, x_t^{(n)})$ et reçoit ensuite les récompenses associées $y_t^{(i,j)}$ pour tous les $(i,j) \in E$.

Objectifs

Comme nous l'avons brièvement mentionné précédemment, il existe deux principaux objectifs différents qu'un apprenant (ici l'entité centrale) peut vouloir atteindre dans un problème de bandit.

Identifier la meilleure allocation. Le premier objectif que nous voulons traiter dans cette thèse est celui où l'apprenant est intéressé à trouver, en un minimum de tours, la meilleure allocations de bras $(x_\star^{(1)}, \dots, x_\star^{(n)})$ qui maximise la récompense globale moyenne obtenue sur le graphe :

$$(x_\star^{(1)}, \dots, x_\star^{(n)}) = \underset{(x^{(1)},\dots,x^{(n)})\in\mathcal{X}^n}{\arg\max} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)} \quad .$$

Cet objectif implique que l'entité centrale ne se soucie pas de choisir une allocation sous-optimale $(x_t^{(1)}, \dots, x_t^{(n)})$ à chaque instant t tant qu'elle donne suffisamment d'informations sur le paramètre

inconnu $\mathbf{M}_\star$ afin de construire une estimation précise $\hat{\mathbf{M}}$. Cet objectif est connu sous le nom d'*exploration pure* ou d'*identification du meilleur bras* [10, 24].

Maximiser les récompenses cumulées. Le deuxième objectif que nous voulons traité est le plus souvent considéré dans la littérature sur les bandits où l'apprenant souhaite maximiser la somme des récompenses (en esperance) obtenues au cours des tours. Dans notre cas, l'entité centrale souhaite maximiser les récompenses globales cumulées, données par la formule suivante

$$\sum_{t=1}^{T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \quad .$$

Alors que le premier objectif permet à l'apprenant d'être dans un cadre d'exploration pure, indépendamment des récompenses obtenues tout au long du processus, l'objectif de maximisation des récompenses cumulées nécessite un compromis entre l'exploration des différents bras possibles pour avoir une estimation précise $\hat{\mathbf{M}}$ de $\mathbf{M}_\star$ et l'exploitation des bras qui semblent être les plus optimaux étant donné $\hat{\mathbf{M}}$ afin d'obtenir les récompenses cumulées maximales.

Pour ces deux objectifs (identification du meilleur bras ou maximisation des récompenses cumulées) et étant donné une estimation $\hat{\mathbf{M}}$, l'apprenant devra résoudre à un moment donné le problème d'optimisation suivant

$$\max_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top} \hat{\mathbf{M}} x^{(j)} \quad . \tag{C.2}$$

En effet, pour l'identification du meilleur bras, ce problème d'optimisation doit être résolu à la fin lorsque l'apprenant veut retourner la meillieure allocation étant donné l'estimation $\hat{\mathbf{M}}$ construite pendant la procédure d'apprentissage. Pour la maximisation des récompenses cumulées, ce problème d'optimisation peut devoir être résolu pendant la procédure d'apprentissage lorsque l'apprenant veut exploiter et renvoyer le meilleur bras articulé estimé compte tenu de sa connaissance actuelle de l'environnement qui est l'estimation construite $\hat{\mathbf{M}}$.

La résolution de ce problème d'optimisation n'est pas triviale, aussi pour les deux objectifs nous considérons l'objectif sous-jacent commun de résolution de ce problème.

C.3 Trouver la meilleure allocation lorsque la matrice est connu

C.3.1 Un problème NP-Dur

Nous abordons le problème de la recherche de la meilleure alloaction étant donné $\mathbf{M}_\star$ et nous le désignons comme suit :

$$\left(x_\star^{(1)},\dots,x_\star^{(n)}\right) = \operatorname*{arg\,max}_{\left(x^{(1)},\dots,x^{(n)}\right)\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)} \quad . \tag{C.3}$$

Remarquez que si le couple $(x_\star, x'_\star) = \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \big(\mathbf{M}_\star + \mathbf{M}_\star^\top\big)x'$ est tel que $x_\star = x'_\star$ alors trouver la meilleure allocation est trivial et la solution est d'attribuer $x_\star$ à tous les nœuds. À l'inverse, si $x_\star \neq x'_\star$, le problème peut être plus difficile : selon le graphe $\mathcal{G}$, l'allocation optimal pourrait soit être composée exclusivement du couple $(x_\star, x'_\star)$, soit être composée d'autres bras dans $\mathcal{X}$. On pourrait vouloir utiliser la programmation dynamique comme dans [7] pour résoudre ce problème d'optimisation, cependant dans ce cadre particulier, cela conduirait à utiliser un algorithme en temps non-polynomial. En effet, le théorème suivant indique que, même en connaissant le vrai paramètre $\mathbf{M}_\star$, l'identification de la meilleure allocation $(x_\star^{(1)}, \dots, x_\star^{(n)})$ est NP-Dur par rapport au nombre de nœuds n.

Theorem C.1. *Considérons une matrice donnée $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ et un ensemble de bras finis $\mathcal{X} \subset \mathbb{R}^d$. A moins que P=NP, il n'existe pas d'algorithme en temps polynomial pour trouver la solution optimale de*

$$\max_{(x^{(1)},\dots,x^{(n)})\in\mathcal{X}^n} \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star\, x^{(j)} \quad .$$

Par conséquent, étant donné la vraie matrice $\mathbf{M}_\star$, l'apprenant n'est pas assuré de trouver en temps polynomial l'alloaction $\left(x_\star^{(1)}, \dots, x_\star^{(n)}\right)$ maximisant la récompense globale attendue. Dans les sections suivantes, nous donnons des algorithmes d'approximation en temps polynomial qui ont des garanties sur la récompense globale attendue retournée par rapport à la récompense optimale.

C.3.2 Algorithmes d'approximation et guaranties théoriques

Étant donné la matrice $\mathbf{M}_\star$, l'objectif est de concevoir un algorithme qui renvoie une allocation $\left(x^{(1)}, \dots, x^{(n)}\right)$ telle que sa récompense globale associée $y = \sum_{(i,j)\in E} x^{(i)\top}\mathbf{M}_\star x^{(j)}$ ait la garantie d'être proche de la récompense globale optimale $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top}\mathbf{M}_\star x_\star^{(j)}$. En d'autres termes, nous voulons trouver un paramètre d'approximation $0 < \alpha \leq 1$ tel que, $y \geq \alpha y_\star$. Bien que le problème d'optimisation que nous cherchons à résoudre se trouve dans la littérature sur les *Champs aléatoires de Markov* lorsqu'on traite un graphe multi-labeélisé (voir *e.g.*, [5]), à notre connaissance, les algorithmes qui donnent un rapport d'approximation sur la solution optimale n'ont pas été explorés.

L'approche que nous présentons dans cette section consiste d'abord à considérer le problème localement, *i.e.*, au niveau des arêtes. En effet, considérons deux nœuds voisins i et j dans V et seulement les récompenses liées à ces nœuds, qui sont $x^{(i)\top}\mathbf{M}_\star x^{(j)}$ et $x^{(j)\top}\mathbf{M}_\star x^{(i)}$. En additionnant ces deux quantités,[1] on obtient $x^{(i)\top}\mathbf{M}_\star x^{(j)} + x^{(j)\top}\mathbf{M}_\star x^{(i)} = x^{(i)\top}\big(\mathbf{M}_\star + \mathbf{M}_\star^\top\big)x^{(j)}$ qui représente la récompense entre les deux nœuds voisins (i) et (j). Une stratégie locale que l'entité centrale devrait mettre en œuvre consiste donc à allouer $(x^{(i)}, x^{(j)}) = \arg\max_{(x,x')\in\mathcal{X}^2} x^\top\big(\mathbf{M}_\star + \mathbf{M}_\star^\top\big)x' = (x_\star, x'_\star)$. Naturellement, si cette stratégie locale est facile à appliquer pour un couple de voisins (i, j), elle ne peut pas etre simultanément appliquée à tous les autres couples du graphe puisque certains d'entre eux partagent les mêmes nœuds. Cepen-

[1] Ces quantités ne sont pas égales puisque la matrice $\mathbf{M}_\star$ n'est pas nécessairement symétrique.

dant, on peut tirer un enseignement de cette stratégie, à savoir qu'étant donné le bras conjoint optimal $(x_\star^{(1)}, \dots, x_\star^{(n)})$, on a pour tout $(i,j) \in E$,

$$x_\star^{(i)} \left(\mathbf{M}_\star + \mathbf{M}_\star^\top \right) x_\star^{(j)} \leq x_\star^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top \right) x'_\star \ . \tag{C.4}$$

Ainsi, au lieu de chercher l'allocation optimal (ce qui est NP-Dur), on peut alternativement rechercher l'allocation qui, pour toute arête $(i,j) \in E$, construit autant de paires $(x^{(i)}, x^{(j)}) = (x_\star, x'_\star)$ que possible. Affecter $x_\star$ à un sous-ensemble de nœuds et $x'_\star$ au complémentaire revient à couper le graphe en deux morceaux et à créer deux ensembles distincts de nœuds V_1 et V_2 tels que $V = V_1 \cup V_2$ et $V_1 \cap V_2 = \emptyset$. Ainsi, la stratégie décrite se résume à trouver une coupe passant par le nombre maximal d'arêtes.

Ce problème est connu sous le nom de Max-Cut (voir *e.g.*, [44, 84]), qui est également NP-Dur. Cependant, l'attention considérable portée à ce problème nous permet d'utiliser l'un des nombreux algorithmes d'approximation (voir, *e.g.*, Algorithm 13) qui garantissent de produire une coupe passant par au moins une fraction donnée des arêtes du graphe. La plupart des garanties pour l'approximation du problème de Max-Cut sont données par rapport à la solution optimale de Max-Cut, ce qui n'est pas exactement la garantie que nous recherchons : nous avons besoin d'une garantie en proportion du nombre total d'arêtes. Nous devons donc faire attention à l'algorithme que nous choisissons.

Algorithm 13: Approx-MAX-CUT [84]

Entrée: $\mathcal{G} = (V, E)$
Initialiser $V_1 = \emptyset, V_2 = \emptyset$
for *i in* V **do**
  $n_1 = |\{(i,j) \in E \mid j \in V_1\}|$;
  $n_2 = |\{(i,j) \in E \mid j \in V_2\}|$;
  si $n_1 > n_2$ **alors** $V_2 \leftarrow V_2 \cup \{i\}$ **sinon** $V_1 \leftarrow V_1 \cup \{i\}$;
end
retourner (V_1, V_2)

A partir de l'Algorithme 13, on peut avoir une garantie sur la proportion d'arêtes coupées par rapport au nombre total d'arêtes m. Nous énonçons cette garantie dans la proposition suivante.

Proposition C.1. *Étant donné un graphe* $\mathcal{G} = (V, E)$*, l'Algorithme 13 retourne un couple* (V_1, V_2) *tel que*

$$|\{(i,j) \in E \,|\, (i \in V_1 \wedge j \in V_2) \vee (i \in V_2 \wedge j \in V_1)\}| \geq \frac{m}{2} \ .$$

Étant donné cette garantie par rapport au nombre total d'arêtes, il ne reste plus qu'à présenter la stratégie complète qui consiste à allouer aux nœuds de V_1 le bras $x_\star$ et aux nœuds de V_2 le bras $x'_\star$. Nous donnons la stratégie dans l'Algorithme 14.

