École doctorale 072 : Sciences Pour l'Ingénieur

THÈSE de Doctorat

pour obtenir le grade de docteur délivré par

Université des Sciences et des Technologies de Lille

Spécialité doctorale "Informatique"

présentée et soutenue publiquement par

Xuedong SHANG

September, 2021

Méthodes Adaptatives pour l'Optimisation dans un Environnement Stochastique

Directeur de thèse : **Michal VALKO**
Co-directrice de thèse : **Emilie KAUFMANN**

Jury

M. Pierre ALQUIER,	Professeur, RIKEN AIP	Rapporteur
M. Aurélien GARIVIER,	Professeur, ENS Lyon	Examinateur
Mme Emilie KAUFMANN,	Chargée de recherche, CNRS, Inria	Co-directrice de thèse
M. Balázs KÉGL,	Directeur de recherche, CNRS, Huawei France	Examinateur
M. Daniel RUSSO,	Professeur assistant, Columbia Business School	Rapporteur
M. Michal VALKO,	Chargé de recherche, Inria, DeepMind Paris	Directeur de thèse

To my parents,

for always loving and supporting me.

Xuedong Shang

致永远爱我和我爱的父母。

商雪崇

Résumé

Dans cette thèse, nous étudions le problème d'optimisation séquentielle dans des environnements stochastiques. A chaque instant, nous pouvons interroger un point de l'environnement, et recevoir une récompense bruitée. Nous nous concentrons d'abord sur le cas où l'environnement est représenté par un nombre fini de points, et ensuite sur le cas plus général où l'environnement est composé d'un nombre infini dénombrable de points, voire continu. Dans les deux cas, le coût d'une requête pouvant être élevée, nous envisageons ainsi à repérer au plus vite le point (quasi)-optimal. Cette étude est motivée par de nombreux scénarios réels comme, entre autres, les essais cliniques, les tests A/B, ou l'optimisation des placements publicitaires. Ainsi pour terminer, nous nous intéressons en particulier à l'une de ces applications plus importantes pour la communauté d'apprentissage statistique, c'est-à-dire l'optimisation des hyper-paramètres.

Abstract

In this thesis, we study the problem of sequential optimization under stochastic environments. At each round, we can query a data point from the environment, and receive a noisy reward. We first focus on the case where the environment is abstracted as a finite search space, then we investigate also on a more general setting where the environment is composed of an infinite number of points or even continuous. In both cases, the cost of a single query would be high, and we thus aim at identify the (near)-optimum as efficiently as possible. The whole study is motivated by numerous real scenarios including, but not limited to, clinical trial, A/B testing, advertisement placement optimization. We therefore conclude by some particular focus on one of its most important contributions for the machine learning community, *i.e.* hyper-parameter optimization.

Résumé des travaux de thèse

0.1 Contexte de la thèse

0.1.1 Qu-étudions-nous et pourquoi ?

Imaginons que nous ayons accès à un simulateur qui modélise le comportement d'une tâche numérique complexe. Considéré comme une boîte noire, nous ne pouvons obtenir des informations utiles qu'en exécutant le simulateur avec différentes entrées. Par exemple, le processus d'inférence de la structure 3D d'une protéine à partir de sa séquence d'acides aminés peut être considéré comme une tâche complexe, qui peut être modélisée par un simulateur. Les entrées du simulateur sont les séquences d'acides aminés et les sorties sont les structures 3D prédites. Une famille populaire de méthodes cherche à optimiser une fonction énergétique appropriée - produite par le simulateur - qui décrit la relation entre la structure d'une protéine et sa séquence d'acides aminés. Ces méthodes sont intéressantes car elles sont capables de construire des structures de protéines sans connaissance préalable des structures résolues (voir par exemple Zhang 2008).

Dans le contexte de cette thèse, nous modélisons un tel scénario comme le problème dit de l'optimisation séquentielle, dans lequel un agent alimente séquentiellement un environnement (le simulateur dans l'exemple précédent) avec des entrées et reçoit un retour (déterministe ou stochastique) appelé gain, récompense ou observation. L'agent doit produire une estimation de l'entrée optimale après un certain nombre d'essais. Dans certaines circonstances, une seule interaction avec l'environnement peut être extrêmement coûteuse. Par exemple, dans l'exemple de la prédiction de la structure des protéines, de vastes ressources informatiques sont nécessaires si le simulateur reçoit une très grande protéine (avec de longues séquences d'acides aminés). Il est donc très intéressant de choisir soigneusement l'entrée à chaque pas de temps en fonction des observations passées afin de réduire le nombre de simulations.

L'optimisation séquentielle dans des environnements stochastiques est un sujet de recherche actif dans les communautés des mathématiques appliquées et de l'informatique. Par exemple, le problème de planification dans un processus de décision markovien, sur lequel la récente percée de l'intelligence de jeu du Go [Silver et al., 2016] est construite, est étroitement lié à l'optimisation séquentielle. Précisément, étant donné l'état actuel du jeu, l'intelligence de jeu est conçue pour maximiser une certaine fonction de valeur, dont les observations (bruitées) peuvent être obtenues en explorant des trajectoires bien choisies.

Un autre exemple, qui est aussi une motivation importante pour cette thèse, est celui de l'optimisation des hyper-paramètres des classifieurs d'apprentissage automatique. Les algorithmes modernes d'apprentissage automatique dépendent souvent de nombreux paramètres qui ne peuvent pas être appris par le processus d'apprentissage, mais qui doivent être spécifiés manuellement. Le réglage de ces hyper-paramètres est souvent considéré comme une partie fastidieuse d'une tâche d'apprentissage automatique. Il est

donc intéressant de concevoir des algorithmes qui automatisent le processus de choix de ces hyper-paramètres. L'optimisation des hyper-paramètres peut être considéré comme un problème d'optimisation boître noire où les évaluations de fonctions sont supposées être très coûteuses. En général, l'évaluation d'une fonction dans l'optimisation des hyper-paramètres implique l'exécution complète de l'algorithme principal d'apprentissage automatique sur un ensemble de données important et hautement dimensionnel, ce qui prend souvent beaucoup de temps ou de ressources.

En outre, l'optimisation séquentielle peut également servir d'abstraction pour de nombreux problèmes du monde réel. Pour n'en citer que quelques-uns, on peut penser aux problèmes de sélection de portefeuille (averse au risque) en finance [Ziemba and Vickson, 2010], à la conception de stratégies efficaces d'allocation de traitement en médecine [Durand et al., 2018], à la minimisation de l'énergie libre en génie chimique ou à la prédiction de la structure des protéines [Floudas and Pardalos, 2000], à la métamodélisation pour l'optimisation de la conception technique [Wang and Shan, 2007], à la estimation des paramètres (problème inverse) des voies biochimiques dynamiques non linéaires [Moles et al., 2003], à la distorsion du maillage en science des matériaux [Charpagne et al., 2019], et bien plus encore.

Pour résumer, un environnement peut simplement être considéré comme une fonction cible à optimiser. Cette fonction peut être **discrète ou continue**. Dans cette thèse, nous nous intéressons en particulier à l'optimisation de fonctions pour lesquelles **aucune (ou peu)** hypothèse de régularité est faite, et seules des évaluations de fonctions (interactions avec l'environnement) **bruitées (ou stochastiques)** peuvent être observées.

0.1.2 Comment traitons-nous le problème ?

L'outil principal que nous utilisons pour traiter le problème d'optimisation séquentielle dans cette thèse est un modèle statistique qui s'appelle le modèle de bandit à plusieurs bras. Ce modèle a été étudié pour la première fois par Thompson [1933], et peut être décrit de la manière suivante : On donne à un agent un ensemble *fini* de bras K et un horizon N. Tirer un bras conduit à une récompense stochastique qui suit une certaine distribution *inconnue* sous ce bras. À chaque pas de temps, l'agent peut choisir de tirer l'un des bras et observe une récompense échantillonnée à partir de sa distribution sous-jacente correspondante.

Dans son article de référence, Robbins [1952] définit l'objectif d'un agent de bandits comme la maximisation des récompenses totales à long terme. On observe que l'agent doit simultanément acquérir de nouvelles informations en vue d'un bien-être potentiel futur (exploration), et optimiser la décision actuelle basée sur les observations passées (exploitation). Ce phénomène est le fameux *dilemme de l'exploration et l'exploitation* et est présent dans de nombreuses tâches du monde réel. Le modèle de bandits est donc populaire parmi différentes communautés, car les algorithmes font un compromis entre l'exploration et l'exploitation.

Cependant, l'exploitation ne fournit pas nécessairement des incitations significatives dans certaines applications réelles. Typiquement, dans les exemples précédents présentés dans la section 0.1.1, nous ne nous soucions pas vraiment des pertes potentielles encourues pendant toute la phase d'apprentissage. En effet, nous cherchons uniquement à trouver rapidement le (quasi-)optimum de la fonction cible. Dans ce contexte, il est plus naturel d'évaluer l'agent dans une optique d'optimisation. Ce cadre, souvent appelé *identification du meilleurs bras*, est donc plus fortement lié à ce que nous allons étudier dans cette thèse. L'identification du meilleurs bras a été étudié en premier lieu par Even-dar

et al. [2003] et Bubeck et al. [2009] dans deux cadres différents que nous présentons en détail dans le chapitre 2.

Plus généralement, nous parlons de problèmes de *l'exploration pure* [Bubeck et al., 2011], où l'agent est censé gagner autant d'informations possibles sur le modèle de bandit indépendamment des récompenses. L'identification du meilleurs bras est simplement une instance particulière de l'exploration pure, pour laquelle l'objectif d'apprentissage est de trouver le bras optimal. D'autres objectifs d'apprentissage existent également, comme trouver des bras qui dépassent un certain seuil prédéfini (voir par exemple Locatelli et al. 2016). Cependant, nous nous concentrons principalement sur l'identification du meilleurs bras dans cette thèse car il a un large éventail d'applications.

0.1.3 Du bandit manchot à l'apprentissage par renforcement

Le modèle de bandits est populaire d'un autre point de vue car certains problèmes de bandits font partie d'un cadre plus général qui est l'apprentissage par renforcement (RL). Dans un modèle du RL, l'environnement est caractérisé par son état actuel et l'agent interagit avec l'environnement en prenant différentes actions. Chaque action conduit à une récompense de la part de l'environnement ainsi qu'à un changement d'état. La définition formelle et les résultats généraux du RL dépassent le cadre de cette thèse, les lecteurs peuvent se référer à Bertsekas [2011]; Sutton and Barto [1998] pour les études.

L'apprentissage par renforcement moderne combiné avec l'apprentissage profond a conduit à des avancées passionnantes, notamment AlphaGo [Silver et al., 2016], AlphaStar [Vinyals et al., 2019], etc. Cependant, il existe encore une grande lacune dans la compréhension de l'énorme succès de RL profond. Bandit manchot, en tant que modèle statistique fortement fondé par la théorie, peut potentiellement servir de première étape pour combler cette lacune dans la recherche sur le RL profond. Plus précisément, bandit manchot est parfois considéré comme la forme la plus simple de RL car les agents de bandits ne subissent aucun changement d'état (voir Fig. 1). Nous discuterons un peu plus du lien entre bandits et RL plus tard dans la conclusion générale 7 car les bandits contextuelles (bandits avec informations secondaires) – une variante du modèle classique – décrit finalement mieux la façon dont le bandit manchot est lié au RL.

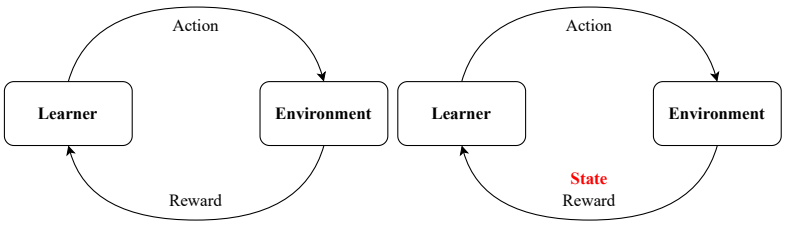


Figure 1: Gauche: cycle d'apprentissage bandit vs. Droite: cycle d'apprentissage RL.

0.2 Bandit manchot multi-bras et optimisation

Cette thèse traite des problèmes d'optimisation séquentielle dans des environnements stochastiques en utilisant des outils de bandits. Un environnement stochastique se réfère à un environnement à partir duquel des retours stochastiques sont acquis lorsqu'une entrée est demandée depuis l'espace de recherche/action[1] $\mathscr{X}$. Formellement, et sans perte

[1] Ces termes peuvent être employés de manière interchangeable.

de généralité, notre objectif est de maximiser une fonction cible $f : \mathscr{X} \to \mathbb{R}$, c'est-à-dire de trouver

$$\arg\max_{x \in \mathscr{X}} f(x) \tag{1}$$

en fonction d'une séquence de valeurs de la fonction f. Évidemment, sans aucune information préalable sur la fonction cible et/ou l'espace de recherche $\mathscr{X}$, il s'agit d'une mission impossible. Cette thèse étudie plusieurs instances particulières de (1) avec différents espaces de recherche et/ou différentes hypothèses (de régularité) sur la fonction cible, et apporte de nouvelles perspectives théoriques et pratiques.

Dans le reste de ce chapitre, je donne un aperçu informel des différents contextes étudiés dans cette thèse ainsi qu'un résumé de mes contributions à chaque contexte.

Une discussion plus approfondie sur la formulation du problème, en particulier sur la manière d'évaluer la performance des algorithmes dans différents contextes, est donnée dans le chapitre 2, qui constitue une introduction au modèle de bandit. L'objectif de ce chapitre introductif est de présenter le problème de bandits de manière plus formelle. Nous rappelons d'abord quelques notions de base ainsi que certains résultats fondamentaux pour les bandits stochastiques. Nous nous concentrons ensuite sur la façon dont différents cadres d'identification du meilleur bras sont formulés.

0.2.1 Identification du meilleurs bras dans un modèle de bandit stochastique

Le premier cadre d'intérêt consiste en un espace de recherche fini et unidimensionnel $\mathscr{X} = \{x_1, x_2, \cdots, x_K\}$. Supposons que la distribution de récompense sous-jacente du bras x_k soit caractérisée par sa moyenne $\mu_k \in \mathbb{R}$, la fonction cible f peut être simplement interprétée comme une correspondance entre chaque bras et sa moyenne. L'agent cherche alors à trouver

$$\arg\max_{k \in [K]} \mu_k \tag{2}$$

étant donné une certaine condition d'arrêt. Il s'agit de l'identification du meilleur bras pour les bandits multi-bras stochastiques. Il existe plusieurs objectifs d'apprentissage pour ce genre de problèmes, parmi lesquels nous sommes particulièrement intéressés par le cas où le but est d'identifier le meilleur bras avec une confiance élevée avec un minimum d'évaluations de fonctions. Il s'agit du *cadre dit de confiance fixée* pour lequel la définition formelle, ainsi que celles pour d'autres objectifs d'apprentissage, sont fournies et discutées plus loin dans le chapitre 2.

Les méthodes existentes de ce problème nécessitent la construction d'intervalles de confiance compliqués sur les récompenses moyennes. Dans cette thèse, nous profitons des outils bayésiens pour résoudre ce problème, qui est basée sur le célèbre Thompson sampling (Thompson 1933, voir aussi Russo et al. 2018 pour un tutoriel). Thompson sampling est un algorithme bayésien bien connu pour l'objectif classique de maximisation de la récompense, pour lequel il est maintenant considéré comme un concurrent majeur des approches populaires de type `UCB` [Auer et al., 2002a]. Une question naturelle à se poser est de savoir si les méthodes bayésiennes peuvent également être un bon concurrent des approches classiques de l'identification du meilleurs bras basées sur des intervalles de confiance. Cependant, il est bien connu que l'utilisation directe de Thompson sampling ne permet pas d'obtenir une performance optimale pour l'identification du meilleurs tant d'un point de vue pratique que théorique. Plus précisément, elle ne peut pas atteindre une *complexité d'échantillonage* asymptotique qui correspond à une borne inférieure

fournie par Garivier and Kaufmann [2016]. Une telle propriété est appelée *optimalité asymptotique* dont nous donnerons une définition formelle plus tard dans le chapitre 2. Une adaptation telle que TTTS proposée par Russo [2016] est nécessaire : en choisissant entre deux bras candidats différents à chaque tour, elle impose l'exploration de bras sous-optimaux, qui seraient sous-échantillonnés par Thompson sampling original.

Dans le chapitre 3, nous proposons une nouvelle étude de TTTS, et fournissons de nouvelles compréhensions théoriques sur sa complexité d'échantillonnage. Plus précisément, nous montrons que TTTS atteint l'optimalité asymptotique qui répond alors à une question ouverte de Russo [2016]. Nous proposons en outre une amélioration computationnelle T3C de TTTS, tout en gardant les mêmes garanties. De plus, nous fournissons également de nouveaux résultats sur la convergence de la loi a posteriori de TTTS.

0.2.2 Extension à l'identification du meilleurs bras dans un modèle de bandit linéaire

Une extension du cadre précédent largement étudiée consiste à prendre un ensemble fini de K bras/contextes $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K\} \subset \mathbb{R}^d$ comme espace de recherche. La récompense de chaque bras dans cette circonstance est supposée linéairement dépendante d'un *paramètre de régression* $\boldsymbol{\theta} \in \mathbb{R}^d$. La fonction cible f peut donc être considérée comme une correspondance entre chaque bras $\mathbf{x}$ et sa combinaison linéaire avec $\boldsymbol{\theta}$, et est donc appelée bandits linéaires. Précisément, l'agent cherche à trouver

$$\arg\max_{k \in [K]} \boldsymbol{\theta}^\top \mathbf{x}_k . \tag{3}$$

Le paramètre $\boldsymbol{\theta}$ est bien sûr *inconnu* de l'agent. Ce paramètre contextuel (linéaire) décrit mieux certains scénarios du monde réel. Un exemple typique est l'optimisation du placement des publicités, dans lequel un site Web cherche à identifier le modèle d'affichage publicitaire le plus performant. Dans de telles applications, les caractéristiques des utilisateurs peuvent être utilisées comme informations secondaires (contexte) pour aider à la conception de l'exploration (voir par exemple Li et al. 2010).

Une fois de plus, comme pour l'identification du meilleurs bras pour les bandits stochastiques, nous nous sommes intéressés au cadre de confiance fixée. Les algorithmes précédents sur ce sujet n'atteignent qu'une faible borne de complexité d'échantillon qui est liée à la *G-optimalité* de la théorie du plan d'expérience (voir par exemple Pukelsheim 2006). Nous conjecturons que la G-optimalité ne décrit pas au mieux la complexité de l'identification du meilleurs bras pour les bandits linaires, et essayons donc d'adapter d'autres complexités plus appropriées.

Une ligne de recherche naturelle est alors de concevoir un algorithme asymptotiquement optimal. Une adaptation simple de `Track-and-Stop` [Garivier and Kaufmann, 2016] au cadre linéaire s'avère asymptotiquement optimale [Jedra and Proutière, 2020], mais reste défavorable sur le plan des resources computationnelles. Nous cherchons donc également à concevoir des algorithmes légers en complexité temporelle.

Dans le chapitre 4, nous proposons une nouvelle complexité pour l'identification du meilleurs bras linéaire, et fournissons une comparaison complète des complexités existantes. Nous étudions ensuite à la fois les approches bayésiennes et les algorithmes basés sur les intervalles de confiance. En particulier, nous proposons plusieurs extensions différentes de TTTS et T3C au cadre linéaire. Malheureusement, nous montrons empiriquement qu'elles ne sont pas asymptotiquement optimales. Dans le même temps, nous développons une approche utilisant le point de selle qui conduit à un algorithme optimal `LinGame`.

0.2.3 Bandits infinis et optimisation boîte noire

Enfin, un problème plus général consiste à considérer un espace infini ou continu $\mathcal{X}$, et chaque bras $x \in \mathcal{X}$ obtient sa récompense moyenne $f(x)$ par la fonction de récompense f. On retrouve donc (1) :

$$\arg\max_{x \in \mathcal{X}} f(x) .$$

Il s'agit du problème de l'optimisation globale ou optimisation boîte noire. Parfois, nous pouvons également parler de l'optimisation d'ordre zéro, par opposition à l'*optimisation du premier ordre* pour lequel des informations basées sur le gradient sont disponibles.

Nous étudions le cas stochastique dans cette thèse. Les approches typiques pour traiter l'optimisation globale incluent l'optimisation bayésienne (voir par exemple Brochu et al. 2010), les algorithmes évolutionnaires et les bandits hiérarchiques (voir par exemple Bubeck et al. 2010). Dans cette thèse, nous nous concentrons sur les algorithmes de bandits hiérarchiques. Dans la littérature, nous faisons souvent référence aux *bandits à bras infinis* ou bien aux *bandits à bras continus*.

De toute évidence, on ne peut s'attendre qu'à une solution *quasi-optimale* dans le cas des bandits à bras continus. Nous utilisons donc une autre mesure de performance, à savoir le *regret simple*, qui est la différence entre la valeur de la fonction optimale réelle et la valeur de la fonction de notre estimation finale. La définition formelle est fournie plus loin dans le chapitre 2. Le regret simple diffère du *regret cumulé* qui sert de mesure de performance pour la maximisation de la récompense.

Dans le chapitre 5, nous explorons la possibilité de concevoir des algorithmes de bandit hiérarchique sans paramètre avec un minimum d'hypothèses. À cette fin, nous utilisons un shéma de validation croisée et construisons un algorithme appelé GPO. GPO est un méta-algorithme qui peut utiliser n'importe quel algorithme de bandit hiérarchique comme sous-routine. En particulier, GPO atteint presque la même garantie de regret simple que sa sous-routine. Comme résultat secondaire, nous montrons également que HCT est un algorithme sous-jacent valide pour GPO ainsi que pour POO proposé par Grill et al. [2015].

0.2.4 Optimisation des hyper-paramètres

Enfin, nous abordons une question plus pratique : l'optimisation des hyper-paramètres. Comme présenté dans la section 0.1.1, le réglage efficace des hyper-paramètres pourrait être d'une grande importance pour les praticiens de l'apprentissage automatique. Comme indiqué, l'optimisation des hyper-paramètres peut être naturellement modélisée comme un problème d'optimisation séquentielle. L'espace de recherche dans ce cadre peut être à la fois discret (variables catégoriques, variables à valeur entière, etc.) et continu (variables à valeur réelle).

Inspirés par un algorithme récent Hyperband basé sur l'identification du meilleurs bras [Li et al., 2017], nous cherchons à proposer d'autres algorithmes pour l'optimisation des hyper-paramètres aussi basés là-dessus. En effet, Hyperband est construit sur un algorithme basé sur l'élimination et a de bonnes performances par rapport aux méthodes précédentes. D'autre part, nous nous intéressons à la possibilité d'adapter des algorithmes bayésiens tels que TTTS pour résoudre l'optimisation des hyper-paramètres. Notez que TTTS n'est conçu que pour les *bandits à bras finis*, un contournement appropriée est donc nécessaire.

Dans le chapitre 6, nous concevons un algorithme robuste et dynamique D-TTTS basé sur TTTS, et montrons que de tels algorithmes à saveur bayésienne peuvent être de bons

candidats pour des applications comme l'optimisation des hyper-paramètres. Nous discutons également d'un inconvénient majeur de D-TTTS, et proposons une solution dans le même chapitre.

Acknowledgement

Contents

0.1 Contexte de la thèse v
0.2 Bandit manchot multi-bras et optimisation vii

Contents **xv**

List of Algorithms **xvii**

List of Figures **xix**

List of Tables **xxi**

1 Introduction **1**
1.1 Context of the Thesis 2
1.2 Multi-Armed Bandits and Optimization 5
1.3 Publications of the PhD 8

2 Stochastic Multi-Armed Bandits **9**
2.1 The Multi-Armed Bandits Model 10
2.2 Best-Arm Identification 16
2.3 Extensions of Best-Arm Identification 20
2.4 Many-armed bandits 22
2.5 Performance Measure 23

3 A Bayesian Study of Best-Arm Identification **27**
3.1 Introduction 28
3.2 Bayesian BAI Strategies 29
3.3 Two Related Optimality Notions 33
3.4 Fixed-Confidence Analysis 35
3.5 Optimal Posterior Convergence 40
3.6 Numerical Illustrations 40
3.7 Discussion 43

4 Optimal Algorithms for Linear Best-Arm Identification **45**
4.1 Introduction 46
4.2 Problem Setting and Assumptions 47
4.3 Fixed-Confidence Optimality and Complexities 49
4.4 Related Work 55
4.5 Bayesian Algorithms for the Linear Case 56
4.6 A Gamified Algorithm 62
4.7 Other Saddle-Point Approaches 68
4.8 Discussion 70

5 Hierarchical Bandits for Black-Box Optimization **71**
5.1 Introduction . . . 72
5.2 Required Assumptions . . . 73
5.3 General Parallel Optimization . . . 75
5.4 HCT under Local Smoothness w.r.t. $\mathscr{P}$ . . . 78
5.5 Experimental Illustrations . . . 83
5.6 Discussion . . . 85

6 Bandits and Hyper-Parameter Optimization **87**
6.1 Introduction . . . 88
6.2 A Brief Survey of Automated Machine Learning . . . 89
6.3 Hyper-Parameter Optimization Framework . . . 91
6.4 Best-Arm Identification for Hyper-Parameter Tuning . . . 92
6.5 Active TTTS for Hyper-Parameter Optimization . . . 93
6.6 Experiments . . . 97
6.7 Adaptivity to $\mu^\star$ . . . 100
6.8 Discussion . . . 104

7 General Conclusion and Perspectives **105**
7.1 General Discussion . . . 106
7.2 Future Perspectives . . . 106

A Mathematical Tools **123**
A.1 Some Reminders on Probability . . . 123
A.2 Concentration Inequalities . . . 124
A.3 Information Theory . . . 126

B Additional Proofs of Chapter 3 **129**
B.1 Notation . . . 129
B.2 Technical Lemmas . . . 129
B.3 Fixed-Confidence Analysis for TTTS . . . 131
B.4 Fixed-Confidence Analysis for T3C . . . 144
B.5 Proof of Lemma 3.1 . . . 149
B.6 Proof of Posterior Convergence for Gaussian Bandits . . . 151
B.7 Proof of Posterior Convergence for Bernoulli Bandits . . . 155

C Additional Proofs of Chapter 4 **165**
C.1 Notation . . . 165
C.2 Technical Lemmas . . . 165
C.3 Sample Complexity of LinGame . . . 168
C.4 A Fair Comparison of Stopping Rules . . . 175

D Additional Proofs of Chapter 5 **179**
D.1 Notation . . . 179
D.2 Detailed regret analysis for HCT under Assumption 5.2 . . . 180
D.3 General analysis of POO . . . 188

E Acronyms **191**

F Glossary **193**

List of Algorithms

2.1 Algorithm of UCB 16
3.1 Sampling rule of TTTS 31
3.2 Sampling rule of T3C 31
4.1 Frank-Wolfe heuristic for computing $\mathcal{X}\mathcal{X}$-design 53
4.2 Saddle Frank-Wolfe heuristic for computing generic $\mathcal{X}\mathcal{Y}$-design 54
4.3 Sampling rule of L-T3S/L-T3C 57
4.4 Sampling rule of L-T3C-Greedy 59
4.5 Algorithm of LinGame 63
4.6 Algorithm of SLinGapE 69
4.7 Sampling rule of SL-T3C 69
5.1 Algorithm of POO($\mathcal{A}$) with base algorithm $\mathcal{A}$ 76
5.2 Algorithm of GPO 77
5.3 Algorithm of HCT 79
5.4 Snippet OptTraverse of HCT 79
5.5 Snippet UpdateBackward of HCT 80
6.1 Sampling rule of Dynamic D-TTTS 96
6.2 Sampling rule of H-TTTS 97

List of Figures

1 Gauche: cycle d'apprentissage bandit vs. Droite: cycle d'apprentissage RL. . vii

1.1 Left: a bandit learning cycle vs. Right: a reinforcement learning cycle. 5

2.1 An example of modelling clinical trials as a MAB problem. 10
2.2 A bandit learning cycle. 11

3.1 Dots: empirical sample complexity; Solid line: theoretical sample complexity. 42
3.2 Sample complexity of different BAI sampling rules over some random problem instances. Black dots represent means and oranges lines represent medians. 43

4.1 A hard problem instance for linear best-arm identification. 58
4.2 T3C-Greedy vs. Track-and-Stop for different value of δ. 60
4.3 Comparison of the sample complexity of L-T3C-Greedy and LinGapE on the pathological instance in $\mathbb{R}^d$ for different values of d. 61
4.4 Sample complexity of different linear BAI sampling rules over the usual counter-example with $\delta = 0.1, 0.01, 0.0001$ respectively. CG = LinGame-C, Lk = LinGame, RR = uniform sampling, fix = tracking the fixed weights, GS = $\mathcal{XY}$-Static with $\mathcal{XX}$-allocation, XYS = $\mathcal{XY}$-Static with $\mathcal{XY}_{\text{dir}}$-allocation, LG = LinGapE. The mean stopping time is represented by a black cross. 67
4.5 Sample complexity of different linear BAI sampling rules over random unit sphere vectors with $d = 6, 8, 10, 12$ from left to right. 68
4.6 Average stopping time (Left: $d = 2, \alpha = \pi/4, \delta = 0.0000001$, Right: $d = 2, \alpha = 0.1, \delta = 0.1$), with SFW-T = SL-T3C, SFW-L = SLinGapE, T-G = L-T3C, LG = LinGapE. 70

5.1 Benchmark functions for testing black-box optimization algorithms. 84
5.2 Simple regret of POO and PCT run for different ρ values. 85

6.1 The full-stack pipeline of a machine learning task. 90
6.2 Simple regret as a function of allocation budget for various BAI algorithms. 95
6.3 Simple regret of D-TTTS (against Hyperband) as a function of the number of arms evaluations for different Beta reservoir. 98
6.4 Mean cross-validation error of different HPO algorithms with (a) SVM run on the UCI wine dataset, (b) SVM run on the UCI breast cancer dataset, (c) SVM run on the UCI adult dataset and (d) MLP run on the MNIST dataset. 99
6.5 Simple regret of D-TTTS (against Hyperband) for shifted Beta reservoir. . . . 100
6.6 Illustration of over-exploration of D-TTTS under shifted reservoirs. 101
6.7 Simple regret of D-TTTS for shifted Beta reservoir after the fix. 103
6.8 Distribution of effectively sampled arms of D-TTTS before and after the fix. . 104

List of Tables

3.1 Average execution time in seconds for different BAI sampling rules. 43

4.1 Optimal weights for various complexities with $\Delta_{\min} = 0.0049958$. 53
4.2 Some examples of different transductive sets. 53
4.3 Comparison between the two algorithms. 59
4.4 Average number of pulls of each arm ($d = 2, \delta = 0.1$). 60
4.5 Average number of pulls of each arm ($d = 2, \delta = 0.1$). 60
4.6 average number of pulls of each arm ($d = 2, \delta = 0.1$). 61
4.7 Average number of pulls of `LinGame` and `LinGame-C` (against `DKM`) for each arm. 67

5.1 Smoothness assumptions for hierarchical bandits algorithms. 75

6.1 Casting HPO as a BAI problem. 93

B.1 Table of notation for Chapter 3 129

C.1 Table of notation for Chapter 4. 165
C.2 Stopping rules for different linear BAI algorithms. 175

D.1 Table of notation for Chapter 5. 179

Chapter 1

Introduction

" *The thing about quotes on the internet is that you can not confirm their validity.*

Abraham Lincoln

Contents

1.1 Context of the Thesis 2

1.1.1 What do we study and why? 2

1.1.2 How do we approach the problem? 3

1.1.3 From multi-armed bandits to reinforcement learning 4

1.2 Multi-Armed Bandits and Optimization 5

1.2.1 Best-arm identification for stochastic multi-armed bandits 5

1.2.2 Extension to best-arm identification for linear bandits 6

1.2.3 Infinitely-armed bandits and black-box optimization 7

1.2.4 Hyper-parameter optimization 7

1.3 Publications of the PhD 8

The purpose of this thesis is to provide a summary of the main research line of my PhD work carried out in between October 2017 and March 2021. During my PhD, I was hosted at the Inria Lille-Nord Europe (France) research center, in the SequeL team (now becomes Scool team). I was fortunate to be advised by Dr. Michal Valko, and also co-supervised by Dr. Emilie Kaufmann. The research thematic of SequeL lies in sequential decision making problems, to which all my contributions are devoted. In particular, this document mainly investigates sequential decision making in optimization problems.