Avec cet algorithme, étant donné l'allocation retournée $(x^{(1)}, \dots, x^{(n)})$, pour certaines arêtes $(i,j) \in E$, les bras alloués associés seront les couples optimaux $(x_\star, x'_\star)$ ou $(x'_\star, x_\star)$ et pour

Algorithm 14: Algorithm d'approximation pour notre problème NP-Dur

Entrée: Graphe $\mathcal{G} = (V, E)$, ensemble de bras $\mathcal{X}$, matrice $\mathbf{M}_\star$
$(V_1, V_2) = \text{Approx-MAX-CUT}(\mathcal{G})$;
Trouver $(x_\star, x'_\star) \in \arg\max_{(x,x')\in\mathcal{X}^2} x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x'$;
for *i in* V_1 **do**
  $x_t^{(i)} = x_\star$; // Peut etre fait en parallèle
end
for *i in* V_2 **do**
  $x_t^{(i)} = x'_\star$; // Peut etre fait en parallèle
end
retourner $(x^{(1)}, \dots, x^{(n)})$

d'autres arêtes $(i, j) \in E$ les bras alloués associés seront les couples sous-optimaux et non désirés $(x_\star, x_\star)$ ou $(x'_\star, x'_\star)$.

Avant d'énoncer la garantie de cet algorithme par rapport à la récompense globale optimale, introduisons m_1 (respectivement m_2) le nombre d'arêtes qui vont des nœuds de V_1 (respectivement V_2) aux nœuds de V_1 (respectivement V_2) et $m_{1\to2}$ (respectivement $m_{2\to1}$) le nombre d'arêtes qui vont des nœuds de V_1 (respectivement V_2) aux nœuds de V_2 (respectivement V_1). Remarquez que le nombre total d'arêtes $m = m_{1\to2} + m_{2\to1} + m_1 + m_2$ et que par définition de l'ensemble d'arêtes E et en utilisant la Proposition C.1 nous avons $m_{1\to2} = m_{2\to1} \geq m/4$ et $m_1 + m_2 \leq m/2$.

Theorem C.2. *Considérons le graphe $\mathcal{G} = (V, E)$, un ensemble fini de bras $\mathcal{X} \subset \mathbb{R}^d$ et la matrice $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ donnée en entrée de l'Algorithme 14. Soit $(x_\star^{(1)}, \dots, x_\star^{(n)})$ l'allocation optimale telle que défini dans (C.3) et soit $0 \leq \xi \leq 1$ un paramètre dépendant du problème défini par*

$$\xi = \min_{x\in\mathcal{X}} \frac{x^\top \mathbf{M}_\star x}{\frac{1}{m}\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}} \ ,$$

et on fixe $\alpha = \frac{1+\xi}{2}$. Alors, la récompense globale $y = \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ associée à l'allocation $\left(x^{(1)}, \dots, x^{(n)}\right) \in \mathcal{X}^n$ retournée par l'Algorithme 14 vérifie :

$$y \geq \alpha y_\star \ .$$

où $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$.

Enfin, la complexité de l'algorithme est de $\mathcal{O}(K^2 + n^2)$.

C.3.3 Algorithme amélioré utilisant la structure du graphe

Dans cette section, nous voulons capitaliser sur l'Algorithme 14 et sa solution $\frac{1+\xi}{2}$-optimale pour affiner l'allocation des bras $x_\star$ et $x'_\star$ de telle sorte que les récompenses sous-optimales obtenues $x_\star^\top \mathbf{M}_\star x_\star$ et $x_\star'^\top \mathbf{M}_\star x'_\star$ pénalisent le moins possible la récompense globale.

En effet, dans l'Algorithme 14, le choix du couple $(x_\star, x'_\star)$ est uniquement guidé par le gain potentiel que l'on pourrait obtenir au niveau des arêtes coupées (*i.e.*, qui vont d'un nœud dans V_1 à un nœud dans V_2 ou vice versa). Elle ne prend pas en compte toutes les m_1 récompenses de la forme $x_\star^\top \mathbf{M}_\star x_\star$ et les m_2 récompenses de la forme $x_\star'^\top \mathbf{M}_\star x'_\star$ que l'on obtient en attribuant $x_\star$ aux nœuds de V_1 et $x'_\star$ aux nœuds de V_2.

Ici, l'amélioration que nous pouvons apporter est de les inclure dans le problème d'optimisation et de pondérer les différentes récompenses obtenues par le graphe en utilisant les proportions $m_{1\to2}$, $m_{2\to1}$, m_1 et m_2. En désignant par $(\tilde{x}_\star, \tilde{x}'_\star)$ la solution du problème d'optimisation suivant,

$$\max_{(x,x')\in\mathcal{X}^2} m_{1\to2}\cdot x^\top \mathbf{M}_\star x' + m_{2\to1}\cdot x'^\top \mathbf{M}_s tar x + m_1 \cdot x^\top \mathbf{M}_s tar x + m_2 \cdot x'^\top \mathbf{M}_\star x' \ , \tag{C.5}$$

nous optimisons la récompense globale totale que l'on obtiendrait en allouant seulement deux bras $(x, x') \in \mathcal{X}^2$ dans le graphe. Cette stratégie est décrite dans l'Algorithme 15.

Algorithm 15: Algorithm d'approximation amélioré pour notre problème NP-Dur

Entrée : Graphe $\mathcal{G} = (V, E)$, ensemble de bras $\mathcal{X}$, matrice $\mathbf{M}_\star$
$(V_1, V_2) =$ Approx-MAX-CUT$(\mathcal{G})$;
$m_{1\to2} = |\{(i,j) \in E | i \in V_1 \wedge j \in V_2\}|$;
$m_{2\to1} = |\{(i,j) \in E | i \in V_2 \wedge j \in V_1\}|$;
$m_1 = |\{(i,j) \in E | i \in V_1 \wedge j \in V_1\}|$;
$m_2 = |\{(i,j) \in E | i \in V_2 \wedge j \in V_2\}|$;
Trouver $(\tilde{x}_\star, \tilde{x}'_\star)$ solution of (C.5);
for *i in* V_1 **do**
  $x_t^{(i)} = \tilde{x}_\star$; // Peut etre fait en parallèle
end
for *i in* V_2 **do**
  $x_t^{(i)} = \tilde{x}'_\star$; // Peut etre fait en parallèle
end
retourner $(x^{(1)}, \ldots, x^{(n)})$

Pour comprendre et analyser ce nouvel algorithme, définissons $\Delta \geq 0$ la différence de récompense globale de l'attribution de $(\tilde{x}_\star, \tilde{x}'_\star)$ au lieu de $(x_\star, x'_\star)$ comme suit :

$$\Delta = m_{1\to2}\left(\tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}'_\star - x_\star^\top \mathbf{M}_\star x'_\star\right) + m_{2\to1}\left(\tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}_\star - x_\star'^\top \mathbf{M}_\star x_\star\right)$$

$$+ m_1\left(\tilde{x}_\star^\top \mathbf{M}_\star \tilde{x}_\star - x_\star^\top \mathbf{M}_\star x_\star\right) + m_2\left(\tilde{x}_\star'^\top \mathbf{M}_\star \tilde{x}_\star' - x_\star'^\top \mathbf{M}_\star x_\star'\right) \ .$$

Les nouvelles garanties que nous obtenons sur la récompense de l'allocation obtenue par l'Algorithme 15 sont énoncées dans le théorème suivant.

Theorem C.3. *Considérons le graphe $\mathcal{G} = (V, E)$, un ensemble fini de bras $\mathcal{X} \subset \mathbb{R}^d$ et la matrice $\mathbf{M}_\star \in \mathbb{R}^{d\times d}$ donnés en entrée de l'Algorithme 15. Soit $(x_\star^{(1)}, \dots, x_\star^{(n)})$ l'allocation optimale telle que définie dans (C.3) et que $0 \leq \xi \leq 1$ soit défini comme dans le Théorème C.2. Soit $0 \leq \epsilon \leq \frac{1}{2}$ un paramètre dépendant du problème qui mesure le gain relatif de l'optimisation sur les récompenses sous-optimales définies comme :*

$$\epsilon = \frac{\Delta}{\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}} \ ,$$

et on fixe $\alpha = \frac{1+\xi}{2} + \epsilon$. Alors, la récompense globale $y = \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$ associée à l'allocation $\left(x^{(1)}, \dots, x^{(n)}\right) \in \mathcal{X}^n$ retournée par l'Algorithme 15 vérifie :

$$y \geq \alpha y_\star \ .$$

où $y_\star = \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)}$.

Corollary C.1. *Considérons le même cadre que dans le théorème C.3, le rapport d'approximation peut être défini avec les paramètres $m_{1\to 2}$, $m_{2\to 1}$, m_1 et m_2 qui dépendent du graphe et de l'algorithme d'approximation du problème de coupe maximale tel que*

$$\alpha = \frac{m_{1\to 2} + m_{2\to 1}}{m} + \frac{m_1 + m_2}{m}\xi + \epsilon \ .$$

Ce corollaire est utile pour comprendre en pratique le type de garanties que nous pouvons avoir en fonction de la structure du graphe et de l'algorithme d'approximation que nous utilisons pour résoudre le problème de Max-Cut.

C.4 Identification du meilleur bras pour les bandits bilineaires graphiques

Dans cette section, nous supposons que nous ne connaissons pas la matrice $\mathbf{M}_\star$ et qu'une entité centrale fait face à un problème de bandits bilinéaires graphiques où à chaque tour elle choisit un bras pour chaque nœud du graphe et observe une récompense bilinéaire pour chaque arête du graphe. Dans ce chapitre, nous nous concentrerons sur l'objectif d'identification du meilleur bras.

C.4.1 Préliminaires

Pour simplifier, nous considérons que la matrice inconnue $\mathbf{M}_\star$ est symétrique, ce qui simplifie grandement le raisonnement. Dans le chapitre C.3, nous avons conçu des algorithmes en temps polynomial qui nous permettent de calculer une solution d'approximation α au problème NP-Dur consistant à trouver la meilleure allocation étant donné $\mathbf{M}_\star$. Remarquez que dans l'algorithme d'α-approximation 14, $\mathbf{M}_\star$ n'est utilisé que pour identifier la meilleure paire $(x_\star, x'_\star)$ comme suit :

$$\begin{aligned}\left(x_\star, x'_\star\right) &= \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top\right) x' \\ &= \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \mathbf{M}_\star x' \ . \end{aligned} \tag{C.6}$$

Ainsi, utiliser une estimation $\hat{\mathbf{M}}$ de $\mathbf{M}_\star$ ayant la propriété suivante :

$$\underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \hat{\mathbf{M}} x' = \underset{(x,x')\in\mathcal{X}^2}{\arg\max}\, x^\top \mathbf{M}_\star x' = \left(x_\star, x'_\star\right) \ , \tag{C.7}$$

nous permet d'identifier la paire $(x_\star, x'_\star)$, et nous donne donc les mêmes garanties que celles présentées dans le Théorème C.2. Nous abordons donc le problème de construire $\hat{\mathbf{M}}$ tel que, en un nombre minimum de tours, avec grande probabilité, nous soyons capables d'identifier la paire $(x_\star, x'_\star)$ et d'appliquer l'Algorithme 14.

Condition d'arrêt

Remarquons que le problème d'optimisation (C.6) est équivalent au problème d'optimisation suivant

$$\left(x_\star, x'_\star\right) = \underset{(x,x')\in\mathcal{X}}{\arg\max} \left\langle \operatorname{vec}\left(xx'^\top\right), \operatorname{vec}\left(\mathbf{M}_\star\right)\right\rangle \ .$$

Simplifions les notations et désignons par $\theta_\star \triangleq \operatorname{vec}\left(\mathbf{M}_\star\right)$ la version vectorisée de la matrice inconnue $\mathbf{M}_\star$. Utilisons également la notation $z_{xx'} \triangleq \operatorname{vec}\left(xx'^\top\right)$, et définissons $\mathcal{Z} = \{z_{xx'} | (x, x') \in \mathcal{X}^2\}$ l'ensemble contenant de tels vecteurs. Alors, chercher le couple $(x_\star, x'_\star)$ revient à chercher le vecteur $z_\star \in \mathcal{Z}$ où

$$z_\star = \underset{z\in\mathcal{Z}}{\arg\max} \langle z, \theta_\star \rangle \ .$$

En d'autres termes, nous voulons trouver un bras $z \in \mathcal{Z}$, tel que pour tout $z' \in \mathcal{Z}$, $(z - z')^\top \theta_\star \geq 0$. Cependant, nous n'avons pas accès à $\theta_\star$, donc nous devons utiliser son estimation empirique.

Ainsi, à chaque tour t, l'entité centrale peut choisir pour chaque couple de voisins (i, j) un bras $z \in \mathcal{Z}$ et obtenir une récompense linéaire bruitée de la forme $\langle z, \theta_\star \rangle + \eta$ où η est une variable aléatoire σ-sous-gaussienne, qui peut être utilisée pour calculer une estimation $\hat{\theta}_t$.

Pour plus de clarté, nous appellerons tout $x \in \mathcal{X}$ un *bras de nœud* et tout $z \in \mathcal{Z}$ un *bras d'arête*. Si $x_t^{(i)} \in \mathcal{X}$ représente le bras de nœud alloué au nœud $i \in V$ au temps t, pour chaque arête $(i, j) \in E$ nous désignerons le bras d'arête associé par $z_t^{(i,j)} \triangleq \operatorname{vec}\left(x_t^{(i)} x_t^{(j)\top}\right) \in \mathcal{Z}$.

Le but ici est de définir la séquence optimale $(z_1, \dots, z_{mt}) \in \mathcal{Z}^{mt}$ qui devrait être tirée dans les t premiers tours de façon à ce que (C.7) soit atteint le plus tôt possible. Une approche naturelle consiste à s'appuyer sur les stratégies classiques développées pour l'identification du meilleur bras dans les bandits linéaires. Nous définissons $(y_1, \dots, y_{mt})$ les récompenses bruitées correspondantes de la séquence $(z_1, \dots, z_{mt})$. Nous supposons que les termes de bruit dans les récompenses sont *i.i.d.*, suivant une distribution σ-sous-gaussienne. Soit $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \in \mathbb{R}^{d^2}$ la solution du problème des moindres carrés ordinaires avec $\mathbf{A}_t = \sum_{i=1}^{mt} z_i z_i^\top \in \mathbb{R}^{d^2 \times d^2}$ et $b_t = \sum_{i=1}^{mt} z_i y_i \in \mathbb{R}^{d^2}$.

En suivant les étapes de [90], nous pouvons montrer que s'il existe $z \in \mathcal{Z}$ tel que pour tout $z' \in \mathcal{Z}$ ce qui suit est vrai :

$$\|z - z'\|_{\mathbf{A}_t^{-1}} \sqrt{8\sigma^2 \log\left(\frac{6m^2t^2K^4}{\delta\pi^2}\right)} \leq \hat{\Delta}_t(z, z') \ , \tag{C.8}$$

où $\hat{\Delta}_t(z, z') = (z - z')^\top \hat{\theta}_t$ est l'écart empirique entre z et z', alors avec une probabilité d'au moins $1-\delta$, l'estimation $\hat{\theta}_t$ conduit au meilleur bras de bord $z_\star$, ce qui signifie que $\arg\max_{z \in \mathcal{Z}} \langle z, \hat{\theta}_t \rangle = \arg\max_{z \in \mathcal{Z}} \langle z, \theta_\star \rangle$. Par conséquent, lorsque l'équation (C.8) est vraie, l'apprenant peut arrêter de tirer les bras, nous l'appelons la *condition d'arrêt*.

Comme mentionné dans [90], en remarquant que $\max_{(z,z') \in \mathcal{Z}^2} \|z - z'\|_{\mathbf{A}_t^{-1}} \leq 2 \max_{z \in \mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$, une stratégie admissible est de tirer des bras d'arête minimisant $\max_{z \in \mathcal{Z}} \|z\|_{\mathbf{A}_t^{-1}}$ afin de satisfaire la condition d'arrêt dès que possible.

Une strategie G-Allocation contrainte

Étant donné la condition d'arrêt (C.8) dérivée dans la section précédente, on veut trouver la séquence de bras d'arête $\mathbf{z}_{mt}^\star = (z_1^\star, \dots, z_{mt}^\star)$ telle que :

$$\mathbf{z}_{mt}^\star \in \operatorname*{arg\,min}_{(z_1, \dots, z_{mt}) \in \mathcal{Z}^{mt}} \max_{z' \in \mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i z_i^\top\right)^{-1} z' \ . \tag{G-opt-$\mathcal{Z}$}$$

Ceci est connu sous le nom de *G-allocation* (voir *e.g.*, [76, 90]) et est NP-Dur à calculer ([32, 104]). Une façon de trouver une solution approximative consiste à s'appuyer sur une relaxation

convexe du problème d'optimisation (G-opt-$\mathcal{Z}$) et à calculer d'abord une allocation à valeur réelle $\Gamma^\star \in \mathcal{S}_\mathcal{Z}$ telle que

$$\Gamma^\star \in \arg\min_{\Gamma \in \mathcal{S}_\mathcal{Z}} \max_{z' \in \mathcal{Z}} z'^\top \left(\sum_{z \in \mathcal{Z}} \Gamma_z z^\top \right)^{-1} z' \ . \tag{G-relaxed-$\mathcal{Z}$}$$

On pourrait soit utiliser un échantillonnage aléatoire pour tirer des bras d'arête comme *i.i.d.* échantillons de la distribution $\Gamma^\star$, soit des procédures d'arrondi pour convertir efficacement chaque $\Gamma^\star_z$ en un nombre entier. Cependant, ces méthodes ne prennent pas en compte la structure graphique du problème. En effet, à un tour donné, les arêtes choisies peuvent donner lieu à deux affectations différentes pour le même nœud, nous appelons ce phénomène une *collision*.

Par conséquent, les procédures d'échantillonnage aléatoire ou d'arrondi ne peuvent pas être utilisées directement pour sélectionner les bras d'arête dans $\mathcal{Z}$.

Néanmoins, (G-relaxed-$\mathcal{Z}$) donne tout de même des informations précieuses sur le nombre de fois, en proportion, où chaque bras d'arête $z \in \mathcal{Z}$ doit être allouée au graphe. Dans la section suivante, nous présentons un algorithme satisfaisant à la fois les exigences de proportion et les contraintes graphiques.

C.4.2 Algorithme et garanties

Allocation aléatoire sur les noeuds

Notre algorithme est basé sur une méthode aléatoire d'allocation directe des bras de nœud aux nœuds, évitant ainsi la tâche difficile de choisir les bras d'arête et d'essayer de les allouer au graphe tout en s'assurant que chaque nœud a une affectation unique. La validité de cette allocation aléatoire est basée sur le théorème C.4 ci-dessous montrant que l'on peut tirer des bras de noeuds dans $\mathcal{X}$ et les allouer au graphe de telle sorte que les bras d'arête associés suivent la distribution de probabilité $\Gamma^\star$ solution de (G-relaxed-$\mathcal{Z}$).

Theorem C.4. *Soit $\gamma^\star$ une solution du problème d'optimisation suivant :*

$$\min_{\gamma^star \in \mathcal{S}_\mathcal{X}} \max_{x' \in \mathcal{X}} x'^\top \left(\sum_{x \in \mathcal{X}} \gamma_x x x^\top \right)^{-1} x' \ . \tag{G-relaxed-$\mathcal{X}$}$$

Soit $\Gamma^\star \in \mathcal{S}_\mathcal{Z}$ défini pour tout $z = \mathrm{vec}\left(xx^{prime\top}\right) \in \mathcal{Z}$ par $\Gamma^\star_z = \gamma^\star_x \gamma^\star_{x'}$. Alors, $\Gamma^\star$ est une solution de (G-relaxed-$\mathcal{Z}$).

Ce théorème implique que, à chaque tour $t > 0$ et pour chaque nœud $i \in V$, si $x_t^{(i)}$ est tiré de $\gamma^\star$, alors pour toutes les paires de voisins $(i, j) \in E$, les bras d'arête associés $z_t^{(i,j)}$ suivent la distribution de probabilité $\Gamma^\star$. De plus, comme $\gamma^\star$ est une distribution sur l'ensemble des bras de nœud, $\mathcal{X}$, $\Gamma^\star$ peut etre considéré comme une distribution de probabilité jointe (produit) sur $\mathcal{X}^2$ avec pour marginale $\gamma^\star$.

Étant donné la caractérisation dans le Théorème C.4 et notre objectif de vérifier la condition d'arrêt dans (C.8), nous présentons notre procédure d'échantillonnage dans l'Algorithme 16. Nous

notons également qu'à chaque tour, l'échantillonnage des bras de nœuds peut être effectué en parallèle.

Algorithm 16: Algorithme d'Exploration Pure : G-Allocation randomisé pour GBB

Entrée : Graphe $\mathcal{G} = (V, E)$, ensemble de bras $\mathcal{X}$
Définir $A_0 = I$; $b_0 = 0$; $t = 1$;
Appliquer l'algorithme de Frank-Wolfe pour trouver la solution $\gamma^\star$ de (G-relaxed-$\mathcal{X}$).
while *la condition d'arrêt* (C.8) *n'est pas vérifiée* **do**
  // Échantillonnage des bras de nœud
  Tirer $x_t^{(1)}, \dots, x_t^{(n)} \overset{\text{iid}}{\sim} \gamma^\star$ et obtenir pour tout (i, j) dans E les récompenses $y_t^{(i,j)}$;
  // Estimation de $\hat{\theta}_t$ avec les bras d'arête associés
  $\mathbf{A}_t = \mathbf{A}_{t-1} + \sum_{(i,j)\in E} z_t^{(i,j)} z_t^{(i,j)\top}$;
  $b_t = b_{t-1} + \sum_{(i,j)\in E} z_t^{(i,j)} y_t^{(i,j)}$;
  $\hat{\theta}_t = \mathbf{A}_t^{-1} b_t$
  $t \leftarrow t + 1$;
end
retourner $\hat{\theta}_t$

Cette procédure d'échantillonnage implique que chaque bras d'arête suit la distribution optimale $\Gamma^\star$. Dans cette thèse, nous présentons une stratégie de G-allocation aléatoire simple et standard, mais d'autres méthodes plus élaborées pourraient être envisagées, tant qu'elles incluent le caractère aléatoire nécessaire.

Analyse de la convergence

Nous prouvons maintenant la validité de la procédure d'échantillonnage aléatoire détaillée dans l'Algorithme 16 en contrôlant la qualité de l'approximation $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ par rapport à l'optimum du problème d'optimisation G-allocation $\max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top}\right)^{-1} z'$ décrit dans (G-opt-$\mathcal{Z}$). Comme cela est généralement fait dans la littérature *optimal design* (voir *e.g.*, [76, 83, 90]), nous limitons l'erreur relative β_t :

$$\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z \leq (1+\beta_t) \max_{z'\in\mathcal{Z}} z'^\top \left(\sum_{i=1}^{mt} z_i^\star z_i^{\star\top}\right)^{-1} z' \ .$$

Notre analyse s'appuie sur plusieurs résultats de la théorie de la concentration matricielle. On peut se référer par exemple à [96] et à ses références pour une introduction approfondie sur ce sujet. Nous introduisons d'abord quelques notations supplémentaires.

Soit $f_\mathcal{Z}$ la fonction telle que, pour toute matrice non singulière $\mathbf{Q} \in \mathbb{R}^{d^2\times d^2}$, $f_\mathcal{Z}(\mathbf{Q}) = \max_{z\in\mathcal{Z}} z^\top \mathbf{Q}^{-1} z$ et pour toute distribution $\Gamma \in \mathcal{S}_\mathcal{Z}$ on rappelle que $\Sigma_\mathcal{Z}(\Gamma) \triangleq \sum_{z\in\mathcal{Z}} \Gamma_z z z^\top$ est la matrice de covariance associée. Enfin, laissons $\mathbf{A}_t^\star = \sum_{i=1}^{mt} z_i^\star z_i^{\star\top}$ être la matrice de G-optimal design construite pendant t tours.