Timeline of the thesis. In the very beginning, the research was motivated by designing efficient hyper-parameter tuning algorithms (Chapter 6). The starting point was to compare hierarchical bandits approaches (Chapter 5) with traditional Bayesian optimization algorithms. I could observe, however, that hierarchical bandits algorithms often suffer from the curse of dimensionality, and also do not necessarily show superior performance to their Bayesian competitors.

At the same time, a great work on the topic, namely `Hyperband` [Li et al., 2017], was published. Based on a simple best-arm identification (see Chapter 2 for a formal definition) algorithm `Sequential-Halving` [Bubeck et al., 2009], `Hyperband` achieved both decent practical performances and nice theoretical guarantees. I thus turned my attention to the study of best-arm identification in the purpose of exploring further the potential of best-arm identification for hyper-parameter tuning. This effort then led to a dynamic algorithm `D-TTTS` for hyper-parameter optimization (Chapter 6). `D-TTTS` is constructed upon a Bayesian best-arm identification algorithm `TTTS` proposed by Russo [2016]. In the hope of providing an analysis of `D-TTTS`, which turns out to be sophisticated, I took a step back to revisit `TTTS` and managed to bring new theoretical insights about Bayesian best-arm identification (Chapter 3). Although the new insights still didn't shed light on the analysis of `D-TTTS`, they opened new way to other variants of best-arm identification. In particular, I then studied best-arm identification with linear payoffs (Chapter 4).

The structure of this thesis does not necessarily follow the previous timeline. Indeed, the present manuscript rather follows a scientific logic, in the sense that we go through from the simplest setting to more complicated ones.

In this first chapter, I will introduce the different problem settings studied in this thesis from a high-level perspective. I will focus on motivating the settings and also summarizing my contributions to each of them.

1.1 Context of the Thesis

1.1.1 What do we study and why?

Imagine that we have access to a simulator that models the behaviour of some complex numerical task. Being considered as a black box, we can only get useful information by running the simulator with different inputs. For example, the process of inferring the 3D structure of a protein from its amino-acid sequence can be regarded as such a complex task, that can be modelled by a simulator. The inputs of the simulator are the amino-acid sequences and the outputs are the predicted 3D structures. A popular family of methods seek to optimize a suitable energy function – produced by the simulator – that describes the relation between the structure of a protein and its amino-acid sequence. These methods are of interest because they are able to build protein structures without prior knowledge on solved structures (see e.g. Zhang 2008).

In the context of this thesis, we model such a scenario as the so-called *sequential optimization*[1] problem where a learner sequentially feeds inputs to an environment (the simulator in the previous example) and from which they receive (deterministic or stochastic) feedback/payoffs/rewards/observations[2]. The learner needs to output a guess for the optimal input after a number of trials. Under some circumstances, a single interaction with the environment could be extremely costly. For instance, in the example of protein structure prediction, vast computational resources are required if the simulator is given a very large protein (with long amino-acid sequences). It is therefore of great interest to carefully choose the input at each time step based on past observations to reduce the number of simulations.

Sequential optimization in a stochastic environment is an active research topic in both applied mathematics and computer science communities. For example, the planning problem in a *Markov decision process* (MDP), upon which the recent breakthrough of game intelligence of Go [Silver et al., 2016] is constructed, is closely related to sequential optimization. Precisely, given the current state of the game, the game intelligence is designed to maximize a certain value function, whose (noisy) observations can be obtained by exploring well-chosen trajectories.

Another example, which is also one important driving force that motivates this thesis, is *hyper-parameter optimization* (HPO) of machine learning classifiers. Modern machine learning algorithms often contain many parameters that cannot be learned through the learning process, but instead, need to be manually specified. Tuning those so-called *hyper-parameters* is often considered as a tedious part in a machine leanrning task. It is hence appealing to design HPO algorithms that automates the process of choosing those hyper-parameters. HPO can be viewed as a *black-box optimization* (BBO) problem where function evaluations are supposed to be very expensive. Typically, a function evaluation in HPO involves running the primary machine learning algorithm to completion on a large and high-dimensional dataset, which often takes a considerable amount of time or resources.

Besides, sequential optimization can also serve as an abstraction of numerous real-world problems. To name a few of them, we can think of (risk-averse) portfolio selection problems in finance [Ziemba and Vickson, 2010], designing effective treatment allocation strategies in medicine [Durand et al., 2018], free-energy minimization in chemical engineering or protein structure prediction [Floudas and Pardalos, 2000], metamodelling for engineering design optimization [Wang and Shan, 2007], parameter estimation (inverse problem) of nonlinear dynamic biochemical pathways [Moles et al., 2003], mesh distortion in material science [Charpagne et al., 2019], and way more.

To summarize, an environment can simply be regarded as a target function to be optimized. This function can be **discrete or continuous**. In this thesis, we are in particular interested in optimization of functions for which **none (or few)** regularity assumptions are made, and only **noisy (or stochastic)** function evaluations (interactions with the environment) can be observed.

1.1.2 How do we approach the problem?

The main tool that we use to address the sequential optimization problem in this thesis is *multi-armed bandits* (MAB). The original MAB problem is first studied by Thompson [1933], and can be described in the following way: A learner is given a *finite* set of K arms

[1] We thus do not consider parallelization in this thesis.
[2] Those terms can be interchangeably employed.

and a time horizon N. Pulling an arm leads to a stochastic reward that follows some *unknown* distribution underpin that arm. At each time step, the learner can choose to pull one of the arms and observes a reward sampled from its corresponding underlying distribution.

In his seminal work, Robbins [1952] defines the objective of a MAB learner as maximizing the total rewards in the long run. An observation is that the learner is required to simultaneously acquire new information for potential future well-being (exploration), and optimize the current decision based on past observations (exploitation). Such phenomena is stated as the *exploration-exploitation dilemma*, and is present in many real-world tasks. The MAB model is thus popular among different communities as MAB algorithms trade-off between exploration and exploitation.

However, exploitation does not necessarily provide meaningful incentives in some real applications. Typically, in the previous working examples presented in Section 1.1.1, we do not really care about the potential losses incurred during the whole learning phase. Indeed, we only aim at finding the (near-)optimum of the target function quickly. In this context, it is more natural to assess the learner in an optimization fashion. This setting, often named as *best-arm identification* (BAI), is thus more closely related to what we are to investigate in this thesis. BAI has been firstly studied by Even-dar et al. [2003] and Bubeck et al. [2009] in two different frameworks that we introduce in detail in Chapter 2.

More generally, we talk about *pure exploration* problems [Bubeck et al., 2011] instead of BAI, where the learner is supposed to gain as much information about the bandit model regardless of rewards. BAI is merely a particular instance of pure exploration, for which the learning objective is to find the optimal arm. Other learning objectives also exist such as finding arms that surpass some pre-defined threshold (see e.g. Locatelli et al. 2016). However, we mostly focus on BAI in this thesis since it has a wide range of applications.

1.1.3 From multi-armed bandits to reinforcement learning

The MAB model is popular from another perspective as some bandit problems are part of the more general *reinforcement learning* (RL) framework. In RL, the environment is characterized by its current state and the learner interacts with the environment by taking different actions. Each action leads to a reward from the environment as well as a change of states. Formal definition and general results of RL are beyond the scope of this thesis, readers can refer to Bertsekas [2011]; Sutton and Barto [1998] for surveys.

Modern RL combined with *deep learning* (DL) has marked a handful of exciting breakthroughs including AlphaGo [Silver et al., 2016], AlphaStar [Vinyals et al., 2019], etc. However, a big gap still exists in understanding the huge success of Deep RL. MAB, as a strongly theoretically-grounded statistical model, can potentially serve as a first step towards filling that gap in Deep RL research. More precisely, MAB is sometimes considered as the simplest form of RL as MAB learners do not incur any change of states (see Fig. 1.1). We shall discuss a little more the link between MAB and RL later in the general conclusion Chapter 7 as contextual MAB (MAB with side information) – a variant of the vanilla MAB model – describes eventually better how MAB is related to RL.

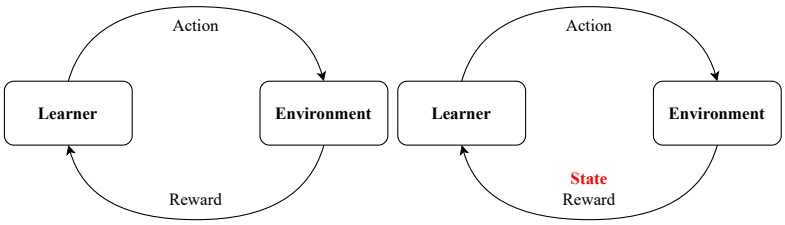


Figure 1.1: Left: a bandit learning cycle vs. Right: a reinforcement learning cycle.

1.2 Multi-Armed Bandits and Optimization

This thesis addresses sequential optimization problems under stochastic environments from a bandit point of view. A stochastic environment refers to an environment from which stochastic feedback are acquired when an input is queried from the search/action space[3] $\mathcal{X}$. Formally, and without loss of generality, we aim to maximize[4] a target function $f : \mathcal{X} \to \mathbb{R}$, i.e. find

$$\arg\max_{x \in \mathcal{X}} f(x) \tag{1.1}$$

based on a sequence of function values of f. Obviously without any prior information on the target function and/or the search space $\mathcal{X}$, it is just a find-a-needle-in-a-haystack mission. This thesis studies several particular instances of (1.1) with various search spaces and/or different (regularity) assumptions on the target function, and brings both novel theoretical and practical insights.

In the rest of this section, I provide a high-level overview of different settings investigated in this thesis along with a summary of my contributions to each setting.

More thorough discussion on the problem formulation, in particular how do we assess the performance of algorithms under different settings is given in an introductory Chapter 2 about the MAB model. The purpose of this introductory chapter is to present the MAB problem in a more formal way. We first recall some basic notions as well as some fundamental results for stochastic MAB. We then focus on how different best-arm identification/global optimization settings are formulated from a bandit point of view.

1.2.1 Best-arm identification for stochastic multi-armed bandits

The first setting of interest consists of a finite and one-dimensional search space $\mathcal{X} = \{x_1, x_2, \cdots, x_K\}$. Assume that the underlying reward distribution of arm x_k is characterized by its mean $\mu_k \in \mathbb{R}$, the target function f can be simply interpreted as a mapping form each arm to its mean. The learner then aims to find

$$\arg\max_{k \in [K]} \mu_k \tag{1.2}$$

given some stopping condition. This is the best-arm identification for stochastic multi-armed bandits. Several learning objectives exist for a BAI problem, among which we are in particular interested in the setting of which the goal is to identify the best arm with high confidence based on a minimum of function evaluations. This is the *fixed-confidence setting* for whom the formal definition, along with those for other learning objectives, are provided and discussed later in Chapter 2.

[3] Those terms can be interchangeably employed.
[4] Minimization is obviously the same problem.

Existing methods for BAI for stochastic MAB often requires construction of complicated confidence intervals of the mean estimates. We opt for Bayesian machinery to address this problem in this thesis, which is based on the famous Thompson Sampling (TS) (Thompson 1933, see also Russo et al. 2018 for a tutorial). TS is a Bayesian algorithm well known for the classical reward maximization objective, for which it is now seen as a major competitor to the popular UCB-typed approaches [Auer et al., 2002a]. A natural question to ask is whether Bayesian methods can be also a good competitor to classical BAI approaches constructed upon confidence intervals. However, it is well known that direct use of TS cannot yield optimal performance for BAI both in a practical and theoretical point of view. More precisely, it cannot achieve an asymptotic *sample complexity* that matches the lower bound provided by Garivier and Kaufmann [2016]. Such property is called *asymptotic optimality* that we shall give formal definition later in Chapter 2. An adaptation such as TTTS proposed by Russo [2016] is needed: by choosing between two different candidate arms in each round, it enforces the exploration of sub-optimal arms, which would be under-sampled by vanilla TS due to its objective of maximizing rewards.

In Chapter 3, we revisit TTTS, and provide new theoretical understandings on its sample complexity. More precisely, we show that TTTS achieves asymptotic optimality on the sample complexity which then answers to one open question of Russo [2016]. We further propose a computational improvement T3C of TTTS, whilst keeping the same guarantees. Besides, we also provide some new results on the posterior convergence of TTTS.

1.2.2 Extension to best-arm identification for linear bandits

Beyond the previous vanilla setting of BAI, a widely studied extension is to take a finite set of K arms/contexts/features[5] $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K\} \subset \mathbb{R}^d$ as the search space. The reward of each arm under this circumstance is assumed to depend linearly on a *regression parameter* $\boldsymbol{\theta} \in \mathbb{R}^d$. The target function f therefore can be considered as a mapping from each arm $\mathbf{x}$ to its linear combination with $\boldsymbol{\theta}$, and is thus called linear bandits. Precisely, the learner seeks to find

$$\arg\max_{k \in [K]} \boldsymbol{\theta}^\top \mathbf{x}_k \,. \tag{1.3}$$

The parameter $\boldsymbol{\theta}$ is of course *unknown* to the learner. This (linear) contextual setting describes better some real-world scenarios. A typical example is advertising placement optimization in which a website seeks to identify the best-performing ad display design. In such applications, user features can be used as side information (context) to help the design of exploration (see e.g. Li et al. 2010).

Again, like for vanilla BAI, we are interested in the fixed-confidence setting. Previous algorithms on this topic only achieve loose sample complexity bound which is linked to the *G-optimality* of experimental design theory (see e.g. Pukelsheim 2006). We conjecture that G-optimality does not describe best the complexity of linear bandits BAI, and therefore try to opt for other more appropriate complexities.

One natural research line is then to design asymptotically optimal algorithm. A simple adaptation of Track-and-Stop [Garivier and Kaufmann, 2016] to the linear setting is shown to be asymptotically optimal [Jedra and Proutière, 2020], but remains computationally unfavorable. We thus also seek to design computational friendly algorithms.

In Chapter 4, we propose a new complexity notion for linear bandits BAI, and provide a comprehensive comparison of existing complexities. We then investigate both Bayesian

[5] Those terms can be interchangeably employed in the context of this thesis.

approaches and confidence interval-based algorithms. In particular, we propose several different extensions of TTTS and T3C to the linear setting. Unfortunately, we show empirically that they are not asymptotically optimal. In the meantime, we develop a saddle-point approach that leads to an optimal algorithm LinGame.

1.2.3 Infinitely-armed bandits and black-box optimization

Finally, a more general problem is to consider an infinite or continuous space $\mathscr{X}$, and each arm $x \in \mathscr{X}$ gets its mean reward $f(x)$ through the reward function f. We thus recover (1.1):

$$\arg\max_{x \in \mathscr{X}} f(x) .$$

This is the *global optimization* (GO) or BBO problem. Sometimes we can also refer to *zeroth-order optimization* (ZO), in contrast to *first-order optimization* for which gradient-based information is available.

We study the noisy setting in this thesis. Typical approaches to address GO include *Bayesian optimization* (BO) (see e.g. Brochu et al. 2010), evolutionary algorithms and hierarchical bandits (see e.g. Bubeck et al. 2010). In this thesis, we focus on hierarchical bandits algorithms. In the literature of bandits, we often refer to *infinitely-armed bandits* or more precisely *continuum-armed bandits*.

Obviously, we can only expect a *near-optimal* solution under continuum-armed bandits. We thus opt for another performance measure, namely the *simple regret*, which is the difference between the true optimal function value and the function value of our final guess. The formal definition is provided later in Chapter 2.

In Chapter 5, we explore the possibility of designing parameter-free hierarchical-bandit algorithms with minimum (smoothness) assumptions and are adaptive to the smoothness. To this end, we use a cross-validation scheme and construct an algorithm called GPO. GPO is a meta-algorithm that can use any hierarchical bandit algorithms as subroutine. In particular, GPO almost achieves the same simple regret guarantee as its subroutine. As a side result, we also show that HCT is a valid underlying algorithm for GPO as well as for POO proposed by Grill et al. [2015].

1.2.4 Hyper-parameter optimization

Finally, we deal with a more practical question: hyper-parameter optimization. As introduced in Section 1.1.1, efficient hyper-parameter tuning could be of great importance for machine learning practitioners. As is stated, HPO can be naturally modelled as a sequential optimization problem.

The search space of HPO can be both discrete (categorical variables, integer-valued variables, etc) and continuous (real-valued variables). It is natural to ask if hierarchical-bandit algorithms are able to achieve competitive performances for HPO against classical BO algorithms.

Additionally, inspired by a recent BAI-based algorithm Hyperband [Li et al., 2017], we seek to propose other BAI-based algorithms for HPO. Indeed, Hyperband is constructed upon an elimination-based BAI algorithm and achieves state-of-the-art performances compared to previous methods. We, on the other hand, are interested in whether Bayesian algorithms like TTTS can also be adapted to tackle HPO problems. Note that TTTS is only designed for *finitely-armed bandits*, an appropriate workaround is hence needed.

In Chapter 6, we design a robust and dynamic algorithm D-TTTS based on TTTS, and show that such Bayesian-flavored algorithms can be good candidates for applications like

hyper-parameter optimization. We also discuss a major drawback of D-TTTS, and propose a fix in the same chapter.

1.3 Publications of the PhD

The entire thesis is dedicated to one single problem – sequential bandit optimization – only with different settings, although the full journey of my PhD also includes other work that are less relevant to optimization. I restrict myself to the sequential optimization problem in this manuscript as it is the main research line that has been motivating me so far.

The purpose of this section is to provide a summary of the publications that I have participated to just for the record.

List of papers (peer-reviewed publications or preprints) included in this thesis.

- ♠ Gamification of pure exploration for linear bandits. In Proceedings of the 37th International Conference on Machine Learning (ICML), 2020. [Degenne et al., 2020a]
- ♠ Fixed-confidence guarantees for Bayesian best-arm identification. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.[Shang et al., 2020a]
- ♠ Simple (dynamic) bandit algorithms for hyper-parameter optimization. Preprint, 2020. [Shang et al., 2020b]
- ♠ A simple dynamic bandit algorithm for hyper-parameter tuning. In 6th Workshop on Automated Machine Learning at International Conference on Machine Learning (ICML-AutoML), 2019. [Shang et al., 2019b]
- ♠ General parallel optimisation without a metric. In Proceedings of the 30th International Conference on Algorithmic Learning Theory (ALT), 2019. [Shang et al., 2019a]

List of papers (peer-reviewed publications or preprints) not included in this thesis. The list below presents other work that are not included in this thesis since they do not really fit in with the scope of the main story line.

- ♠ UCB momentum Q-learning: Correcting bias without forgetting. In Proceedings of the 38th International Conference on Machine Learning (ICML), 2021. [Ménard et al., 2021]
- ♠ Stochastic bandits with vector losses: Minimizing infinite norm of relative losses. Preprint, 2020. [Shang et al., 2020c]

Open source software.

- ♠ rlberry - A reinforcement learning library for research and education. Github Repository, 2021. [Domingues et al., 2021]

Chapter 2

Stochastic Multi-Armed Bandits

" *Por que somos bandidos.*

Pablo Escobar

Contents

2.1 The Multi-Armed Bandits Model 10
2.1.1 Problem Formulation 11
2.1.2 Common assumptions on the rewards 12
2.1.3 Regret minimization 13
2.1.4 Optimism and UCB 15
2.2 Best-Arm Identification 16
2.2.1 Two frameworks of best-arm identification 16
2.2.2 Sampling rules 17
2.2.3 Stopping rules 18
2.2.4 Decision rules 19
2.3 Extensions of Best-Arm Identification 20
2.3.1 Pure-exploration game 20
2.3.2 Best-arm identification for linear bandits 21
2.3.3 Other variants of best-arm identification 22
2.4 Many-armed bandits 22
2.5 Performance Measure 23
2.5.1 δ-correctness and PAC learning 23
2.5.2 Sample complexity 24
2.5.3 Simple regret 25

2.1 The Multi-Armed Bandits Model

The problem of sequentially allocating resources to a defined set of actions (arms) based on successive *partially observable* (see Definition 2.1 and Remark 2.1 below) feedback refers to the MAB game in probability theory. The term *bandit* is named, by analogy, after slot machines (or one-armed bandits) in a casino. A sequential decision making problem comes up then when facing with several slot machines (multi-armed bandits).

The study of MAB problems can date back to as early as 1933 [Thompson, 1933], and was originally proposed to model sequential clinical trials. For example, researchers testing the efficacy of potential vaccines for a new coronavirus have to choose a vaccine (arm) from the following 4 options as shown in Fig. 2.1 on each patient from an experimental group of N person. For each patient $n \in [\mathrm{N}]$, researchers receive a reward signal $r_n \in \{0, 1\}$. $r_n = 1$ indicates that the vaccine is effective, otherwise the vaccine fails. We thus assume that the efficacy of each vaccine follows some Bernoulli distribution that is unknown to the researchers.

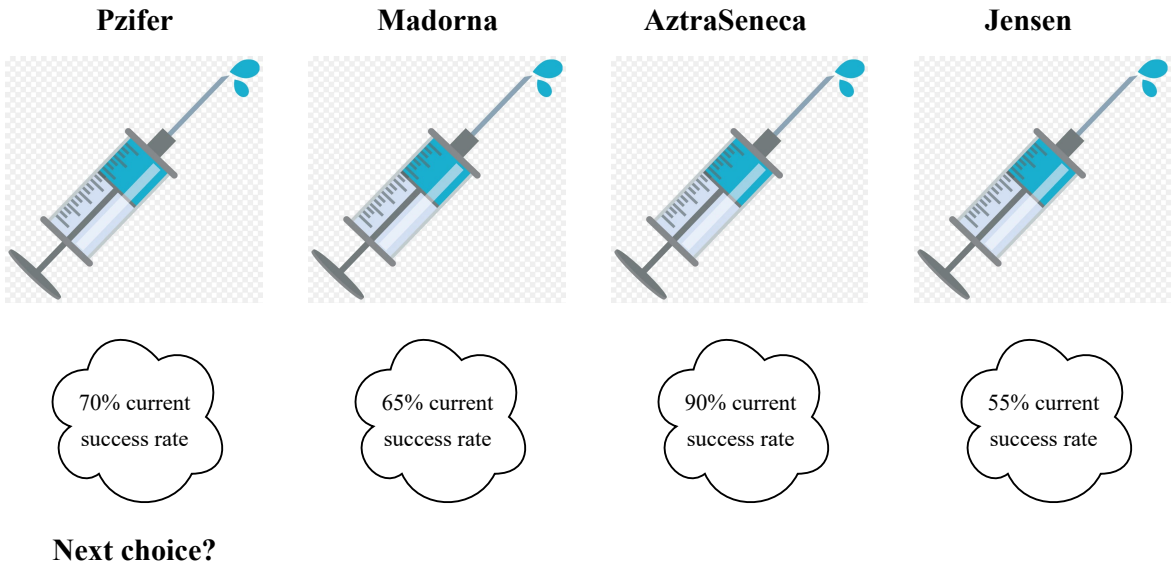


Figure 2.1: An example of modelling clinical trials as a MAB problem.

As stated in Chapter 1, a common learning objective for stochastic MAB is to maximize the total reward obtained given a sequence of observations. In the previous example, researchers need to decide which vaccine to employ for each patient depending on the previous success rates with the purpose of maximizing the total success rate $\sum_{n=1}^{\mathrm{N}} r_n$ at the end.

However, due to the complex nature of medical treatment, it turns out that the MAB model is hardly applied in real clinical trials despite its primary purpose [Réda et al., 2020]. Nevertheless, the model has been widely employed in many other applications recently, in particular online recommender systems for example (see e.g. Li et al. 2010; Zeng et al. 2016). Other application scenarios include network routing [Talebi et al., 2018], dynamic pricing [Zhai et al., 2011], demand and supply management [Brégère et al., 2019], sensor placement [Grant et al., 2019], auction bidding [Cesa-Bianchi et al., 2015], wireless communications in Internet of Things [Besson, 2019], etc.

Those applications sometimes give birth to new variants of MAB. One of the most studied variants is contextual bandits for which the average reward depend on some external context (see e.g. Krause and Ong 2011; Li et al. 2010). Linear bandits (see e.g. Abbasi-

Yadkori et al. 2011) that we study further in detail in this thesis is typically a particular case of contextual bandits. Other variants include combinatorial bandits in which a subset of arms can be selected at each round (see e.g. Cesa-Bianchi and Lugosi 2012; Chen et al. 2014; Perrault et al. 2020); structured bandits for which prior knowledge on the structure of the arm means is available (see e.g. Degenne et al. 2020b; Karnin 2016); adversarial bandits where the payoffs are controlled by a stochastic process, but rather by a (potentially oblivious) adversary (see e.g. Auer et al. 2002b); non-stationary bandits where the rewards are changing over time (see e.g. Allesiardo et al. 2017; Mellor and Shapiro 2013); multi-player bandits for which several learners exist and need to take decisions at some pre-defined moments (see e.g. Besson and Kaufmann 2018); dueling bandits for which the rewards are implicit pairwise comparison results (see e.g. Komiyama et al. 2015); delayed bandits where the rewards of current actions are not available immediately (see e.g. Vernade et al. 2017) and so on and so forth.

In the next, we go a little beyond the intuition and provide the formal definition of the model. We also recall some fundamental results for the sake of self-containedness. Of course, we do not intend to write a survey of MAB, for which the content is far too rich for this thesis. Interested readers can refer to Bubeck and Cesa-Bianchi [2012]; Lattimore and Szepesvari [2018] or Slivkins [2019] for further readings and more general results.

2.1.1 Problem Formulation

From a mathematical point of view, a MAB model is a collection of K *unknown* probability distributions $(\nu_k)_{1\leq k\leq \mathrm{K}}$. At each time step n, the learner chooses a distribution ν_{I_n} where $\mathrm{I}_n \in [\mathrm{K}]$ and receive a reward r_n that is generated from ν_{I_n}. We then recover the bandit learning cycle as shown in Fig. 2.2. We summarize such a sequential learning procedure in Definition 2.1.

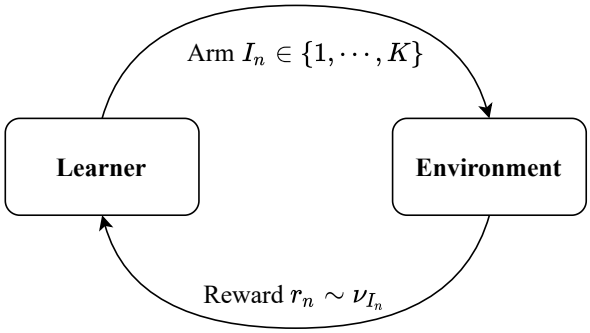


Figure 2.2: A bandit learning cycle.

Definition 2.1 (multi-armed bandit game)**.** *We are given a set of* K *arms* $\{1, \cdots, \mathrm{K}\}$ *that follow* K *unknown distributions* $(\nu_k)_{1\leq k\leq \mathrm{K}}$*, and a time horizon* N*. At each stage* $n \in \mathbb{N}$*, the bandit game consists of the following steps:*

- *a vector of rewards* $(r_{n,1} \sim \nu_1, \cdots, r_{n,\mathrm{K}} \sim \nu_{\mathrm{K}})$ *is generated,*
- *the learner picks an arm* $\mathrm{I}_n \in \{1, \cdots, \mathrm{K}\}$*, and*
- *the learner observes the reward* $r_n \triangleq r_{n,\mathrm{I}_n}$*.*

Remark 2.1. *Rewards of unchosen arms at time n are not revealed, this partial feedback setting is a special case of the online learning with experts setting.*

The way that the learner chooses the arm to pull is sometimes called a *sampling rule* or *sampling strategy*. Clearly, the sampling rule should make use of the past observations as well as the past external randomness if present. A very simple way to understand this intuition is to consider a bandit model with two arms with $\mu_1 = 1, \mu_2 = 0$. Suppose that the rewards are deterministic. In this case, if we consider a naive sampling strategy that uniformly pull the two arms would yield an expected reward of 50 after 100 rounds. However, if we pull each of the two arms once at the beginning and then start exploiting the large one (since the rewards are deterministic), we can achieve a total reward of 99 after 100 rounds. Ignoring past observations is clearly not reasonable.

In the rest of this manuscript, when ν_k are some common probability distributions, we can simply call our MAB model by the corresponding probability distribution name. For example, if the underlying reward distributions are Bernoulli (resp. Gaussian, exponential, Poisson, etc) distributions, then we can simply use Bernoulli bandits (resp. Gaussian bandits, exponential bandits, Poisson bandits, etc) to represent our bandit model.

Some useful notation. We present some useful notation that are frequently used in the rest of the thesis. First, we denote by μ_i the true mean of arm i. We further denote by $\mathrm{T}_{n,i}$ the number of selections of arm i before round n. Mathematically, $\mathrm{T}_{n,i}$ can be written as

$$\mathrm{T}_{n,i} \triangleq \sum_{\ell=1}^{n-1} \mathbb{1}\{\mathrm{I}_\ell = i\}. \tag{2.1}$$

An unbiased estimate of the true mean μ_i at time n is the empirical average reward which can be then written as

$$\mu_{n,i} = \frac{1}{\mathrm{T}_{n,i}} \sum_{\ell=1}^{n-1} \mathbb{1}\{\mathrm{I}_\ell = i\} r_{\mathrm{I}_\ell,\ell}. \tag{2.2}$$

Finally, let $\mathscr{F}_n$ be the σ-algebra generated by $(\mathrm{U}_1, \mathrm{I}_1, r_1, \cdots, \mathrm{U}_n, \mathrm{I}_n, r_n)$ where $\mathrm{U}_i \sim \mathscr{U}([0,1])$ for each $i \in [n]$.