Theorem C.5. *Soit $\Gamma^\star$ une solution du problème d'optimisation* (G-relaxed-$\mathcal{Z}$)*. Soit $0 < \delta \leq 1$ et soit t_0 tel que*

$$t_0 = 2Ld^2 \log\big(2d^2/\delta\big)/\lambda_{\min} \ ,$$

où $\lambda_{\min}$ est la plus petite valeur propre de la matrice de covariance $\frac{1}{K^2}\sum_{z\in\mathcal{Z}} zz^\top$. Alors, à chaque tour $t \geq t_0$ avec une probabilité d'au moins $1-\delta$, la stratégie randomisée G-allocation pour les bandits bilinéaires graphiques de l'Algorithme 16 produit une matrice $\mathbf{A}_t$ telle que :

$$f_{\mathcal{Z}}(\mathbf{A}_t) \leq (1+\beta_t) f_{\mathcal{Z}}(\mathbf{A}_t^\star) \ ,$$

où

$$\beta_t = \frac{Ld^2}{m\lambda_{\min}^2}\sqrt{\frac{2v}{t}\log\left(\frac{2d^2}{\delta}\right)} + o\left(\frac{1}{\sqrt{t}}\right) \ ,$$

et $v \triangleq \|\mathbb{E}\big[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2\big]\|$.

Nous venons de montrer que la valeur d'approximation $\max_{z\in\mathcal{Z}} z^\top \mathbf{A}_t^{-1} z$ converge vers la valeur optimale avec un taux de $O\big(\sqrt{v}/(m\sqrt{t})\big)$. Dans la section C.4.3, nous montrons que le meilleur cas de graphe implique un $v = O(m)$ correspondant au taux de convergence $O\big(1/\sqrt{mt}\big)$ d'un algorithme de bandit linéaire utilisant un échantillonnage aléatoire pour tirer mt edge-arms sans contraintes (graphiques). De plus, nous verrons que le graphique du pire cas implique que $v = O\big(m^2\big)$.

Puisque nous avons comblé l'écart entre notre objectif contraint et le problème de l'identification du meilleur bras dans les bandits linéaires, grâce au Théorème C.4 et C.5, nous sommes en mesure d'étendre les résultats connus pour l'identification du meilleur bras dans les bandits linéaires sur la complexité d'échantillonnage et sa borne inférieure associée.

Corollary C.2 ([90], Théorème 1)**.** *Si la G-allocation est mise en œuvre avec la stratégie aléatoire de l'Algorithme 16, résultant en une approximation de β_t, alors avec une probabilité d'au moins $1-\delta$, le meilleur bras obtenu avec $\hat{\theta}_t$ est $z_\star$ et*

$$t \leq \frac{128\sigma^2 d^2(1+\beta_t)\log\left(\frac{6m^2t^2K^4}{\delta\pi}\right)}{m\Delta_{\min}^2} \ ,$$

où $\Delta_{\min} = \min_{z\in\mathcal{Z}\setminus\{z_\star\}}(z_\star - z)^\top\theta_\star$.

De plus, soit τ le nombre de tours suffisant pour qu'un algorithme quelconque détermine le meilleur bras avec une probabilité d'au moins $1-\delta$. Une borne inférieure sur l'espérance de τ peut être obtenue à partir de celle dérivée pour le problème de l'identification du meilleur bras dans les bandits linéaires (voir *e.g.*, Théorème 1 dans [39]) :

$$\mathbb{E}[\tau] \geq \min_{\Gamma\in\mathcal{S}_{\mathcal{Z}}} \max_{z\in\mathcal{Z}\setminus\{z_\star\}} \log\left(\frac{1}{2.4\delta}\right)\frac{2\sigma^2\|z_\star - z\|^2_{\Sigma_{\mathcal{Z}}(\Gamma)^{-1}}}{m\Big((z_\star - z)^\top\theta_\star\Big)^2} \ .$$

Comme observé dans [90], cette limite inférieure peut être bornée, dans le pire des cas, par $4\sigma^2 d^2/(m\Delta_{\min}^2)$, ce qui correspond à notre borne jusqu'aux termes logarithmiques et à l'erreur relative β_t.

C.4.3 Influence de la structure du graphe sur le taux de convergence

Caractérisation de la variance associée à la stratégie aléatoire

La limite de convergence du théorème C.5 dépend de $v = \|\mathbb{E}\big[(\mathbf{A}_1 - \mathbb{E}\mathbf{A}_1)^2\big]\|$. Dans cette section, nous caractérisons l'impact de la structure du graphe sur cette quantité et, par extension, sur le taux de convergence. Les limites sont énoncées dans le tableau C.1.

Graphique	Limite supérieure sur $\|\mathrm{Var}(\mathbf{A}_1)\|$	β
Étoile	$mP + (M+N)O(m^2)$	$O\big(1/\sqrt{t}\big)$
Complète	$mP + (M+N)O(m\sqrt{m})$	$O\Big(1/\Big(m^{\frac{1}{4}}\sqrt{t}\Big)\Big)$
Cercle	$mP + (M+N)O(m)$	$O\big(1/\sqrt{mt}\big)$
Correspondance	$mP + mN$	$O\big(1/\sqrt{mt}\big)$

Table C.1: Borne supérieure de la variance et du taux de convergence de l'Algorithme 16 pour le graphe en étoile, le graphe complet, le cercle et le graphe couplage par rapport au nombre d'arêtes m et au nombre de tours t.

Ces quatre exemples mettent en évidence la forte dépendance de la variance à la structure du graphe. Plus les arêtes sont indépendantes (sans nœuds communs), plus la quantité $\|\mathrm{Var}(\mathbf{A}_1)\|$ est petite. Pour un nombre fixe d'arêtes m, le meilleur cas est le graphe couplage où aucune arête ne partage le même nœud et le pire cas est le graphe en étoile où toutes les arêtes partagent un nœud central.

Résultats expérimentaux validant la dépendance du graphe

Dans cette section, nous considérons la version modifiée d'une expérience standard introduite par [90] et utilisée dans la plupart des articles sur l'identification du meilleur bras dans les bandits linéaires [39, 92, 106, 109] pour évaluer la complexité d'échantillonnage de notre algorithme sur différents graphes.

Nous considérons $d+1$ bras de nœuds dans $\mathcal{X} \subset \mathbb{R}^d$ où $d \geq 2$. Cet ensemble est constitué des d vecteurs $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ formant la base canonique de $\mathbb{R}^d$ et d'un bras supplémentaire $x_{d+1} = (\cos(\omega), \sin(\omega), 0, \ldots, 0)^\top$ avec $\omega \in]0, \pi/2]$. Notez que par construction, l'ensemble des bras d'arête $\mathcal{Z}$ contient la base canonique $(\mathbf{e}'_1, \ldots, \mathbf{e}'_{d^2})$ de $\mathbb{R}^{d^2}$. La matrice $\mathbf{M}_\star$ a sa première coordonnée égale à 2 et les autres égales à 0, ce qui donne $\theta_\star = \mathrm{vec}\,(\mathbf{M}_\star) = (2, 0, \ldots, 0)^\top \in \mathbb{R}^{d^2}$. Le meilleur bras d'arête est donc $z_\star = z^{(1,1)} = \mathbf{e}'_1$. On peut noter que lorsque ω tend vers 0, il est plus difficile de différencier $z^{(1,1)}$ et $z^{(d+1,d+1)} = \mathrm{vec}\left(x_{(d+1)}x_{(d+1)}^\top\right)$ que $z^{(1,1)}$ et les autres bras. Nous fixons $\eta_t^{(i,j)} \sim \mathcal{N}(0,1)$, pour toute arête (i,j) et tour t.

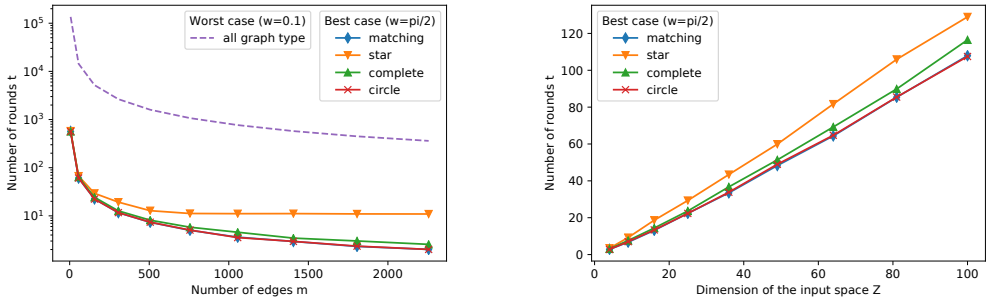


Figure C.2: Nombre de tours t nécessaires pour vérifier la condition d'arrêt (C.8) par rapport à **gauche:** le nombre d'arêtes m où la dimension de l'espace de $\mathcal{Z}$ est fixée et égale à 25 et **right:** la dimension de l'espace de $\mathcal{Z}$ où le nombre d'arêtes est fixé et égal à 156. Pour les deux expériences, nous les exécutons 100 fois et nous traçons le nombre moyen de tours nécessaires pour vérifier la condition d'arrêt.

Nous considérons deux cas où $\omega = 0.1$ qui rend les bras $z^{(1,1)}$ et $z^{(d+1,d+1)}$ difficiles à différencier, et $\omega = \pi/2$ qui rend le bras $z^{(1,1)}$ facilement identifiable comme le bras optimal. Pour chacun de ces deux cas, nous évaluons l'influence de la structure du graphe, du nombre d'arêtes m et de la dimension de l'espace des bras d'arête d^2 sur la complexité d'échantillonnage. Les résultats sont présentés dans la figure C.2.

Lorsque $\omega = 0, 1$, le type de graphe n'a pas d'impact sur le nombre de tours nécessaires pour vérifier la condition d'arrêt. Ceci est principalement dû au fait que l'ampleur de la variance associée est négligeable par rapport au nombre de tours. Par conséquent, même si nous faisons varier le nombre d'arêtes ou la dimension, nous obtenons les mêmes performances pour tout type de graphe, y compris le graphe matching. Cela implique que notre algorithme est aussi performant qu'un bandit linéaire qui tire m arêtes en parallèle à chaque tour. Lorsque $\omega = \pi/2$, le nombre de tours nécessaires pour vérifier la condition d'arrêt est plus petit et l'amplitude de la variance n'est plus négligeable. En effet, lorsque le nombre d'arêtes ou la dimension augmente, on remarque que le graphe en étoile prend plus de temps pour satisfaire la condition d'arrêt. De plus, notons que les complexités d'échantillonnage obtenues pour le cercle et le graphe d'appariement sont similaires. Cette observation est en accord avec la dépendance à la variance montrée dans le Tableau C.1.

C.5 Algorithmes basés sur le regret pour les bandits bilinéaires graphiques

Dans cette section, nous supposons également que nous ne connaissons pas la matrice des paramètres $\mathbf{M}_\star$ et comme décrit dans la configuration du problème dans la section C.2, une entité centrale fait face au problème des bandits bilinéaires graphiques où à chaque tour elle choisit un bras pour chaque nœud du graphe et observe une récompense bilinéaire pour chaque arête du graphe. L'objectif de l'entité centrale est de concevoir un algorithme qui maximise l'espérance de la récompense globale cumulée obtenue pendant T tours $\sum_{t=1}^{T} \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)}$. Nous nous appuierons naturellement sur certaines idées et résultats présentés dans la section C.3 où la matrice

était supposée connue par l'entité centrale. Nous suivons les notations que nous avons établies dans la section C.2.