2.1.2 Common assumptions on the rewards

One important thing to take into consideration before starting any bandit game is to take care of the assumptions on the rewards. Intuitively, we shall have a minimum prior knowledge of the 'shape' of the rewards. Obviously, the less the learner knows about that shape, the more difficult the problem is. In this thesis, several different assumptions on the reward distributions are considered depending on the problem settings. In the next, we offer a brief overview of commonly used assumptions in the literature.

Bounded rewards. The mostly considered assumption is whether the supports of the reward distributions are bounded, and if so, whether the bounds are known to the learner. In that latter case, we can assume without loss of generality that the rewards are supported on $[0,1]$. Indeed, if the rewards are contained in an arbitrary bounded interval $[a,b]$, then we can simply apply a normalization trick to recover the $[0,1]$ case.

The previous vaccine example of Fig. 2.1 with Bernoulli bandits is a typical example of known bounded rewards. Bounded rewards are widely used in many MAB research work. It is also the case for instance in Chapter 5 of this thesis.

One-dimensional exponential family. Unbounded reward distributions are obviously considered in the literature as well. An usual example of infinitely-supported reward distributions is Gaussian distribution. Therefore in the literature, we sometimes think of a more general parametric framework, namely the exponential family. The exponential family contains a large set of natural distributions as Bernoulli distributions and Gaussian distributions, hence covers a wide range of both bounded and unbounded rewards.

In practice, we often further consider a specific sub-family of distributions that is the *one-dimensional exponential family* or *single-parameter exponential family*. Typical distributions in the one-dimensional exponential family include Bernoulli distributions, Gaussian distributions with *known* variance, etc. Formally, given a random variable X whose probability distribution belongs to the single-parameter exponential family, then its *probability density function* (or *probability mass function* if X is discrete), depending only on one single parameter θ, can be written as

$$p_X(x \mid \theta) = b(x) \exp\left[\eta(\theta) \cdot T(x) + A(\theta)\right], \tag{2.3}$$

where T(X) is the *natural sufficient statistic* and b, η, A are known functions. A more formal reminder of one-dimensional exponential family is given in Appendix A.

The MAB community is interested in exponential family not only because it covers a large family of most common distributions, but also because it holds some nice properties for statistical analysis. For example, exponential family has sufficient statistics that can summarize arbitrary amounts of *independent and identically distributed* (iid) data with a finite number of samples, which is a great property in bandit analysis. Another important fact is that exponential family distributions have conjugate priors, which is extremely useful in Bayesian statistics. The latter one is for example used in Chapter 3.

Beyond... More general reward distributions are also considered in the literature. To list a few of them, we can think of sub-Gaussian distributions whose tails decay at least as fast as Gaussian distributions (see Appendix A.1.2 for a reminder of the definition), and also heavy-tailed distributions whose tails are not exponentially bounded (see e.g; Bubeck et al. 2013; Yu et al. 2018). Those reward distributions also incite interesting theoretical questions as well as applications, but are out of the scope of this manuscript.

2.1.3 Regret minimization

Once we have imposed some assumptions on the reward distributions, the next step is to fix a learning goal and set an evaluation measure accordingly.

As previously stated that the classical learning objective of a MAB learner is to maximize the total return in the long run, hence trades-off between exploration and exploitation. To achieve that goal, the learner needs to design a clever (in a precise sense) way of pulling arms based on past observations, and we call this design an *allocation strategy* or *policy*. To evaluate a strategy under this reward maximization setting, one can use the metric often referred to as *regret* defined in the next.

Suppose that each unknown distribution ν_k is associated with a mean μ_k, and that $\mu^\star$ is the mean of the optimal arm. One natural way to assess the quality of the given policy would be minimizing the total loss w.r.t the optimal arm during the whole process, which leads to the notion of *cumulative regret* (sometimes simply called regret if there is no ambiguity).

Definition 2.2 (cumulative regret)**.** *At the end of round* N*, a given policy which observes a sequence of rewards* $(r_n)_{1\leq n\leq \mathrm{N}}$ *suffers from a cumulative regret:*

$$\widehat{\mathrm{R}}_{\mathrm{N}} \triangleq \max_{i=1\cdots \mathrm{K}} \sum_{n=1}^{\mathrm{N}} r_{i,n} - \sum_{n=1}^{\mathrm{N}} r_n.$$

In general, both rewards and choices of the learner might be stochastic, it is thus often more convenient to consider a related *pseudo-regret* that involves only the mean rewards μ_k.

Definition 2.3 (cumulative pseudo-regret)**.** *At the end of round* N*, a given policy which observes a sequence of rewards* $(r_n)_{1\leq n\leq \mathrm{N}}$ *suffers from a cumulative pseudo-regret:*

$$\mathrm{R}_{\mathrm{N}} \triangleq \mu^\star \mathrm{N} - \sum_{n=1}^{\mathrm{N}} \mu_{\mathrm{I}_n}.$$

Proposition 2.1. *The expected value* $\mathbb{E}[\widehat{\mathrm{R}}_{\mathrm{N}}]$ *of the cumulative regret and the expected value* $\mathbb{E}[\mathrm{R}_{\mathrm{N}}]$ *of the cumulative pseudo-regret are the same, where the expectation is taken with respect to both rewards and choices from the learner.*

Proof. Let us define a function that relates each arm to its mean reward

$$f : \left| \begin{array}{ccc} \{1,\cdots,\mathrm{K}\} & \longrightarrow & \mathbb{R} \\ \mathrm{I}_n & \longmapsto & \mu_{\mathrm{I}_n}, \end{array} \right.$$

then by the tower rule, we have

$$\mathbb{E}[r_n] = \mathbb{E}[\mathbb{E}[r_n|\mathrm{I}_n]] = \mathbb{E}[f(\mathrm{I}_n)] = \mu_{\mathrm{I}_n}.$$

□

In practice, people are essentially interested in bounding: (a) the *expected cumulative regret*, or (b) the *cumulative regret with high probability*. One can notice that the two definitions of cumulative regret above are equivalent if their objective is to obtain an expected regret bound. People therefore often only focus on the pseudo-regret.

Clearly, minimizing the cumulative regret is equivalent to maximizing the total rewards, whence comes the name 'regret minimization'.

Since the seminal work of Robbins [1952], a significant number of research work has been made to address the regret minimization problem. Two major research lines are Upper-Confidence Bound (UCB)-type algorithms [Auer et al., 2002a; Cappé et al., 2013; Honda and Takemura, 2015], and their Bayesian competitor TS [Agrawal and Goyal, 2013; Kaufmann et al., 2012; Korda et al., 2013; Thompson, 1933]. There are also some recent works that extend the problem to the non-parametric setting [Baransi et al., 2014; Baudry et al., 2020; Chan, 2020]. Some of them even match an *asymptotic* lower regret bound proved by [Lai and Robbins, 1985].

2.1.4 Optimism and UCB

In the first chapter, we have mentioned that regret minimization is not always the most appropriate learning objective under some circumstances, but we should rather study MAB from an optimization point of view. However, before we jump into details of MAB for optimization, it may still be relevant to briefly introduce some regret-minimization methods. In particular, we recall UCB. Indeed, UCB is designed based on the *optimism in the face of uncertainty* (OFU) principle, which is inspiring for a large amount of MAB literature including ours (e.g. Chapter 4 and Chapter 5).

Before we introduce UCB in the next section, we first present a fundamental lemma that form the basis of a large number of analyzes in regret minimization. The proof of the lemma is quite straightforward and is omitted[1].

Lemma 2.1 (regret decomposition)**.** *Given a finite or countable bandit model and a horizon* N*, the cumulative regret of any strategy satisfies*

$$\mathrm{R_N} = \sum_{i=1}^{\mathrm{K}} \Delta_i \mathbb{E}\left[\mathrm{T}_{\mathrm{N},i}\right].$$

The quantities $(\Delta_i)_{i\in[\mathrm{K}]}$ is called *sub-optimality gap*. The sub-optimality gap is an important notion in MAB since it often defines the difficulty of a bandit problem instance. Its definition is given below.

Definition 2.4 (sub-optimality gap)**.** *The sub-optimality gap* Δ_i *of arm* i *is given by:*

$$\Delta_i \triangleq \mu^\star - \mu_i .$$

The algorithm of UCB is popularized by Auer et al. [2002a], is one of the first strategies that achieves a uniform logarithmic regret over the horizon N. As we just mentioned, UCB follows the OFU principle. That is to say, despite the lack of knowledge on which action is the best, we can still construct an optimistic guess that picks an optimal arm in the most favorable environments that are compatible with the observations. Here by 'compatible environments' we mean the set of possible distributions of the arms that are likely to have generated the observed rewards.

To translate OFU into mathematics, we can make use of the following upper-confidence bound index defined as

$$\mathrm{UCB}_{n,i} \triangleq \mu_{n,i} + \sqrt{\frac{3\log(n)}{2\mathrm{T}_{n,i}}}$$

for arm i until round n.

The simplest version of UCB is then given in Algorithm 2.1[2]. And the regret bound of UCB can be obtained then by Lemma 2.1 and the use of Hoeffding's inequality (see Appendix A.2 for details).

[1] Readers can refer to Chapter 4 of Lattimore and Szepesvari [2018] for a proof.
[2] The initial exploration phase of the algorithm is not mandatory.

Algorithm 2.1 Algorithm of UCB

1: **for** $n = 1..\mathrm{K}$ **do**
2: Play arm n and observe the reward r_n
3: Update the UCB index of arm i
4: **end for**
5: **for** $n \leftarrow \mathrm{K}+1, \cdots, \mathrm{N}$ **do**
6: Choose arm by $i = \arg\max_{i \in [\mathrm{K}]} \mathrm{UCB}_{n,i}$
7: Play arm i and observe the reward r_n
8: Update the UCB index of arm i
9: **end for**

2.2 Best-Arm Identification

The rest of this chapter is dedicated to MAB for optimization. We aim to provide a formal presentation of different problem settings and related performance metrics. We put a specific focus of course on the settings to be investigated in this thesis. We begin by the general best-arm identification setting.

2.2.1 Two frameworks of best-arm identification

Recall that for the vanilla problem setup of BAI, we consider a finitely-armed bandit model, which is a collection of K probability distributions, called arms $\mathscr{X} \triangleq \{x_1, \cdots, x_\mathrm{K}\}$, parameterized by their means $\mu_1, \cdots, \mu_\mathrm{K}$. When clear from the context, we can simply denote the arms by $\{1, 2, \cdots, \mathrm{K}\}$. We assume the (unknown) best arm is unique and we denote it by $\mathrm{I}^\star_{\boldsymbol{\mu}} \triangleq \arg\max_i \mu_i$[3].

A BAI strategy or algorithm can be characterized by a triple $(\mathrm{I}_n, \mathrm{J}_n, \tau)$ at each time step, hence consists of three components:

- The first is a *sampling rule*, which selects an arm $\mathrm{I}_n \in [\mathrm{K}]$. Recall that in a MAB problem, a vector of rewards $(r_{n,1}, \cdots, r_{n,\mathrm{K}})$ is generated for all arms independently from past observations at each round, but only $r_n = r_{n,\mathrm{I}_n}$ is revealed to the learner. Note that I_n is $\mathscr{F}_{n-1}$-measurable, i.e., it can only depend on the past $n-1$ observations, and some exogenous randomness, materialized into $\mathrm{U}_{n-1} \sim \mathscr{U}([0,1])$;
- The second component is a $\mathscr{F}_n$-measurable *decision rule* J_n, which returns a guess for the best arm;
- And thirdly, the *stopping rule* τ, a stopping time with respect to $(\mathscr{F}_n)_{n \in \mathbb{N}}$, decides when the exploration is over.

In general, there are two learning frameworks of BAI: (1) *fixed-confidence setting*, first studied by [Even-dar et al., 2003] and (2) *fixed-budget setting*, first proposed by [Audibert and Bubeck, 2010].

Fixed-budget setting. In the fixed-budget setting, the learner tries to maximize the probability of returning the best (or ε-best) arm with a fixed horizon N. Therefore, the stopping rule in this case can be simply written as $\tau = \mathrm{N}$. The protocol of fixed-budget BAI can be summarized as below.

Definition 2.5 (fixed-budget best-arm identification)**.** *We are given a set of* K *arms* $\{1, \cdots, \mathrm{K}\}$ *that follow* K *unknown distributions* $(\nu_k)_{1 \le k \le \mathrm{K}}$*, and a time horizon* N*. At each time step n, the learning process consists of the following actions:*

[3] The subscript $\boldsymbol{\mu}$ can be omitted when clear from the context.

- *a vector of rewards* $(r_{n,1} \sim \nu_1, \cdots, r_{n,K} \sim \nu_K)$ *is generated,*
- *the learner picks an arm* $I_n \in \{1, \cdots, K\}$ *(according to the sampling rule),*
- *the learner observes the reward* $r_n \triangleq r_{n,I_n}$*,*
- *the learner stops when* $n = N$*, and*
- *the learner outputs a guess for the best arm* $J_N \in \{1, \cdots, K\}$ *(according to the decision rule) when they stop.*

The ultimate objective is thus to make the probability of J_N not being the optimal arm, i.e. $\mathbb{P}\left[J_N \neq I^\star\right]$, as small as possible. We postpone the discussion about the performance measure to Section 2.5.3.

Fixed-confidence setting. In the fixed-confidence setting, the learner is given a confidence level/risk δ about the quality of the returned guess of the best arm. The goal is to reach a quality level of $1-\delta$ with as few samples as possible. The learning protocol of fixed-confidence BAI can be summarized as follow.

Definition 2.6 (fixed-confidence best-arm identification)**.** *We are given a set of* K *arms* $\{1, \cdots, K\}$ *that follow* K *unknown distributions* $(\nu_k)_{1 \leq k \leq K}$*, a confidence level* δ*, and a stopping time* τ *w.r.t. the observations. At each time step n, the learning process consists of the following actions:*

- *a vector of rewards* $(r_{n,1} \sim \nu_1, \cdots, r_{n,K} \sim \nu_K)$ *is generated,*
- *the learner picks an arm* $I_n \in \{1, \cdots, K\}$ *(according to the sampling rule),*
- *the learner observes the reward* $r_n \triangleq r_{n,I_n}$*,*
- *the learner stops if* $\mathbb{P}\left[J_\tau \neq I^\star\right] \leq \delta$*, where* $I^\star$ *is the optimal arm, and*
- *the learner outputs a guess for the best arm* $J_\tau \in \{1, \cdots, K\}$ *(according to the decision rule) when they stop.*

The goal is to obtain a small expected number of samples $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$, where

$$\boldsymbol{\mu} \triangleq (\mu_1, \mu_2, \cdots, \mu_K)$$

is the underlying bandit model associated to the given set of K arms. In the rest of this thesis, we ignore the subscripts $\boldsymbol{\mu}$ for expectations and probabilities if there is no ambiguity. We postpone the discussion about the performance measure to Section 2.5.2.

Remark 2.2. *Note that these two frameworks are very different in general and do not share transferable performance guarantees, readers can refer to Carpentier and Locatelli [2016] for a detailed discussion on the topic.*

2.2.2 Sampling rules

Designing smart sampling rules is the main focus of a large part of the thesis, thus we only survey the related work in this chapter, and leave the technical details to subsequent chapters.

Fixed-budget designs. For fixed-budget BAI, the sampling rules depend on the budget N. A first line of research propose to construct lower and upper confidence bounds on the arm means, and then make use of the OFU principle to choose arms (somewhat similar to UCB-type algorithms for regret minimization, see Section 2.1.4). Those methods include UCB-E [Audibert and Bubeck, 2010], and UGapE [Gabillon et al., 2012]. Another way of treating the problem is based on arm eliminations such as Successive-Reject [Audibert and Bubeck, 2010], and Sequential-Halving [Karnin et al., 2013], where less promising arms are gradually eliminated.

Fixed-confidence designs. For fixed-confidence BAI, the majority of existing sampling rules rely on the confidence level δ: Again, some of them rely on confidence intervals such as LUCB [Kalyanakrishnan et al., 2012], UGapE [Gabillon et al., 2012], KL-LUCB and KL-Racing [Kaufmann and Kalyanakrishnan, 2013], lil'UCB [Jamieson et al., 2014]; others are elimination-based like Successive-Elimination, Median-Elimination [Even-dar et al., 2003], Exponential-Gap-Elimination [Karnin et al., 2013]. The first algorithm that does not depend on δ, Track-and-Stop, is proposed by Garivier and Kaufmann [2016].

The fixed-confidence setting is an important topic of interest of this thesis, and will be covered more thoroughly in particular in Chapter 3 and Chapter 4.

Anytime designs. The fact that the two frameworks produce sampling rules that depend either on a confidence parameter δ or a budget parameter N is not desirable in some real applications. To address this problem, Jun and Nowak [2016] propose to use a *doubling trick* upon fixed-budget algorithms like Successive-Reject and Sequential-Halving, or use a time-varying confidence parameter when dealing with the fixed-confidence setting. This allows us to stop the learning process *anytime* we want. In other words, the probability of not recommending the true best arm $I^\star$, when the learner stops, needs to decay as fast as possible. Russo [2016] provides an interesting alternative that evaluates sampling rules in a Bayesian perspective. We provide further insights on this topic in Chapter 3.

2.2.3 Stopping rules

One can observe from Definition 2.6 and Definition 2.5 that stopping rules are more sophisticated in fixed-confidence BAI, as for fixed-budget BAI the learner stops simply when the budget is exhausted even though the learner needs to know the budget in order to design the sampling rule.

One of the most applied stopping rules is constructed upon the *generalized likelihood ratio*. The stopping rule was first studied by Chernoff [1959], and has recently been reformulated by Garivier and Kaufmann [2016].

Chernoff stopping rule. Finding an appropriate stopping time τ is actually a classical hypothesis test, namely *generalized likelihood ratio test* (GLRT), to decide whether we can tell an arm is larger than another arm with a small risk δ based only on past observations.

Let $\boldsymbol{\mu}'$ denote a bandit model. For any pair of arms indexed by $i, j \in [K]$, we consider

the following generalized likelihood ratio statistic:

$$\mathrm{Z}_{n,i,j} \triangleq \log \frac{\max\limits_{\mu'_i \geq \mu'_j} p_i\left(\mathbf{r}^i_{\mathrm{T}_{n,i}}\right) p_j\left(\mathbf{r}^j_{\mathrm{T}_{n,j}}\right)}{\max\limits_{\mu'_i \leq \mu'_j} p_i\left(\mathbf{r}^i_{\mathrm{T}_{n,i}}\right) p_j\left(\mathbf{r}^j_{\mathrm{T}_{n,j}}\right)},$$

where $\mathbf{r}^i_{\mathrm{T}_{n,i}} \triangleq \{r_t : \mathrm{I}_t = i, t \leq n\}$ is the vector of observations of arm i up to round n and $p_i(r_1, \cdots, r_n$ is the likelihood of n iid samples from the underlying distribution ν_i of arm i. A more thorough discussion is given by Kaufmann and Garivier [2017].

A strong point of this statistic is that it has a closed-form expression for exponential family bandit models. Indeed, we can define for all pairs of arm i, j a weighted average of their empirical mean:

$$\mu_{n,i,j} \triangleq \frac{\mathrm{T}_{n,i}}{\mathrm{T}_{n,i} + \mathrm{T}_{n,j}} \mu_{n,i} + \frac{\mathrm{T}_{n,j}}{\mathrm{T}_{n,i} + \mathrm{T}_{n,j}} \mu_{n,j},$$

where we recall that $\mu_{n,i}$ is the empirical mean of arm i. It can be shown that if $\mu_{n,i} \geq \mu_{n,j}$, then the generalized likelihood ratio statistic can be rewritten as

$$\mathrm{Z}_{n,i,j} = \mathrm{T}_{n,i} d(\mu_{n,i}, \mu_{n,i,j}) + \mathrm{T}_{n,j} d(\mu_{n,j}, \mu_{n,i,j}).$$

and we also have $\mathrm{Z}_{n,i,j} = -\mathrm{Z}_{n,j,i}$. The quantity $d(a, b)$ denotes the KL-divergence between two probability distributions characterized respectively by a and b (see Appendix A.3.2 for a reminder). The following stopping rule thus emerges naturally:

$$\begin{aligned} \tau_\delta &\triangleq \inf\left\{n \in \mathbb{N} : \exists i \in \mathscr{X}, \forall j \in \mathscr{X} \setminus \{i\}, \mathrm{Z}_{n,i,j} > d_{n,\delta}\right\} \\ &= \inf\left\{n \in \mathbb{N} : \max_{i \in \mathscr{X}} \min_{j \in \mathscr{X} \setminus \{i\}} \mathrm{Z}_{n,i,j} > d_{n,\delta}\right\}, \end{aligned} \tag{2.4}$$

where $d_{n,\delta}$ is an exploration rate that needs to be chosen carefully.

Other options. Other stopping rules also exist, but are often explicitly or implicitly equivalent to the Chernoff stopping rule. For example, in Chapter 3, we introduce a *Bayesian stopping rule* and we can show that it has implicitly the same behaviour as the Chernoff one. It is also the case for many stopping rules in the linear bandits BAI literature, as we show in Chapter 4 that they are explicitly equivalent to the Chernoff stopping rule up to constant factors.

2.2.4 Decision rules

There exist several natural and simple decision rules that can be applied to most of the existing BAI algorithms, namely *empirical best arm* (EBA), *most played arm* (MPA) and *empirical distribution of plays* (EDP) [Bubeck et al., 2009].

We introduce first EBA which returns, according to its name, the arm with the largest empirical average reward (see Definition 2.7. EBA is the most natural decision rule that one can think of as the empirical mean is a good estimation of the true mean when the corresponding arm is sufficiently pulled.

Definition 2.7 (empirical best arm decision rule)**.** *At the end of round n, the learner decides to recommend the arm with the best empirical average reward,*

$$J_n = \arg\max_{i \in [K]} \mu_{n+1,i} .$$

Another possible decision is to output the most pulled arm (see Definition 2.8)[4].

Definition 2.8 (most played arm decision rule)**.** *At the end of round n, the learner decides to recommend the most played arm,*

$$J_n = \arg\max_{i \in [K]} T_{n+1,i} .$$

The learner can also recommend arm i with probability $T_{n,i}/n$, this is the EDP decision rule (see Definition 2.9).

Definition 2.9 (empirical distribution of plays decision rule)**.** *At the end of round n, the learner decides to recommend arm i with probability $T_{n+1,i}/(n+1)$, that is*

$$J_n \sim \boldsymbol{p}_n \triangleq \left(\frac{T_{n+1,1}}{n}, \frac{T_{n+1,2}}{n}, \cdots, \frac{T_{n+1,K}}{n} \right).$$

In practice, EBA is often used in the literature. A more detailed discussion of the three decision rules can be found in the work of Bubeck et al. [2009]. We do not try to go further on the topic in this thesis. Note, however, that the present rules are obviously not the only options for decision rules. Specific rules can be adopted for certain sampling rules. We will see that it is indeed the case in Chapter 3.

2.3 Extensions of Best-Arm Identification

For real-world applications, sometimes specific need should be met when applying BAI. The problem formulation needs to be adapted with potentially additional assumptions. Furthermore, as stated in Section 1.1.2, BAI can be studied within a more general context of pure exploration. The purpose of this section is thus to provide a brief overview of commonly studied extensions of BAI as well as other pure exploration problems. We pay particular attention to BAI for linear bandits as it is the main topic of Chapter 4.

2.3.1 Pure-exploration game

A learner in a general pure-exploration game interacts with the environment by sequentially taking actions to identify the *answer* to a pre-specified question. We denote by Θ

[4] This decision rule is, however, more natural with a regret-minimizing strategy.

the set of possible mean parameters for a specific bandit problem. We further assume that there is a finite set of answers $\mathscr{I}$. For each parameter in Θ, a *unique correct answer* is given by the function $\mathrm{I}^\star : \Theta \to \mathscr{I}$ among the $|\mathscr{I}|$ possible ones (the extension of pure exploration to multiple correct answers is studied by Degenne and Koolen 2019).

While BAI is the mostly studied setting of pure-exploration game, other types of question exist as well, e.g. threshold bandits [Locatelli et al., 2016], minimum threshold [Kaufmann et al., 2018], signed bandits [Ménard, 2019], pure exploration combinatorial bandits [Chen et al., 2014], Monte-Carlo tree search [Teraoka et al., 2014], etc.

The learning protocol of a general pure exploration problem is summarized in Definition 2.10.

Definition 2.10 (pure-exploration game)**.** *We are given a set of* K *arms that is parameterized by a parameter vector* $\boldsymbol{\mu} \in \Theta$*. At each time step* n*, the learning process consists of the following actions:*

- *a vector of rewards* $(r_{n,1}, \cdots, r_{n,\mathrm{K}})$ *is generated,*
- *the learner picks an arm* $\mathrm{I}_n \in \{1, \cdots, \mathrm{K}\}$ *(according to the sampling rule),*
- *the learner observes the reward* $r_n \triangleq r_{n,\mathrm{I}_n}$ *(possibly noisy),*
- *the learner stops according to the stopping rule* τ*, and*
- *the learner outputs a guess for the answer* $\mathrm{J}_\tau \in \mathscr{I}$ *(according to the decision rule) when they stop.*

Notably, we can recover the BAI problem from a pure-exploration game by setting the pre-specified question to be finding the best arm: $\mathrm{I}^\star = \arg\max_{i \in [\mathrm{K}]} \mu_i$, and by setting the answer set to be equal to the arm set: $\mathscr{I} = \mathscr{X}$.

2.3.2 Best-arm identification for linear bandits

In linear bandits BAI, we consider a *finitely-armed*[5] linear bandit model, where a collection of K arms[6] $\mathscr{X} \triangleq \{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{K}\} \subset \mathbb{R}^d$., is given. $d \in \mathbb{N}$ is the dimension of the arm space. Usually we assume that the arm set $\mathscr{X}$ spans $\mathbb{R}^d$. Each arm i is parameterized again by its (unknown) mean μ_i.

In the linear case, we assume that μ_i is given by a linear combination of the feature vector and a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, that is

$$\mu_i = \mathbf{x}_i^\mathsf{T} \boldsymbol{\theta}.$$

$\boldsymbol{\theta}$ is called regression parameter as it is *unknown* to the learner and needs to be approached using linear regression methods during the bandit learning. The (unknown) best arm is denoted by $\mathbf{x}^\star \triangleq \arg\max_{\mathbf{x} \in \mathscr{X}} \mathbf{x}^\mathsf{T} \boldsymbol{\theta}$, and indexed by $\mathrm{I}^\star$.

At each time step n, the learner selects an arm I_n whose corresponding context is denoted by $\widehat{\mathbf{x}}_n \triangleq \mathbf{x}_{\mathrm{I}_n}$. The learner then receives a noisy observation of the inner product of $\widehat{\mathbf{x}}_n$ and $\boldsymbol{\theta}$ as payoff,

$$r_n = \widehat{\mathbf{x}}_n^\mathsf{T} \boldsymbol{\theta} + \varepsilon_n,$$

where ε_n is the noise. The learning protocol of linear bandits BAI is given in Definition 2.11.

[5] We can also generalize to an infinite number of arms. The setting appears to be much more intricate, and is out of the scope of this thesis.

[6] Sometimes called contexts or feature vectors as well.

Definition 2.11 (linear bandits best-arm identification)**.** *We are given a set of* K *arms* $\{\mathbf{x}_1, \cdots, \mathbf{x}_K\} \in \mathbb{R}^d$ *that spans* $\mathbb{R}^d$*. At each time step* n*, the learning process consists of the following actions:*

- *a vector of rewards* $(r_{n,1} = \mathbf{x}_1^\top \boldsymbol{\theta} + \varepsilon_{1,n}, \cdots, r_{n,K} = \mathbf{x}_K^\top \boldsymbol{\theta} + \varepsilon_{K,n})$ *is generated,*
- *the learner picks an arm* $I_n \in \{1, \cdots, K\}$ *(according to the sampling rule),*
- *the learner observes the reward* $r_n \triangleq r_{n,I_n}$ *with noise* $\varepsilon_n \triangleq \varepsilon_{I_n,n}$*,*
- *the learner stops according to the stopping rule* τ*, and*
- *the learner outputs a guess for the best arm* $J_\tau \in \{1, \cdots, K\}$ *(according to the decision rule) when they stop.*

Remark 2.3. *One can observe that linear bandits BAI can be reduced to the vanilla BAI setting if we consider a set of* linearly independent *feature vectors.*

2.3.3 Other variants of best-arm identification

Linear bandits is merely a particular case of contextual bandits where the rewards are determined by an arbitrary function applied on the arm rather than its inner product with a regression parameter. BAI for general contextual bandits is much more intricate than linear bandits BAI: for example, the sample complexity lower bound of Garivier and Kaufmann [2016], that we discuss later in Section 2.5.2, has an explicit formula for linear bandits BAI (see Chapter 4), but not for contextual bandits BAI. A relatively simpler setting, namely *generalized linear bandits*, has been studied by Azizi et al. [2021]; Kazerouni and Wein [2019]. To the best of my knowledge, BAI for general contextual bandits is only studied by Deshmukh et al. [2019] in the context of *simple regret* that we define later in Section 2.5.3.

Besides, many other variants mentioned in Section 2.1 can be studied in the context of BAI as well. To name a few of them, we can think of BAI for combinatorial bandits [Chen et al., 2021], BAI for adversarial bandits [Abbasi-Yadkori et al., 2018]. We can also refer to other variants like Top-m identification where instead of finding a single best arm, we aim to find the top-m best arm (see e.g. Kalyanakrishnan and Stone 2010; Réda et al. 2021).

2.4 Many-armed bandits

Now we elaborate a bit on another important extension, for which a large number of arms are available. The arm space could be infinite, or even continuous so that it is not even possible to sample each arm once.

Remember that there exists two general learning objectives in a bandit game: regret minimization and pure exploration (see e.g. Kaufmann and Garivier 2017 for a survey). While regret minimization is also an interesting topic for many-armed bandits, our focus in this thesis is still on optimization. As introduced in Chapter 1, when the search space is infinite, it is also called global optimization.

More precisely, we consider a (measurable) arm space $\mathcal{X}$ that contains infinitely-many arms. The learning goal is to optimize an *unknown* function $f : \mathcal{X} \to \mathbb{R}$ based on N noisy evaluations, that can be sequentially selected. Each arm x is essentially a data point in the

arm space $\mathcal{X}$, and it gets its mean reward $f(x)$ through the reward function f, which is the target function to be optimized. At each round n, the learner chooses an arm $x_n \in \mathcal{X}$ and receives a reward r_n. We study the noisy setting in which the obtained reward is a noisy evaluation of f: $r_n \triangleq f(x_n) + \varepsilon_n$, where ε_n is the noise. Note that in the context of GO, practitioners are often more interested by the fixed-budget setting as the target function is usually extremely costly to evaluate, hence only a considerably limited number of function evaluations are allowed. The learning protocol can thus be given as follow.