C.5.1 Optimisme face à l'incertitude pour les bandits bilinéaires graphiques

Préliminaires

Rappelons que la maximisation des récompenses cumulées est équivalente à la minimisation du regret associé. Nous définissons donc le pseudo-regret global sur T tours comme suit :

$$R(T) = \sum_{t=1}^{T} \left[\sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} - \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \right] .$$

Nous rappelons que l'objectif de l'apprenant est d'avoir un pseudo-regret $R(T)$, tel que

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0$$

Nous savons du Théorème C.1 que trouver la meilleure allocation

$$\left(x_\star^{(1)}, \dots, x_\star^{(n)}\right) = \underset{(x^{(1)},\dots,x^{(n)})\in\mathcal{X}}{\arg\max} \sum_{(i,j)\in E} x^{(i)\top} \mathbf{M}_\star x^{(j)}$$

est NP-Dur par rapport au nombre d'agents n. Nous étendons ce résultat dans le corollaire suivant.

Corollary C.3. *Il n'existe pas d'algorithme en temps polynomial en n tel que*

$$\lim_{T\to\infty} \frac{R(T)}{T} = 0 \ , \tag{C.9}$$

pour toute instance de bandits bilinéaires graphiques décrits dans la section C.2, sauf si $P = NP$.

Par conséquent, l'objectif de concevoir un algorithme avec un regret sous-linéaire en T n'est pas réalisable en temps polynomial par rapport à n. Cependant, certains problèmes NP-Dur sont α-approximables (pour un certain $\alpha \in (0, 1]$), ce qui signifie qu'il existe un algorithme en temps polynomial garanti pour produire des solutions dont les valeurs sont au moins α fois la valeur de la solution optimale. Nous renvoyons le lecteur à la section C.3 pour plus d'informations sur l'approximation de la solution optimale de notre problème. Pour ce type de problèmes, il est logique de considérer l'α-pseudo-regret comme dans [31, 52] qui est défini pour tout $\alpha \in (0, 1]$ comme suit

$$R_\alpha(T) \triangleq \sum_{t=1}^{T} \left[\alpha \sum_{(i,j)\in E} x_\star^{(i)\top} \mathbf{M}_\star x_\star^{(j)} - \sum_{(i,j)\in E} x_t^{(i)\top} \mathbf{M}_\star x_t^{(j)} \right] ,$$

et on se fixe comme objectif de concevoir un algorithme avec un α-regret sous-linéaire.

Enfin, comme nous l'avons fait dans le chapitre C.4, rappelons que la récompense obtenue pour chaque arête du graphe à chaque tour peut être vue comme une récompense linéaire bruitée en dimension supérieure [51] avec

$$y_t^{(i,j)} = \left\langle \mathrm{vec}\left(x_t^{(i)} x_t^{(j)\top} \right), \mathrm{vec}\left(\mathbf{M}_\star \right) \right\rangle + \eta_t^{(i,j)} \ .$$

Pour simplifier la notation, désignons tout $x \in \mathcal{X}$ comme un *node-arm*. Utilisons la notation $z_{xx'} \triangleq \mathrm{vec}\left(xx'^\top\right)$, et définissons l'ensemble de bras $\mathcal{Z} = \{z_{xx'} | (x, x') \in \mathcal{X}^2\}$ où tout $z \in Z$ sera appelé un *bras d'arête*. Si le bras $x_t^{(i)} \in \mathcal{X}$ représente le bras de nœud alloué au nœud $i \in V$ au temps t, pour chaque arête $(i, j) \in E$ nous désignerons le bra d'arête associé par $z_t^{(i,j)} \triangleq \mathrm{vec}\left(x_t^{(i)} x_t^{(j)\top} \right) \in \mathcal{Z}$ et définissons $\theta_\star = \mathrm{vec}\left(\mathbf{M}_\star\right)$ la version vectorisée de la matrice inconnue $\mathbf{M}_\star$. Avec ces notations, la récompense (maintenant) linéaire peut être réécrite comme suit :

$$y_t^{(i,j)} = \left\langle z_t^{(i,j)}, \theta_\star \right\rangle + \eta_t^{(i,j)} \ . \tag{C.10}$$

Dans ce chapitre, nous choisissons de concevoir un algorithme basé sur le principe d'optimisme face à l'incertitude [12], et dans le cas d'une récompense linéaire [2, 63], nous devons maintenir un estimateur du vrai paramètre $\theta_\star$. Pour ce faire, définissons pour tous les tours $t \in \{1, \dots, T\}$ l'estimateur MCO de $\theta_\star$ comme suit :

$$\hat{\theta}_t = \mathbf{A}_t^{-1} b_t \ , \tag{C.11}$$

où,

$$\mathbf{A}_t = \lambda \mathbf{I}_{d^2} + \sum_{s=1}^{t} \sum_{(i,j)\in E} z_s^{(i,j)} z_s^{(i,j)\top} \ ,$$

avec $\lambda > 0$ un paramètre de régularisation et

$$b_t = \sum_{s=1}^{t} \sum_{(i,j)\in E} z_s^{(i,j)} y_s^{(i,j)} \ .$$

Nous définissons également l'ensemble de confiance

$$C_t(\delta) = \left\{ \theta : \|\theta - \hat{\theta}_t\|_{\mathbf{A}_t^{-1}} \leq \sigma \sqrt{d^2 \log\left(\frac{1 + tmL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} S \right\} \ ,$$

où avec une probabilité de $1-\delta$, on a que $\theta_\star \in C_t(\delta)$ pour tout $t \in \{1, \dots, T\}$, et $\delta \in (0, 1]$.

C.5.2 Algorithme et analyse du regret

Dans la section C.3, nous avons présenté deux algorithmes (Algorithm 14 et 15) qui utilisent la vraie matrice de paramètres $\mathbf{M}_\star$ pour renvoyer une allocation de bras permettant d'obtenir une α-approximation du problème de maximisation de la récompense globale. Naturellement, puisque nous n'avons pas accès à la matrice $\mathbf{M}_\star$, nous ne pouvons pas l'utiliser directement à chaque itération pour maximiser la récompense globale cumulative (et donc minimiser le regret associé). Néanmoins, on peut utiliser l'estimateur construit $\hat{\theta}_t$ et le principe d'optimisme face à l'incertitude pour surmonter le fait que $\mathbf{M}_\star$ est inconnu.

En effet, nous rappelons que dans l'Algorithme 14, le couple $(x_\star, x'_\star)$ est choisi comme suit,

$$(x_\star, x'_\star) = \underset{(x,x') \in \mathcal{X}^2}{\arg\max}\, x^\top \left(\mathbf{M}_\star + \mathbf{M}_\star^\top \right) x' \tag{C.12}$$

$$\tag{C.13}$$

$$= \underset{(x,x') \in \mathcal{X}^2}{\arg\max} \langle z_{xx'} + z_{x'x}, \theta_\star \rangle \ , \tag{C.14}$$

et est utilisé pour créer autant que possible des bras d'arêtes de la forme $z_{xx'}$ et $z_{x'x}$ dans le graphe. Ici, à chaque tour t, l'entité centrale choisit de manière optimiste le couple (x_t, x'_t) comme suit,

$$(x_t, x'_t) = \underset{(x,x') \in \mathcal{X}^2}{\arg\max} \max_{\theta \in C_{t-1}(\delta)} \langle z_{xx'} + z_{x'x}, \theta \rangle \ ,$$

puis alloue les bras-nœuds pour maximiser le nombre de bras-bords localement optimaux $z_{x_t x'_t}$ et $z_{x'_t x_t}$. La méthode est présentée dans l'Algorithme 17

Theorem C.6. *Étant donné le problème de bandits bilinéaires graphiques défini dans la section C.2, soit* $0 \leq \xi \leq 1$ *un paramètre dépendant du problème défini par*

$$\xi = \min_{x \in \mathcal{X}} \frac{\langle z_{xx}, \theta_\star \rangle}{\frac{1}{m} \sum_{(i,j) \in E} \left\langle z_\star^{(i,j)}, \theta_\star \right\rangle} \geq 0 \ ,$$

et on fixe $\alpha = \frac{1+\xi}{2}$*, alors l'*α*-regret de l'Algorithme 18 satisfait*

Algorithm 17: Adaptation de l'algorithme OFUL pour les bandits bilinéaires graphiques

Input : Graphe $\mathcal{G} = (V, E)$, ensemble $\mathcal{X}$
$(V_1, V_2) =$ Approx-MAX-CUT$(\mathcal{G})$
for $t = 1$ *à* T **do**
  // Trouver le meilleur couple optimiste
  $\left(x_t, x'_t, \tilde{\theta}_{t-1}\right) = \arg\max_{(x,x',\theta)\in\mathcal{X}^2\times C_{t-1}} \langle z_{xx'} + z_{x'x}, \theta\rangle$;
  // Allouer x_t et x'_t dans le graphe
  $x_t^{(i)} = x_t$ pour tout i dans V_1 ; $x_t^{(i)} = x'_t$ pour tout i dans V_2 ;
  Obtenir pour tout (i, j) dans E les récompenses $y_t^{(i,j)}$;
  Calculez $\hat{\theta}_t$ comme dans (C.11)
end
return $\hat{\theta}_t$.

$$R_\alpha(T) \leq \tilde{O}\left(\left(\sigma d^2 + S\sqrt{\lambda}\right)\sqrt{Tm\max(2,(LS)^2)}\right) + LSm\left\lceil d^2\log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ ,$$

où $\tilde{O}$ cache les facteurs logarithmiques.

C.5.3 Algorithme amélioré et analyse du regret

Nous abordons le problème de concevoir une version améliorée de l'algorithme proposé en utilisant l'idée présentée dans l'Algorithme 15 qui améliore le taux d'approximation.

Rappelons que dans la section précédente de l'Algorithme 17, l'entité centrale choisit le couple (x_t, x'_t) tel que

$$(x_t, x'_t) = \arg\max_{(x,x')\in\mathcal{X}^2} \max_{\theta\in C_{t-1}(\delta)} \langle z_{xx'} + z_{x'x}, \theta\rangle \ ,$$

qui maximise la récompense optimiste obtenue entre deux nœuds si l'entité centrale était capable d'allouer x_t à un nœud et x'_t au second. Cette stratégie étant optimale localement mais compliquée par la prise en compte des dépendances entre les arêtes, l'entité centrale pourrait prendre en considération les bras d'arêtes de la forme z_{xx} et $z_{x'x'}$ qui sont créés lors de l'allocation des nœuds du graphe en utilisant seulement deux bras de nœuds x et x'. Cette idée suit celle présentée dans l'Algorithme 15 où l'on rappelle avec des notations différentes que le couple $(\tilde{x}_\star, \tilde{x}'_\star)$ choisi pour allouer les noeuds du graphe sont tels que

$$(\tilde{x}_\star, \tilde{x}'_\star) = \arg\max_{(x,x')\in\mathcal{X}^2} \langle m_{1\to 2}\cdot z_{xx'} + m_{2\to 1}\cdot z_{x'x} + m_1 z_{xx} + m_2 z_{x'x'}, \theta_\star\rangle \ .$$

Comme dans la section précédente, nous n'avons pas accès à $\theta_\star$, nous utilisons le principe d'optimisme pour trouver à chaque tour le couple $(\tilde{x}_t, \tilde{x}'_t)$ tel que

$$(\tilde{x}_t, \tilde{x}'_t) = \underset{(x,x')\in\mathcal{X}^2}{\arg\max} \max_{\theta\in C_{t-1}(\delta)} \langle m_{1\to 2}\cdot z_{xx'} + m_{2\to 1}\cdot z_{x'x} = m_1 z_{xx} + m_2 z_{x'x'}, \theta\rangle \ . \tag{C.15}$$

Ici, au lieu de maximiser la récompense locale que l'on peut obtenir entre deux nœuds, l'entité centrale maximise la récompense optimiste globale que l'on obtiendrait en allouant seulement deux bras $(x, x') \in \mathcal{X}^2$ dans le graphe. Cette stratégie est décrite dans l'Algorithme 18.