Definition 2.12 (global optimization)**.** *We are given a measurable arm space $\mathcal{X}$, and a budget of* N *function evaluations. At each time step $n \in [\mathrm{N}]$, the learning process consists of the following actions:*

- *the learner picks a point x_n from $\mathcal{X}$,*
- *the learner observes the noisy function value of x_n as reward: $r_n \triangleq f(x_n) + \varepsilon_n$,*
- *the learner stops if $n = \mathrm{N}$, and*
- *the learner outputs a guess for the maximum when they stop.*

It would be, however, impossible to obtain a sub-linear algorithm if no structure is assumed on the arm space. Two different settings exist in the literature: (1) *infinitely-armed bandits*, and (2) *continuum-armed bandits*.

The first one is initiated by Berry et al. [1997], where a specific case of Bernoulli bandits is treated. In their paper, Berry et al. [1997] regard the Bernoulli parameters as independent observations from a probability distribution, that we call a *reservoir* in subsequent works. This setting is considered in Chapter 6.

Continuum-armed bandits setting, or sometimes also named as *$\mathcal{X}$-armed bandits* (see e.g. Bubeck et al. 2010), considers arms that lie in some *metric* space and their mean rewards form a *deterministic* or *stochastic* function with some *global* or *local* smoothness being assumed. This setting is the focus of Chapter 5.

2.5 Performance Measure

Now that we have described how the learning settings of MAB for optimization are formalized, it remains to define appropriate metrics to assess the performance of the learner.

2.5.1 δ-correctness and PAC learning

In the fixed-confidence setting, a sampling rule is always accompanied by a δ-dependent stopping rule τ_δ. As stated in Section 2.2.1, we seek to construct BAI strategies that output the true best arm as the final guess with high confidence on any bandit models of interest. This objective can be translated into building strategies that are δ-correct.

Definition 2.13 (δ-correct strategy)**.** *A BAI strategy $(\mathrm{I}_n, \mathrm{J}_n, \tau)$ is called δ-correct if for any bandit model $\boldsymbol{\mu}$ with a unique optimal arm, it holds that*

$$\mathbb{P}[\tau_\delta < \infty] = 1 \text{ and } \mathbb{P}\left[j_{\tau_\delta} \neq \mathrm{I}^\star_{\boldsymbol{\mu}}\right] \leq \delta.$$

In reality, we can prove that with a well-chosen threshold in the Chernoff stopping rule, a BAI strategy is δ-correct regardless of the choice of the sampling rule. This result can be formally stated as Theorem 2.4, and will be discussed again in Chapter 3 with more detail.

Theorem 2.4. *With $\mathscr{C}^{g_G}$ a function that satisfies $\mathscr{C}^{g_G}(x) \simeq x + \ln(x)$, we introduce the threshold*

$$d_{n,\delta} = 4\ln(4+\ln(n)) + 2\mathscr{C}^{g_G}\left(\frac{\ln((K-1)/\delta)}{2}\right). \tag{2.5}$$

Then, regardless of the sampling rule, the Chernoff stopping rule with threshold $d_{n,\delta}$ satisfy

$$\mathbb{P}\left[\tau_\delta < \infty \wedge J_{\tau_\delta} \neq I^\star\right] \leq \delta.$$

In a more general setting where the arm space is continuous, we can opt for the *probably approximately correct* (PAC) learning framework [Valiant, 1984].

Definition 2.14 ((ε,δ)-PAC strategy)**.** *A BAI strategy (I_n, J_n, τ) is called (ε,δ)-PAC if for any bandit model, it holds that*

$$\mathbb{P}[\tau_\delta < \infty] = 1 \text{ and } \mathbb{P}\left[\mu^\star - \mu_{J_{\tau_\delta}} \leq \varepsilon\right] \geq 1-\delta.$$

2.5.2 Sample complexity

In fixed-confidence BAI, the goal is to design δ-correct sampling rules with minimum samples $\mathbb{E}[\tau_\delta]$. Garivier and Kaufmann [2016] provide the following lower bound on the sample complexity when the sampling rule is δ-correct.

Theorem 2.5. *[Theorem 1 of Garivier and Kaufmann 2016] Let $\delta \in (0,1)$, for any δ-correct sampling rule and any bandit model $\boldsymbol{\mu}$, we have*

$$\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta] \geq T^\star(\boldsymbol{\mu})\, kl(\delta, 1-\delta).$$

In the theorem above, kl denotes the KL-divergence between two Bernoulli distributions (see Appendix A.3.2). $T^\star(\boldsymbol{\mu})$ is a quantity that characterizes the optimal sample complexity that we define in the next.

Let $\Sigma_K \triangleq \{\boldsymbol{\omega} : \sum_{k=1}^K \omega_k = 1\}$ be the probability simplex of dimension K. We first define a notion of *alternative set* in Definition 2.15.

Definition 2.15 (alternative set)**.** *For any bandit model $\boldsymbol{\mu}$, we define the alternative set, denoted by $\neg\boldsymbol{\mu}$, as the set of bandit models whose true best arm is different from that of $\boldsymbol{\mu}$, i.e.*

$$\neg\boldsymbol{\mu} \triangleq \left\{\boldsymbol{\mu}' : I^\star_{\boldsymbol{\mu}'} \neq I^\star_{\boldsymbol{\mu}}\right\}.$$

Then the characteristic time $T^\star(\boldsymbol{\mu})$ is given by

$$T^\star(\boldsymbol{\mu})^{-1} \triangleq \sup_{\boldsymbol{\omega}\in\Sigma_K} \inf_{\boldsymbol{\mu}'\in\neg\boldsymbol{\mu}} \left(\sum_{i=1}^{K} \omega_i d(\mu_i, \mu'_i) \right), \tag{2.6}$$

where d is the KL-divergence (see Appendix A.3.2). This quantity will be extensively discussed in Chapter 3 and Chapter 4. For the moment, we only need to keep in mind its presence.

One may observe that $\mathrm{kl}(\delta, 1-\delta)$ converges to 0 when δ tends to 0. We can thus derive an *asymptotic* sample-complexity lower bound:

$$\liminf_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\ln(1/\delta)} \geq T^\star(\boldsymbol{\mu}).$$

Those lower bounds can serve as a good criterion for judging the behaviour of a fixed-confidence BAI strategy.

2.5.3 Simple regret

In the context of continuum-armed bandits, there are two common performance criteria. Depending on the applications, cumulative regret can be of interest. However, from an optimization point of view, people are often more interested in the *simple regret* (also called *optimization error*) defined below.

Definition 2.16 (simple regret)**.** *At the end of round* N*, a given policy which observes a sequence of rewards* $(r_n)_{1\leq n\leq N}$ *and a recommendation* j_N *suffers from a simple regret:*

$$S_N \triangleq \mu^\star - \mu_{j_N}.$$

Remark 2.6. *As observed by Bubeck et al. [2009], a good cumulative regret naturally implies a good simple regret. Indeed, if we recommend* J_n *according to the decision rule EDP (see Definition 2.9), we immediately get*

$$\mathbb{E}[S_n] \leq \frac{\mathbb{E}[R_n]}{n}.$$

The converse is not necessarily true.

Finally, one can observe that simple regret is also commonly used for fixed-budget BAI for whom the goal is to minimize the error probability.

Chapter 3

A Bayesian Study of Best-Arm Identification

" *Look for your choices, pick the best one, then go with it.*

Pat Riley

Contents

3.1 Introduction 28
3.2 Bayesian BAI Strategies 29
3.2.1 Sampling rules 29
3.2.2 Rationale for T3C 30
3.2.3 Stopping and decision rules 32
3.3 Two Related Optimality Notions 33
3.4 Fixed-Confidence Analysis 35
3.4.1 Core ingredients 37
3.4.2 Proof of Theorem 3.2 37
3.4.3 Sketch of the proof of Theorem 3.3 38
3.5 Optimal Posterior Convergence 40
3.6 Numerical Illustrations 40
3.6.1 Computation of the optimal error decay rate 40
3.6.2 Empirical vs. theoretical sample complexity 41
3.6.3 Results 42
3.7 Discussion 43

3.1 Introduction

In this chapter we study the very fundamental and general setting of BAI. Recall that we consider a finite-armed bandit model $\mathscr{X} \triangleq \{1, \cdots, K\}$, parameterized by their means $\mu_1, \cdots, \mu_K$. And we focus on the fixed-confidence setting, introduced by Even-dar et al. [2003], in which given a risk parameter δ, the goal is to ensure that the probability to stop and recommend a wrong arm, $\mathbb{P}\left[J_\tau \neq I^\star\right]$, is smaller than δ, while minimizing the expected total number of samples to make this accurate recommendation, $\mathbb{E}[\tau]$.

As already elaborated in Chapter 2 that most of the existing sampling rules for the fixed-confidence setting depend on the risk parameter δ, and they either rely on careful construction of confidence intervals and use of OFU or arm eliminations. The first known sampling rule for BAI that does not depend on δ is the *tracking* rules proposed by Garivier and Kaufmann [2016], which is proved to achieve the minimal sample complexity when combined with the Chernoff stopping rule as δ goes to zero. Such an *anytime* sampling rule (neither depending on a risk δ or a budget N) is very appealing for applications, as advocated by Jun and Nowak [2016], who introduce the anytime best-arm identification framework.

In this chapter, we investigate the problem from a different perspective, and we are in particular interested in another anytime sampling rule for BAI: Top-Two Thompson Sampling (TTTS).

TTTS is inspired by the famous TS [Thompson, 1933] and studies BAI from a Bayesian point of view. TS is a Bayesian algorithm well known for regret minimization, for which it is now seen as a major competitor to UCB-typed approaches [Auer et al., 2002a; Burnetas and Katehakis, 1996; Cappé et al., 2013]. However, it is also well known that regret minimizing algorithms cannot yield optimal performance for BAI [Bubeck et al., 2011; Kaufmann and Garivier, 2017] and as we opt Thompson Sampling for BAI, then its adaptation is necessary. Such an adaptation, TTTS, was given by Russo [2016] along with the other top-two sampling rules TTPS and TTVS. By choosing between two different candidate arms in each round, these sampling rules enforce the exploration of sub-optimal arms, that would be under-sampled by vanilla TS due to its objective of maximizing rewards.

While TTTS appears to be a good anytime sampling rule for the fixed-confidence BAI when coupled with an appropriate stopping rule, so far there is no theoretical support for this employment. Indeed, the (Bayesian-flavored) asymptotic analysis of Russo [2016] shows that under TTTS, the posterior probability that $I^\star$ is the best arm converges almost surely to 1 at the best possible rate. However, this property does not by itself translate into sample complexity guarantees. Since the result of Russo [2016], Qin et al. [2017] proposed and analyzed TTEI, another Bayesian sampling rule, both in the fixed-confidence setting and in terms of posterior convergence rate. Nonetheless, similar guarantees for TTTS have been left as an open question by Russo [2016]. In the present paper, we answer this open question. In addition, we propose Top-Two Transportation Cost (T3C), a computationally more favorable variant of TTTS and extend the fixed-confidence guarantees to T3C as well.

Contributions. **1)** We propose a new Bayesian sampling rule, T3C, which is inspired by TTTS but easier to implement and computationally advantageous. **2)** We investigate two Bayesian stopping and recommendation rules and establish their δ-correctness for a bandit model with Gaussian rewards.[1] **3)** We provide the first sample complexity analysis of TTTS and T3C for a Gaussian model and our proposed stopping rule. **4)** Russo's posterior

[1] hereafter 'Gaussian bandits' or 'Gaussian model'

convergence results for TTTS were obtained under restrictive assumptions on the models and priors, which exclude the two mostly used in practice: Gaussian bandits with Gaussian priors and bandits with Bernoulli rewards[2] with Beta priors. We prove that optimal posterior convergence rates can be obtained for those two as well.

☞ This chapter is based on Shang et al. [2020a].

3.2 Bayesian BAI Strategies

In this section, we give an overview of the sampling rule TTTS and introduce T3C. We provide details for Bayesian updating for Gaussian and Bernoulli models respectively, and introduce associated Bayesian stopping and recommendation rules.

3.2.1 Sampling rules

Both TTTS and T3C employ a Bayesian machinery and make use of a prior distribution Π_1 over a set of parameters Θ, that contains the unknown true parameter vector $\boldsymbol{\mu}$. Upon acquiring observations $(r_{I_1,1}, \cdots, r_{I_{n-1},n-1})$, we update our beliefs according to Bayes' rule and obtain a posterior distribution Π_n which we assume to have density π_n w.r.t. the Lebesgue measure. Russo's analysis is restricted to strong regularity properties on the models and priors that exclude two important useful cases we consider in this paper: (1) the observations of each arm i follow a Gaussian distribution $\mathcal{N}(\mu_i, \sigma^2)$ with common known variance σ^2, with imposed Gaussian prior $\mathcal{N}(\mu_{1,i}, \sigma^2_{1,i})$, (2) all arms receive Bernoulli rewards with unknown means, with a uniform prior on each arm.

Gaussian model. For Gaussian bandits with a $\mathcal{N}(0, \kappa^2)$ prior on each mean, the posterior distribution of μ_i at round n is Gaussian with mean and variance that are respectively given by

$$\frac{\sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\} r_{I_\ell,\ell}}{T_{n,i} + \sigma^2/\kappa^2} \quad \text{and} \quad \frac{\sigma^2}{T_{n,i} + \sigma^2/\kappa^2},$$

where $T_{n,i} \triangleq \sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\}$ is the number of selections of arm i before round n. For the sake of simplicity, we consider improper Gaussian priors with $\mu_{1,i} = 0$ and $\sigma_{1,i} = +\infty$ for all $i \in \mathcal{X}$, for which

$$\mu_{n,i} = \frac{1}{T_{n,i}} \sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\} r_{I_\ell,\ell} \quad \text{and} \quad \sigma^2_{n,i} = \frac{\sigma^2}{T_{n,i}}.$$

Observe that in that case the posterior mean $\mu_{n,i}$ coincides with the empirical mean.

Beta-Bernoulli model. For Bernoulli bandits with a uniform ($\mathcal{B}eta(1,1)$) prior on each mean, the posterior distribution of μ_i at round n is a Beta distribution with shape parameters $\alpha_{n,i} = \sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\} r_{I_\ell,\ell} + 1$ and $\beta_{n,i} = T_{n,i} - \sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\} r_{I_\ell,\ell} + 1$.

Now we briefly recall TTTS and introduce T3C.

[2] hereafter 'Bernoulli bandits'

Description of TTTS. At each time step n, TTTS has two potential actions: (1) with probability β, a parameter vector $\boldsymbol{\theta}$ is sampled from Π_n, and TTTS chooses to play

$$I_n^{(1)} \triangleq \arg\max_{i \in \mathcal{X}} \theta_i ,$$

(2) and with probability $1-\beta$, the algorithm continues sampling new $\boldsymbol{\theta}'$ until we obtain a *challenger*

$$I_n^{(2)} \triangleq \arg\max_{i \in \mathcal{X}} \theta'_i$$

that is different from $I_n^{(1)}$, and TTTS then selects the challenger.

Description of T3C. One drawback of TTTS is that, in practice, when the posteriors become concentrated, it takes many Thompson samples before the challenger $I_n^{(2)}$ is obtained. We thus propose a variant of TTTS, called T3C, which alleviates this computational burden. Instead of re-sampling from the posterior until a different candidate appears, we define the challenger as the arm that has the lowest *transportation cost* $W_n(I_n^{(1)}, i)$ with respect to the first candidate (with ties broken uniformly at random).