Algorithm 18: OFUL amélioré pour les bandits bilineqires graphiques

Input : Graphe $\mathcal{G} = (V, E)$, ensemble $\mathcal{X}$
$(V_1, V_2) =$ Approx-MAX-CUT($\mathcal{G}$);
$m_{1\to 2} = |\{(i,j) \in E | i \in V_1 \wedge j \in V_2\}|$;
$m_{2\to 1} = |\{(i,j) \in E | i \in V_2 \wedge j \in V_1\}|$;
$m_1 = |\{(i,j) \in E | i \in V_1 \wedge j \in V_1\}|$;
$m_2 = |\{(i,j) \in E | i \in V_2 \wedge j \in V_2\}|$;
for $t = 1$ *to* T **do**
  $\left(\tilde{x}_t, \tilde{x}'_t, \tilde{\theta}_{t-1}\right) =$
  $\arg\max_{(x,x',\theta)\in\mathcal{X}^2\times C_{t-1}} \langle m_{1\to 2}\cdot z_{xx'} + m_{2\to 1}\cdot z_{x'x} + m_1\cdot z_{xx} + m_2\cdot z_{x'x'}, \theta\rangle$;
  $x_t^{(i)} = \tilde{x}_t$ pour tout i dans V_1;
  $x_t^{(i)} = \tilde{x}'_t$ pour tout i dans V_2;
  Obtenir pour tout (i,j) dans E les récompenses $y_t^{(i,j)}$;
  Calculer $\hat{\theta}_t$ comme dans (C.11)
end
retourner $\hat{\theta}_t$

Avant d'énoncer les garanties sur l'α-regret, nous rappelons que nous avons défini dans la section C.3 la quantité $\Delta \geq 0$ comme étant la différence entre la récompense de l'allocation $(\tilde{x}_\star, \tilde{x}'_\star)$ et celle de l'allocation $(x_\star, x'_\star)$,

$$\begin{aligned}\Delta = \langle & m_{1\to 2}\left(z_{\tilde{x}_\star \tilde{x}'_\star} - z_{x_\star x'_\star}\right) + m_{2\to 1}\left(z_{\tilde{x}'_\star \tilde{x}_\star} - z_{x'_\star x_\star}\right) \\ & + m_1\left(z_{\tilde{x}_\star \tilde{x}_\star} - z_{x_\star x_\star}\right) + m_2\left(z_{\tilde{x}'_\star \tilde{x}'_\star} - z_{x'_\star x'_\star}\right), \theta_\star\rangle \ .\end{aligned}$$

Les nouvelles garanties que nous obtenons sur l'α-regret de l'Algorithme 18 sont énoncées dans le théorème suivant.

Theorem C.7. *Étant donné le problème de bandits bilinéaires graphiques défini dans la section C.2, on définit ξ comme dans le théorème C.6, laissez $0 \leq \epsilon \leq \frac{1}{2}$ être un paramètre dépendant du prob-*

lème qui mesure le gain de l'optimisation sur les bras sous-optimaux et indésirables défini comme :

$$\epsilon = \frac{\Delta}{\sum_{(i,j)\in E}\langle z_\star^{(i,j)}, \theta_\star\rangle} \ ,$$

et on fixe $\alpha = \frac{1+\xi}{2} + \epsilon$ *où* $\alpha \geq 1/2$ *par construction, alors le* α*-regret de l'Algorithme 18 satisfait*

$$R_\alpha(T) \leq \tilde{O}\left(\left(\sigma d^2 + S\sqrt{\lambda}\right)\sqrt{Tm\max(2,(LS)^2)}\right) + LSm\left\lceil d^2 \log_2\left(\frac{TmL^2/\lambda}{\delta}\right)\right\rceil \ ,$$

où $\tilde{O}$ *cache les facteurs logarithmiques.*

On peut voir ici que l'amélioration se produit dans le α du α-regret. Dans la section suivante, nous confirmons ces résultats par des expériences.

C.5.4 Expériences numériques

Nous concevons une expérience qui compare en pratique les performances de l'Algorithme 17 et de l'Algorithme 18 avec l'algorithme Explore-Then-Commit (ETC) en utilisant la stratégie d'exploration conçue dans la section C.4 pendant la phase d'exploration, et en allouant les nœuds dans V_1 et V_2 avec le meilleur couple estimé $(x, x') = \arg\max_{(x,x')}\langle z_{xx'} + z_{x'x}, \hat{\theta}_t\rangle$ pendant la phase de d'exploitation. Cependant, puisque les algorithmes que nous avons présentés dans cette section ont des garanties sur les α-regrets avec différents α, nous traçons la fraction de la récompense globale optimale pour chaque itération.

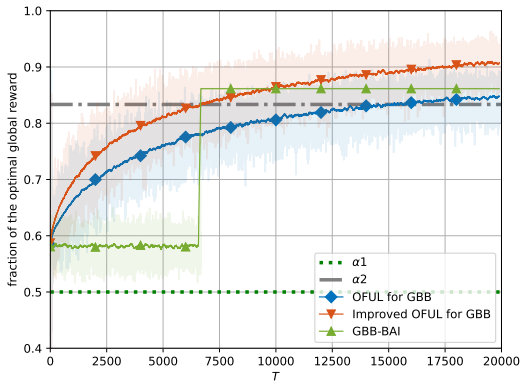


Figure C.3: Fraction de la récompense globale optimale obtenue à chaque tour en appliquant l'Algorithme 17, l'Algorithme 18 et l'algorithme Explore-Then-commit (appelé ici GBB-BAI) en utilisant la stratégie d'exploration de la Section C.4. Nous utilisons un graphe complet de 5 nœuds, nous exécutons l'expérience sur 5 matrices différentes avec $\zeta = 0$ et l'exécutons 10 fois différentes pour tracer la fraction moyenne de la récompense globale

.

Comme dans le chapitre C.3, nous observons une nette amélioration en choisissant à chaque tour t le couple de bras $(\tilde{x}_t, \tilde{x}'_t)$ au lieu de (x_t, x'_t).

C.6 Conclusion & perspectives

C.6.1 Résumé des résultats

Dans cette thèse, nous avons introduit un nouveau modèle que nous avons nommé *Bandits Bilinéaires Graphiques* qui modélise les problèmes multi-agents centralisés où des interactions par paires existent entre les agents.

- Dans la section C.3, nous avons mis en évidence le fait que l'apprenant était confronté à un problème d'optimisation sous-jacent qui est NP-Hard quel que soit le but que l'apprenant souhaite atteindre. Nous avons donc proposé un algorithme d'α-approximation avec $\alpha \geq 1/2$ qui ne nécessite que de trouver le couple de bras $(x_\star, x'_\star)$ pour retourner l'α-approximation. Nous avons ensuite affiné ce paramètre d'approximation par rapport aux paramètres dépendant du problème en nous basant sur la structure du graphe et sur une propriété de la matrice $\mathbf{M}_\star$.

- Dans la section C.4, étant donné l'algorithme d'α-approximation conçu dans la section C.3, nous avons présenté un algorithme de pure exploration qui permettait à l'apprenant de construire une estimation $\hat{\mathbf{M}}$ qui était statistiquement efficace en termes d'optimal design. En effet, le problème de trouver en un nombre minimum de tours le meilleur couple $(x_\star, x'_\star)$ utilisé dans l'algorithme d'approximation α revenait à trouver le G-optimal design, également appelé G-allocation dans la littérature bandit. Résoudre ce problème dans les bandits bilinéaires graphiques impliquait de traiter une contrainte supplémentaire. C'est pourquoi nous avons présenté un algorithme qui respectait cette contrainte et qui utilisait un échantillonnage aléatoire pour construire l'estimation $\hat{\mathbf{M}}$. Nos résultats théoriques ont révélé un terme qui dépendait de la structure du graphe, nous avons donc montré l'impact du graphe dans nos résultats.

- Enfin, dans la section C.5, nous avons capitalisé sur l'algorithme d'α-approximation donné dans le chapitre C.3 et appliqué le principe d'optimisme face à l'incertitude pour concevoir des algorithmes basés sur le regret qui ont atteint un α-regret sous-linéaire en T où $\alpha \geq 1/2$. De plus, nous avons présenté expérimentalement les performances des algorithmes proposés et utilisé en comparaison un algorithme Explore-Then-Commit s'appuyant sur l'algorithme d'exploration pure présenté dans le chapitre C.4.

C.6.2 Perspective et travaux futurs

Cette thèse avait pour but d'introduire le nouveau cadre du bandit bilinéaire graphique et de fournir les premières solutions aux problèmes courants posés dans la littérature sur le bandit. De nombreuses autres approches et modifications peuvent être envisagées. Nous en présentons deux dans ce qui suit.

Des matrices de paramètres différentes $\mathbf{M}_\star^{(i,j)}$ pour chaque arête $(i,j) \in E$. Alors que le traitement d'une matrice commune de paramètres $\mathbf{M}_\star$ pour toutes les arêtes $(i,j) \in E$ était pratique pour agréger les récompenses et construire une estimation commune $\hat{\mathbf{M}}$ pour tous les agents, lorsque les récompenses $y_t^{(i,j)}$ sont définies avec des matrices différentes $\mathbf{M}_\star^{(i,j)}$, le problème devient plus compliqué. En effet, considérons le cas où la récompense $y_t^{(i,j)}$ est définie comme suit pour chaque $(i,j) \in E$:

$$y_t^{(i,j)} = x_t^{(i)\top} \mathbf{M}_\star^{(i,j)} x_t^{(j)} + \eta_t^{(i,j)} \ ,$$

où $\mathbf{M}_\star^{(i,j)}$ sont des matrices à paramètres inconnus et $\eta_t^{(i,j)}$ des variables aléatoires σ-sous-gaussiennes.

Ce paramètre est pertinent lorsque les agents n'ont pas les mêmes interactions entre chacun de leurs voisins, et donc pas la même fonction de récompense.

Question ouverte : Dans le contexte de l'exploration pure, comment la condition d'arrêt change-t-elle ? Existe-t-il une stratégie d'échantillonnage pour chaque agent telle que les estimations $\hat{\mathbf{M}}^{(i,j)}$ sont construites en satisfaisant un critère d'optimal design ?

Cadre décentralisé . Lorsque les agents sont contrôlés par une entité centrale, il est possible d'agréger les différentes récompenses et de construire une estimation commune $\hat{\mathbf{M}}$ de $\mathbf{M}_\star$. De plus, nous avons vu que les différents objectifs qui apparaissent sont relatifs aux bras d'arêtes et non directement aux bras de nœuds sélectionnés par chaque agent. En effet, cela est dû au fait que nous pouvons exprimer le bandit bilinéaire graphique comme des bandits linéaires au niveau des arêtes. Cet aspect particulier rend le cadre décentralisé un peu délicat car la coordination de deux agents sans communication pour tirer respectivement les bras de nœuds qui construiront bras d'arêtes désirés devient encore plus compliqué.

Cependant, d'autres problèmes surgissent même si le problème de coordination est résolu. Par exemple, dans le problème d'identification du meilleur bras, nous avons déjà conçu une procédure d'échantillonnage qui peut être exécutée en parallèle pendant un tour, d'où un choix décentralisé pour chaque agent. Cependant, la condition d'arrêt dépend de l'estimation $\hat{\mathbf{M}}$ construite avec les bras d'arêtes pendant la procédure d'apprentissage, mais lorsque les agents ne communiquent pas, cette estimation ne peut pas être construite. En effet, un agent ne connaît que le bras qu'il tire et observe la récompense. Or, la récompense est linéaire par rapport à au bras d'arête associée et l'agent n'a pas accès à ce bras d'arête puisqu'il est construit avec son bras de nœud mais aussi avec celui de ses voisins (auquel il n'a pas accès).

Questions ouvertes : Dans le cadre d'une décentralisation totale (sans communication), quel type d'algorithmes pouvons-nous concevoir pour tirer parti du cadre du bandit linéaire ? Si nous autorisons la communication, comment pouvons-nous adapter les algorithmes proposés et quel est le compromis entre la quantité de communication et la performance ?

Bibliography

1. M. Abbasi and C. Gagné. "Robustness to adversarial examples through an ensemble of specialists". *arXiv preprint arXiv:1702.06856*, 2017 (cited on page 105).
2. Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. "Improved algorithms for linear stochastic bandits". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2312–2320 (cited on pages 14, 15, 58, 65, 98, 99, 100, 150).
3. M. Agarwal, V. Aggarwal, and K. Azizzadenesheli. "Multi-agent multi-armed bandits with limited communication". *arXiv preprint arXiv:2102.08462*, 2021 (cited on pages 2, 134).
4. R. Agrawal. "Sample mean based index policies by o (log n) regret for the multi-armed bandit problem". *Advances in Applied Probability* 27:4, 1995, pp. 1054–1078 (cited on page 9).
5. T. Ajanthan et al. "Optimization of Markov random fields in computer vision", 2017 (cited on pages 24, 137).
6. S. Amani and C. Thrampoulidis. "Decentralized multi-agent linear bandits with safety constraints". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8. 2021, pp. 6627–6635 (cited on pages 2, 134).
7. K. Amin, M. Kearns, and U. Syed. "Graphical Models for Bandit Problems". In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. 2011, pp. 1–10 (cited on pages 2, 19, 24, 134, 137).
8. M. S. Andersen, J. Dahl, and L. Vandenberghe. "CVXOPT: A Python package for convex optimization, version 1.2.0" (cited on page 99).
9. A. Athalye, N. Carlini, and D. Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, Stockholmsmässan, Stockholm Sweden, 2018, pp. 274–283 (cited on pages 104, 115).
10. J.-Y. Audibert and S. Bubeck. "Best arm identification in multi-armed bandits". In: *Proceedings of the 23th Annual Conference on Learning Theory*. 2010, pp. 41–53 (cited on pages 3, 136).
11. P. Auer. "Using confidence bounds for exploitation-exploration trade-offs". *Journal of Machine Learning Research* 3:Nov, 2002, pp. 397–422 (cited on pages 1, 134).
12. P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multiarmed bandit problem". *Machine learning* 47:2-3, 2002, pp. 235–256 (cited on pages 9, 10, 58, 150).

13. E. Bargiacchi, T. Verstraeten, D. Roijers, A. Nowé, and H. Hasselt. "Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems". In: *International conference on machine learning*. 2018, pp. 482–490 (cited on pages 1, 2, 19, 133, 134).

14. P. Berman and M. Karpinski. "On some tighter inapproximability results". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 1999, pp. 200–209 (cited on page 57).

15. L. Besson and E. Kaufmann. "Multi-player bandits revisited". In: *Algorithmic Learning Theory*. PMLR. 2018, pp. 56–92 (cited on pages 2, 134).

16. A. N. Bhagoji, D. Cullina, and P. Mittal. "Lower Bounds on Adversarial Robustness from Optimal Transport". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 7496–7508 (cited on page 105).

17. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402 (cited on page 104).

18. I. Bistritz and A. Leshem. "Distributed multi-player bandits-a game of thrones approach". *Advances in Neural Information Processing Systems* 31, 2018 (cited on pages 2, 134).

19. S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013 (cited on page 91).

20. S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004 (cited on page 79).

21. G. W. Brown. "Iterative solution of games by fictitious play". *Activity analysis of production and allocation* 13:1, 1951, pp. 374–376 (cited on page 112).

22. M. Brückner and T. Scheffer. "Stackelberg Games for Adversarial Prediction Problems". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. Association for Computing Machinery, San Diego, California, USA, 2011, pp. 547–555. ISBN: 9781450308137 (cited on page 105).

23. S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn. "Adversarial examples from computational constraints". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, Long Beach, California, USA, 2019, pp. 831–840 (cited on page 104).

24. S. Bubeck, R. Munos, and G. Stoltz. "Pure exploration in multi-armed bandits problems". In: *International conference on Algorithmic learning theory*. Springer. 2009, pp. 23–37 (cited on pages 3, 136).

25. O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, and G. Stoltz. "Kullback-Leibler upper confidence bounds for optimal sequential allocation". *The Annals of Statistics*, 2013, pp. 1516–1541 (cited on page 10).

26. N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. "On Evaluating Adversarial Robustness". *arXiv preprint arXiv:1902.06705*, 2019 (cited on page 115).

27. N. Carlini and D. Wagner. "Towards evaluating the robustness of neural networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 39–57 (cited on pages 104, 108, 128).

28. N. Cesa-Bianchi, C. Gentile, and G. Zappella. "A gang of bandits". In: *Advances in Neural Information Processing Systems*. 2013, pp. 737–745 (cited on pages 2, 18, 134).

29. L. F. O. Chamon and A. Ribeiro. "Approximate supermodularity bounds for experimental design". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5403–5412 (cited on page 78).

30. J. Chan, A. Pacchiano, N. Tripuraneni, Y. S. Song, P. Bartlett, and M. I. Jordan. "Parallelizing contextual linear bandits". *arXiv preprint arXiv:2105.10590*, 2021 (cited on pages 2, 17, 61, 65, 66, 134).

31. W. Chen, Y. Wang, and Y. Yuan. "Combinatorial multi-armed bandit: General framework and applications". In: *International Conference on Machine Learning*. 2013, pp. 151–159 (cited on pages 20, 57, 149).

32. A. Çivril and M. Magdon-Ismail. "On selecting a maximum volume sub-matrix of a matrix and related problems". *Theoretical Computer Science* 410:47, 2009, pp. 4801–4811 (cited on pages 12, 38, 78, 79, 143).

33. J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. "Certified Adversarial Robustness via Randomized Smoothing". *CoRR* abs/1902.02918, 2019. arXiv: 1902.02918 (cited on pages 104, 105).

34. S. Damla Ahipasaoglu, P. Sun, and M. J. Todd. "Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids". *Optimisation Methods and Software* 23:1, 2008, pp. 5–19 (cited on page 41).

35. G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. "Stochastic activation pruning for robust adversarial defense". In: *International Conference on Learning Representations*. 2018 (cited on pages 105, 115).

36. M. T. van Dijk, J.-W. van Wingerden, T. Ashuri, Y. Li, and M. A. Rotea. "Yaw-misalignment and its impact on wind turbine loads and wind farm power output". In: *Journal of Physics: Conference Series*. Vol. 753. IOP Publishing. 2016, p. 062013 (cited on pages 1, 133).

37. A. Fawzi, H. Fawzi, and O. Fawzi. "Adversarial vulnerability for any classifier". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 1186–1195 (cited on page 104).

38. A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. "Robustness of classifiers: from adversarial to random noise". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 1632–1640 (cited on page 104).

39. T. Fiez, L. Jain, K. G. Jamieson, and L. Ratliff. "Sequential experimental design for transductive linear bandits". In: *Advances in Neural Information Processing Systems*. 2019, pp. 10667–10677 (cited on pages 42, 43, 48, 52, 146, 147).
40. P. Flaherty, A. Arkin, and M. I. Jordan. "Robust design of biological experiments". In: *Advances in Neural Information Processing Systems 18*. 2006, pp. 363–370 (cited on page 78).
41. M. Frank, P. Wolfe, et al. "An algorithm for quadratic programming". *Naval research logistics quarterly* 3:1-2, 1956, pp. 95–110 (cited on page 41).
42. Y. Freund and R. E. Schapire. "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Second European Conference on Computational Learning Theory (EuroCOLT-95)*. Ed. by P. M. B. Vitányi. Aix-en-Provence, France, 1995, pp. 23–37 (cited on page 112).
43. A. Frieze and M. Jerrum. "Improved approximation algorithms for max k-cut and max bisection". *Algorithmica* 18:1, 1997, pp. 67–81 (cited on page 34).
44. M. X. Goemans and D. P. Williamson. ". 879-approximation algorithms for max cut and max 2sat". In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. 1994, pp. 422–431 (cited on pages 25, 138).
45. I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. 2015 (cited on pages 104, 110).
46. P. Gourdeau, V. Kanade, M. Kwiatkowska, and J. Worrell. "On the Hardness of Robust Classification". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 7444–7453 (cited on page 104).
47. C. Guestrin, M. G. Lagoudakis, and R. Parr. "Coordinated Reinforcement Learning". In: *Proceedings of the Nineteenth International Conference on Machine Learning*. 2002, pp. 227–234 (cited on pages 1, 133).
48. W. He, J. Wei, X. Chen, N. Carlini, and D. Song. "Adversarial example defense: Ensembles of weak defenses are not strong". In: *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*. 2017 (cited on page 105).
49. A. Heliou, J. Cohen, and P. Mertikopoulos. "Learning with bandit feedback in potential games". *Advances in Neural Information Processing Systems* 30, 2017 (cited on pages 2, 134).
50. A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. "Adversarial Examples Are Not Bugs, They Are Features". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 125–136 (cited on page 104).
51. K.-S. Jun, R. Willett, S. Wright, and R. Nowak. "Bilinear Bandits with Low-rank Structure". In: *International Conference on Machine Learning*. 2019, pp. 3163–3172 (cited on pages 15, 16, 58, 150).
52. S. M. Kakade, A. T. Kalai, and K. Ligett. "Playing games with approximation algorithms". *SIAM Journal on Computing* 39:3, 2009, pp. 1088–1106 (cited on pages 20, 57, 149).

53. J. Kiefer and J. Wolfowitz. "The Equivalence of Two Extremum Problems". *Canadian Journal of Mathematics* 12, 1960, pp. 363–366 (cited on pages 40, 45, 46).

54. V. Koltchinskii and K. Lounici. "Asymptotics and concentration bounds for bilinear forms of spectral projectors of sample covariance". In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 52. 4. Institut Henri Poincaré. 2016, pp. 1976–2013 (cited on page 80).

55. A. Krizhevsky and G. Hinton. *Learning multiple layers of features from tiny images*. Technical report. Citeseer, 2009 (cited on page 115).

56. A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images", 2009 (cited on page 128).

57. T. L. Lai and H. Robbins. "Asymptotically efficient adaptive allocation rules". *Advances in applied mathematics* 6:1, 1985, pp. 4–22 (cited on page 9).

58. P. Landgren, V. Srivastava, and N. E. Leonard. "Distributed cooperative decision making in multi-agent multi-armed bandits". *Automatica* 125, 2021, p. 109445 (cited on pages 2, 134).

59. T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2018 (cited on pages 1, 7, 66, 99, 134).

60. M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. "Certified Robustness to Adversarial Examples with Differential Privacy". In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 727–743 (cited on page 105).

61. J. Lee and M. Raginsky. "Minimax Statistical Learning with Wasserstein distances". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2687–2696 (cited on page 104).

62. B. Li, C. Chen, W. Wang, and L. Carin. "Certified Adversarial Robustness with Additive Noise". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 9459–9469 (cited on page 105).

63. 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*. 2010, pp. 661–670 (cited on pages 58, 150).

64. E. H. Lieb. "Convex trace functions and the Wigner-Yanase-Dyson conjecture". *Les rencontres physiciens-mathématiciens de Strasbourg-RCP25* 19, 1973, pp. 0–35 (cited on page 85).

65. Y. Luo, X. Zhao, J. Zhou, J. Yang, Y. Zhang, W. Kuang, J. Peng, L. Chen, and J. Zeng. "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information". *Nature communications* 8:1, 2017, pp. 1–13 (cited on page 16).

66. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018 (cited on pages 104, 110, 114, 128).

67. S. Mannor and O. Shamir. "From bandits to experts: On the value of side-observations". In: *Advances in Neural Information Processing Systems*. 2011, pp. 684–692 (cited on page 18).

68. S. Minsker. "On some extensions of Bernstein's inequality for self-adjoint operators". *arXiv preprint arXiv:1112.5448*, 2011 (cited on page 80).
69. W. I. Notz. "Optimal designs for treatment—control comparisons in the presence of two-way heterogeneity". *Journal of Statistical Planning and Inference* 12, 1985, pp. 61–73 (cited on page 78).
70. T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu. "Improving adversarial robustness via promoting ensemble diversity". *arXiv preprint arXiv:1901.08846*, 2019 (cited on page 105).
71. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. "The limitations of deep learning in adversarial settings". In: *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE. 2016, pp. 372–387 (cited on page 104).
72. N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597 (cited on page 104).
73. J. C. Perdomo and Y. Singer. "Robust Attacks against Multiple Classifiers". *CoRR* abs/1906.02816, 2019. arXiv: 1906.02816 (cited on page 105).
74. K. B. Petersen and M. S. Pedersen. "The matrix cookbook, nov 2012". *URL http://www2.imm. dtu. dk/pubdb/p. php* 3274, 2012, p. 14 (cited on page 41).
75. R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, and J. Atif. "Theoretical evidence for adversarial robustness through randomization". In: *Advances in Neural Information Processing Systems 32 (NeurIPS)*. 2019 (cited on pages 105, 115).
76. F. Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006 (cited on pages 12, 38, 43, 78, 79, 143, 145).
77. M. S. Pydi and V. Jog. *Adversarial Risk via Optimal Transport and Optimal Couplings*. 2019. arXiv: 1912.02794 [cs.LG] (cited on page 105).
78. G. Rizk, I. Colin, A. Thomas, and M. Draief. "Refined bounds for randomized experimental design". *NeurIPS Workshop on Machine Learning with Guarantees*, 2019 (cited on page 44).
79. H. Robbins. "Some aspects of the sequential design of experiments". *Bulletin of the American Mathematical Society* 58:5, 1952, pp. 527–535 (cited on page 8).
80. S. Rosa and R. Harman. "Optimal approximate designs for comparison with control in dose-escalation studies". *TEST* 26:3, 2017, pp. 638–660 (cited on page 78).
81. S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo, and F. Roli. "Randomized Prediction Games for Adversarial Machine Learning". *IEEE Transactions on Neural Networks and Learning Systems* 28:11, 2017, pp. 2466–2478 (cited on page 105).
82. G. Sagnol. "Approximation of a maximum-submodular-coverage problem involving spectral functions, with application to experimental designs". *Discrete Applied Mathematics* 161:1, 2013, pp. 258–276 (cited on page 78).
83. G. Sagnol. "Optimal design of experiments with application to the inference of traffic matrices in large networks: second order cone programming and submodularity". PhD thesis. École Nationale Supérieure des Mines de Paris, 2010 (cited on pages 43, 78, 145).

84. S. Sahni and T. Gonzalez. "P-complete approximation problems". *Journal of the ACM (JACM)* 23:3, 1976, pp. 555–565 (cited on pages 25, 26, 138).

85. A. Sankararaman, A. Ganesh, and S. Shakkottai. "Social learning in multi agent multi armed bandits". *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3:3, 2019, pp. 1–35 (cited on pages 2, 134).

86. S. Sen, B. Ravindran, and A. Raghunathan. "EMPIR: Ensembles of Mixed Precision Deep Networks for Increased Robustness against Adversarial Attacks". *arXiv preprint arXiv:2004.10162*, 2020 (cited on page 105).

87. S. Shahrampour, A. Rakhlin, and A. Jadbabaie. "Multi-armed bandits in multi-agent networks". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 2786–2790 (cited on pages 2, 134).

88. A. Sinha, H. Namkoong, and J. Duchi. "Certifiable Distributional Robustness with Principled Adversarial Training". In: *International Conference on Learning Representations*. 2018 (cited on page 104).

89. I. Siomina, P. Varbrand, and D. Yuan. "Automated optimization of service coverage and base station antenna configuration in UMTS networks". *IEEE Wireless Communications* 13:6, 2006, pp. 16–25 (cited on pages 1, 133).

90. M. Soare, A. Lazaric, and R. Munos. "Best-arm identification in linear bandits". In: *Advances in Neural Information Processing Systems*. 2014, pp. 828–836 (cited on pages 11, 12, 14, 37, 38, 42, 43, 48, 52, 78, 81, 98, 99, 100, 101, 143, 145, 146, 147).

91. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014 (cited on page 104).

92. C. Tao, S. Blanco, and Y. Zhou. "Best Arm Identification in Linear Bandits with Linear Dimension Dependency". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. 2018, pp. 4877–4886 (cited on pages 52, 78, 98, 100, 147).

93. W. R. Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika* 25:3-4, 1933, pp. 285–294 (cited on page 7).

94. F. Tramer, N. Carlini, W. Brendel, and A. Madry. "On adaptive attacks to adversarial example defenses". *arXiv preprint arXiv:2002.08347*, 2020 (cited on pages 105, 115, 129, 130).

95. F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. "The Space of Transferable Adversarial Examples". *arXiv*, 2017 (cited on page 110).

96. J. A. Tropp et al. "An introduction to matrix concentration inequalities". *Foundations and Trends® in Machine Learning* 8:1-2, 2015, pp. 1–230 (cited on pages 43, 44, 82, 90, 94, 145).

97. J. A. Tropp. "An Introduction to Matrix Concentration Inequalities". *Foundations and Trends® in Machine Learning* 8:1-2, 2015, pp. 1–230. ISSN: 1935-8237 (cited on page 80).

98. M. Valko. "Bandits on graphs and structures", 2020 (cited on page 18).

99. M. Valko, R. Munos, B. Kveton, and T. Kocák. "Spectral Bandits for Smooth Graph Functions". In: *International conference on machine learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research. 2014, pp. 46–54 (cited on page 18).

100. G. Verma and A. Swami. "Error Correcting Output Codes Improve Probability Estimation and Adversarial Robustness of Deep Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 8643–8653 (cited on page 105).

101. D. Vial, S. Shakkottai, and R. Srikant. "Robust multi-agent multi-armed bandits". In: *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 2021, pp. 161–170 (cited on pages 2, 134).

102. B. Wang, Z. Shi, and S. Osher. "ResNets Ensemble via the Feynman-Kac Formalism to Improve Natural and Robust Accuracies". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 1655–1665 (cited on page 105).

103. Y. Wang, A. W. Yu, and A. Singh. "On computationally tractable selection of experiments in measurement-constrained regression models". *The Journal of Machine Learning Research* 18:1, 2017, pp. 5238–5278 (cited on page 78).

104. W. Welch. "Algorithmic Complexity: Three NP-Hard Problems in Computational Statistics". *Journal of Statistical Computation and Simulation - J STAT COMPUT SIM* 15, 1982, pp. 17–25 (cited on pages 12, 38, 78, 79, 143).

105. C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. "Mitigating Adversarial Effects Through Randomization". In: *International Conference on Learning Representations*. 2018 (cited on page 105).

106. L. Xu, J. Honda, and M. Sugiyama. "A fully adaptive algorithm for pure exploration in linear bandits". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Vol. 84. 2018, pp. 843–851 (cited on pages 52, 78, 98, 100, 147).

107. W. Xu, D. Evans, and Y. Qi. "Feature squeezing mitigates and detects carlini/wagner adversarial examples". *arXiv preprint arXiv:1705.10686*, 2017 (cited on page 105).

108. S. Zagoruyko and N. Komodakis. "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2016, pp. 87.1–87.12. ISBN: 1-901725-59-6 (cited on page 128).

109. M. Zaki, A. Mohan, and A. Gopalan. "Towards Optimal and Efficient Best Arm Identification in Linear Bandits". *arXiv preprint arXiv:1911.01695*, 2019 (cited on pages 52, 147).

RÉSUMÉ

Nous introduisons un nouveau modèle appelé *Bandits Bilinéaires Graphiques* où un apprenant (ou une entité centrale) alloue des bras aux noeuds d'un graphe et observe pour chaque arête une récompense bilinéaire bruitée représentant l'interaction entre les deux noeuds associés. Dans cette thèse, nous étudions le problème d'identification du meilleur bras et la maximisation des récompenses cumulées. Pour le premier, un apprenant veut trouver l'allocation du graphe maximisant la somme des récompenses bilinéaires obtenues à travers le graphe. Pour le second problème, au cours du processus d'apprentissage, l'apprenant doit faire un compromis entre l'exploration des bras pour acquérir une connaissance précise de l'environnement et l'exploitation des bras qui semblent être les meilleurs pour obtenir la récompense la plus élevée. Quel que soit l'objectif de l'apprenant, le modèle de bandits bilinéaires graphiques révèle un problème combinatoire sous-jacent qui est NP-Dur et qui empêche l'utilisation de tout algorithme existant pour l'identification du meilleur bras (BAI) ou pour la maximisation des récompenses cumulées. Pour cette raison, nous proposons tout d'abord un algorithme d'α-approximation pour le problème NP-Dur sous-jacent, puis nous nous attaquons aux deux problèmes mentionnés ci-dessus. En exploitant efficacement la géométrie du problème du bandit, nous proposons une stratégie d'échantillonnage aléatoire pour le problème BAI avec des garanties théoriques. En particulier, nous caractérisons l'influence de la structure du graphe (par exemple, étoile, complet ou cercle) sur le taux de convergence et proposons des expériences empiriques qui confirment cette dépendance. Pour le problème de la maximisation des récompenses cumulées, nous présentons le premier algorithme basé sur le regret pour les bandits bilinéaires graphiques utilisant le principe d'optimisme face à l'incertitude. L'analyse théorique de la méthode présentée borne l'α-regret par $\tilde{O}(\sqrt{T})$ et souligne l'impact de la structure du graphe sur le taux de convergence. Enfin, nous démontrons par diverses expériences la validité de nos approches.

MOTS CLÉS

Apprentissage séquentiel, Bandits Bilinéaires Graphiques, Multi-agents

ABSTRACT

We introduce a new model called *Graphical Bilinear Bandits* where a learner (or a central entity) allocates arms to nodes of a graph and observes for each edge a noisy bilinear reward representing the interaction between the two end nodes. In this thesis, we study the best arm identification problem and the maximization of cumulative rewards. For the first problem, a learner wants to find the graph allocation maximizing the sum of the bilinear rewards obtained through the graph. For the second problem, during the learning process, the learner has to make a trade-off between exploring the arms to gain accurate knowledge of the environment and exploiting the arms that appear to be the bests to obtain the highest reward. Regardless of the learner's goal, the graphical bilinear bandit model reveals an underlying NP-Hard combinatorial problem that precludes the use of any existing best arm identification (BAI) or regret-based algorithms. For this reason, we first propose an α-approximation algorithm for the underlying NP-hard problem, and then tackle the two problems mentioned above. By efficiently exploiting the geometry of the bandit problem, we propose a random sampling strategy for the BAI problem with theoretical guarantees. In particular, we characterize the influence of the graph structure (e.g., star, complete or circle) on the convergence rate and propose empirical experiments that confirm this dependence. For the problem of maximizing the cumulative rewards, we present the first regret-based algorithm for graphical bilinear bandits using the principle of optimism in the face of uncertainty. Theoretical analysis of the presented method gives an upper bound of $\tilde{O}(\sqrt{T})$on the α-regret and highlights the impact of the graph structure on the convergence rate. Finally, we demonstrate by various experiments the validity of our approaches.

KEYWORDS

Sequential learning, Graphical Bilinear Bandits, Multi-agents