Let $\mu_{n,i}$ be the empirical mean of arm i and

$$\mu_{n,i,j} \triangleq \frac{(T_{n,i}\mu_{n,i} + T_{n,j}\mu_{n,j})}{(T_{n,i} + T_{n,j})},$$

then we define

$$W_n(i,j) \triangleq \begin{cases} 0 & \text{if } \mu_{n,j} \geq \mu_{n,i}, \\ W_{n,i,j} + W_{n,j,i} & \text{otherwise,} \end{cases} \tag{3.1}$$

where

$$W_{n,i,j} \triangleq T_{n,i}\, d\left(\mu_{n,i}, \mu_{n,i,j}\right)$$

for any i, j. One may notice that we actually recover the generalized likelihood ratio statistic stated in Section 2.2.3.

Recall that we have closed-form expressions for the KL-divergence in some cases: in the Gaussian case, $d(\mu;\mu') = (\mu-\mu')^2/(2\sigma^2)$ while in the Bernoulli case $d(\mu;\mu') = \mu\ln(\mu/\mu') + (1-\mu)\ln(1-\mu)/(1-\mu')$ (see Appendix A.3.2). For Gaussian bandits, we can further obtain a nice closed-form expression for the transportation cost,

$$W_n(i,j) = \frac{(\mu_{n,i} - \mu_{n,j})^2}{2\sigma^2(1/T_{n,i} + 1/T_{n,j})} \mathbb{1}\{\mu_{n,j} < \mu_{n,i}\}.$$

The pseudo-code of TTTS and T3C are shown in Algorithm 3.1 and Algorithm 3.2. Note that under the Gaussian model with improper priors, one should pull each arm once at the beginning for the sake of obtaining proper posteriors.

W_n in Line 9 of Algorithm 3.2 is the transportation cost defined in (3.1).

3.2.2 Rationale for T3C

In order to explain how T3C can be seen as an approximation of the re-sampling performed by TTTS, we first need to define the *optimal action probabilities*.

Algorithm 3.1 Sampling rule of TTTS

Input: β
for $n \leftarrow 1, 2, \cdots$ **do**
  Sample $\boldsymbol{\theta} \sim \Pi_n$
  $\mathrm{I}^{(1)} \leftarrow \arg\max_{i \in \mathscr{X}} \theta_i$
  Sample $b \sim \mathscr{B}er(\beta)$
  if $b = 1$ **then**
    Evaluate arm $\mathrm{I}^{(1)}$
  else
    Repeat sample $\boldsymbol{\theta}' \sim \Pi_n$
    $\mathrm{I}^{(2)} \leftarrow \arg\max_{i \in \mathscr{X}} \theta'_i$
    until $\mathrm{I}^{(2)} \neq \mathrm{I}^{(1)}$
    Evaluate arm $\mathrm{I}^{(2)}$
  end if
  Update mean and variance
  $t = t + 1$
end for

Algorithm 3.2 Sampling rule of T3C

Input: β
for $n \leftarrow 1, 2, \cdots$ **do**
  Sample $\boldsymbol{\theta} \sim \Pi_n$
  $\mathrm{I}^{(1)} \leftarrow \arg\max_{i \in \mathscr{X}} \theta_i$
  Sample $b \sim \mathscr{B}er(\beta)$
  if $b = 1$ **then**
    Evaluate arm $\mathrm{I}^{(1)}$
  else
    $\mathrm{I}^{(2)} \leftarrow \arg\min_{i \neq \mathrm{I}^{(1)}} \mathrm{W}_n(\mathrm{I}^{(1)}, i)$, cf. (3.1)
    Evaluate arm $\mathrm{I}^{(2)}$
  end if
  Update mean and variance
  $t = t + 1$
end for

Optimal action probability. The optimal action probability $a_{n,i}$ is defined as the posterior probability that arm i is optimal. Formally, letting Θ_i be the subset of Θ such that arm i is the optimal arm,

$$\Theta_i \triangleq \left\{ \boldsymbol{\theta} \in \Theta \;\middle|\; \theta_i > \max_{j \neq i} \theta_j \right\},$$

then we define

$$a_{n,i} \triangleq \Pi_n(\Theta_i) = \int_{\Theta_i} \pi_n(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}.$$

With this notation, one can show that under TTTS,

$$\Pi_n\left(\mathrm{I}_n^{(2)} = j | \mathrm{I}_n^{(1)} = i\right) = \frac{a_{n,j}}{\sum_{k \neq i} a_{n,k}}. \tag{3.2}$$

Furthermore, when i coincides with the empirical best mean (and this will often be the case for $\mathrm{I}_n^{(1)}$ when n is large due to posterior convergence) one can write

$$a_{n,j} \simeq \Pi_n\left(\theta_j \geq \theta_i\right) \simeq \exp\left(-\mathrm{W}_n(i,j)\right),$$

where the last step is justified in Lemma 3.2 in the Gaussian case (and Lemma B.18 in Appendix B.7.3 in the Bernoulli case). Hence, T3C replaces sampling from the distribution (3.2) by an approximation of its mode which is *easy to compute*. Note that directly

computing the mode would require to compute $a_{n,j}$, which is much more costly than the computation of $\mathrm{W}_n(i,j)$[3].

3.2.3 Stopping and decision rules

In order to use TTTS or T3C as sampling rule for fixed-confidence BAI, we need to additionally define stopping and decision rules. While Qin et al. [2017] suggest to couple TTEI with the "frequentist" Chernoff stopping rule [Garivier and Kaufmann, 2016], we propose in this section natural Bayesian stopping and recommendation rule. They both rely on the optimal action probabilities defined above.

Bayesian recommendation rule. At time step n, a natural candidate for the best arm is the arm with largest optimal action probability, hence we define

$$\mathrm{J}_n \triangleq \operatorname*{arg\,max}_{i \in \mathscr{X}} a_{n,i}.$$

Bayesian stopping rule. In view of the recommendation rule, it is natural to stop when the posterior probability that the recommended action is optimal is large, and exceeds some threshold $c_{n,\delta}$ which gets close to 1. Hence our Bayesian stopping rule is

$$\tau_\delta \triangleq \inf\left\{ n \in \mathbb{N} : \max_{i \in \mathscr{X}} a_{n,i} \geq c_{n,\delta} \right\}. \tag{3.3}$$

Links with frequentist counterparts. Using the transportation cost $\mathrm{W}_n(i,j)$ defined in (3.1), the Chernoff stopping rule of Garivier and Kaufmann [2016] can actually be rewritten as

$$\tau_\delta^{\text{Ch.}} \triangleq \inf\left\{ n \in \mathbb{N} : \max_{i \in \mathscr{X}} \min_{j \in \mathscr{X} \setminus \{i\}} \mathrm{W}_n(i,j) > d_{n,\delta} \right\}. \tag{3.4}$$

This stopping rule coupled with the recommendation rule $\mathrm{J}_n = \operatorname{arg\,max}_i \mu_{n,i}$.

As explained in that paper, $\mathrm{W}_n(i,j)$ can be interpreted as a (log) Generalized Likelihood Ratio statistic for rejecting the hypothesis $\mathscr{H}_0 : (\mu_i < \mu_j)$. Through our Bayesian lens, we rather have in mind the approximation $\Pi_n(\theta_j > \theta_i) \simeq \exp\{-\mathrm{W}_n(i,j)\}$, valid when $\mu_{n,i} > \mu_{n,j}$, which permits to analyze the two stopping rules using similar tools, as will be seen in the proof of Theorem 3.2.

As shown later in Section 3.4, τ_δ and $\tau_\delta^{\text{Ch.}}$ prove to be fairly similar for some corresponding choices of the thresholds $c_{n,\delta}$ and $d_{n,\delta}$. This endorses the use of the Chernoff stopping rule in practice, which does not require the (heavy) computation of optimal action probabilities. Still, our sample complexity analysis applies to the two stopping rules, and we believe that a frequentist sample complexity analysis of a fully Bayesian BAI strategy is a nice theoretical contribution.

Useful notation. We follow the notation of Russo [2016] and define the following measures of effort allocated to arm i up to time n,

$$\psi_{n,i} \triangleq \mathbb{P}\left[\mathrm{I}_n = i | \mathscr{F}_{n-1}\right] \quad \text{and} \quad \Psi_{n,i} \triangleq \sum_{l=1}^{n} \psi_{l,i}.$$

[3] the TTPS sampling rule [Russo, 2016] also requires the computation of $a_{n,i}$, thus we do not report simulations for this Bayesian sampling rule in Section 3.6

In particular, for TTTS we have

$$\psi_{n,i} = \beta a_{n,i} + (1-\beta) a_{n,i} \sum_{j \neq i} \frac{a_{n,j}}{1 - a_{n,j}},$$

while for T3C

$$\psi_{n,i} = \beta a_{n,i} + (1-\beta) \sum_{j \neq i} a_{n,j} \frac{\mathbb{1}\{W_n(j,i) = \min_{k \neq j} W_n(j,k)\}}{\# \left| \arg\min_{k \neq j} W_n(j,k) \right|}.$$

Recall that $\Sigma_K = \{\boldsymbol{\omega} : \sum_{k=1}^K \omega_k = 1\}$ is the probability simplex of dimension K.

3.3 Two Related Optimality Notions

In the fixed-confidence setting, we aim for building δ-correct strategies, i.e. strategies that identify the best arm with high confidence on any problem instance (see Definition 2.13).

Among δ-correct strategies, we seek the one with the smallest sample complexity $\mathbb{E}[\tau_\delta]$. So far, TTTS has not been analyzed in terms of sample complexity; Russo [2016] focusses on posterior consistency and optimal convergence rates. Interestingly, both the smallest possible sample complexity and the fastest rate of posterior convergence can be expressed in terms of the following quantities.

Definition 3.1. *Define for all* $i \neq I^\star$

$$C_i(\omega, \omega') \triangleq \min_{x \in \mathcal{I}} \omega d(\mu_{I^\star}; x) + \omega' d(\mu_i; x),$$

where $d(\mu, \mu')$ *is the KL-divergence and* $\mathcal{I} = \mathbb{R}$ *in the Gaussian case and* $\mathcal{I} = [0,1]$ *in the Bernoulli case. We define*

$$\begin{aligned} T^\star(\boldsymbol{\mu})^{-1} &\triangleq \max_{\boldsymbol{\omega} \in \Sigma_K} \min_{i \neq I^\star} C_i(\omega_{I^\star}, \omega_i), \\ T^\star_\beta(\boldsymbol{\mu})^{-1} &\triangleq \max_{\substack{\boldsymbol{\omega} \in \Sigma_K \\ \omega_{I^\star} = \beta}} \min_{i \neq I^\star} C_i(\omega_{I^\star}, \omega_i). \end{aligned} \tag{3.5}$$

Note that the $T^\star(\boldsymbol{\mu})$ in Definition 3.1 is equivalent to the one defined in Definition 2.6[4].

The quantity $C_i(\omega_{I^\star}, \omega_i)$ can be interpreted as a "transportation cost"[5] from the original bandit instance $\boldsymbol{\mu}$ to an alternative instance in which the mean of arm i is larger than that of $I^\star$, when the proportion of samples allocated to each arm is given by the vector $\boldsymbol{\omega} \in \Sigma_K$. As shown by Russo [2016], the $\boldsymbol{\omega}$ that maximizes (3.5) is unique, which allows us to define the β-*optimal allocation* $\boldsymbol{\omega}^\beta$ in the following proposition.

Proposition 3.1. *There is a unique solution* $\boldsymbol{\omega}^\beta$ *to the optimization problem* (3.5) *satisfying* $\omega^\beta_{I^\star} = \beta$*, and for all* $i, j \neq I^\star$, $C_i(\beta, \omega^\beta_i) = C_j(\beta, \omega^\beta_j)$.

Proof. We handle the existence and the uniqueness separately as below.

[4] Readers can refer to Section 2.2 of Garivier and Kaufmann [2016] for a proof.

[5] For which $W_n(I^\star, i)$ is an empirical counterpart.

Existence: For any arm $i \neq \mathrm{I}^\star$, C_i is a continuous function, so as to $\min_{i\neq \mathrm{I}^\star} \mathrm{C}_i$. According to the *extreme value theorem*, function $\min_{i\neq \mathrm{I}^\star} \mathrm{C}_i(\beta,\cdot)$ must attain its maximum over $[0,1]^{\mathrm{K}-1}$ which is compact. Suppose that $\boldsymbol{\omega}^\beta$ is a such maximizer. We thus have

$$\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1} = \max_{\boldsymbol{\omega}:\omega_{\mathrm{I}^\star}=\beta} \min_{i\neq \mathrm{I}^\star} \mathrm{C}_i(\beta,\omega_i) = \min_{i\neq \mathrm{I}^\star} \mathrm{C}_i(\beta,\omega_i^\beta).$$

Let us assume that $\boldsymbol{\omega}^\beta$ does not verify the second condition, which means there exists some $j \neq \mathrm{I}^\star$ such that

$$\mathrm{C}_j(\beta,\omega_i^\beta) > \mathrm{C}_{i^\star}(\beta,\omega_{i^\star}^\beta),$$

where $i^\star \triangleq \arg\min_{i\neq \mathrm{I}^\star} \mathrm{C}_i(\beta,\omega_i^\beta)$.

Now if we subtract a small quantity $\varepsilon > 0$, from $\mathrm{C}_j(\beta,\omega_j^\beta)$, such that

$$\varepsilon \leq \frac{\mathrm{C}_j(\beta,\omega_j^\beta) - \mathrm{C}_{i^\star}(\beta,\omega_{i^\star}^\beta)}{2},$$

and add $\varepsilon/(\mathrm{K}-2)$ to $\mathrm{C}_i(\beta,\omega_i^\beta)$ for any $i \neq j, \mathrm{I}^\star$, we would not change the order of the $\mathrm{C}_i(\beta,\omega_i^\beta)$. Therefore, $i^\star$ remains unchanged, however, the new $\mathrm{C}_{i^\star}(\beta,\omega_{i^\star}^\beta)$ would be strictly larger than the previous one which contradicts the definition of $\boldsymbol{\omega}^\beta$.

Uniqueness: We now need to show that the solution is unique. Suppose that two different maximizers $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$ exist, and there exists some $i \neq \mathrm{I}^\star$ such that $\omega_i > \omega'_i$. Since $\mathrm{C}_i(\beta,\cdot)$ is an strictly increasing function, thus we have $\mathrm{C}_i(\beta,\omega_i) > \mathrm{C}_i(\beta,\omega'_i)$. By consequence, for any $j \neq i$ and $j' \neq i$,

$$\mathrm{C}_j(\beta,\omega_j) = \mathrm{C}_i(\beta,\omega_i) > \mathrm{C}_i(\beta,\omega'_i) = \mathrm{C}_{j'}(\beta,\omega'_{j'}).$$

Therefore, for any $j \neq \mathrm{I}^\star$ and $j' \neq \mathrm{I}^\star$, $\omega_j > \omega'_j$, and

$$\sum_{j\neq \mathrm{I}^\star} \omega_j > \sum_{j\neq \mathrm{I}^\star} \omega'_j.$$

However, we know that $1 - \sum_{j\neq \mathrm{I}^\star} \omega_j = \omega_{\mathrm{I}^\star} = \beta = \omega'_{\mathrm{I}^\star} = 1 - \sum_{j\neq \mathrm{I}^\star} \omega'_j$, contradiction!

□

For models with more than two arms, there is no closed form expression for $\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}$ or $\mathrm{T}^\star(\boldsymbol{\mu})^{-1}$, even for Gaussian bandits with variance σ^2 for which we have

$$\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1} = \max_{\boldsymbol{\omega}:\omega_{\mathrm{I}^\star}=\beta} \min_{i\neq \mathrm{I}^\star} \frac{(\mu_{\mathrm{I}^\star} - \mu_i)^2}{2\sigma^2(1/\omega_i + 1/\beta)}.$$

Bayesian β-optimality. Russo [2016] proves that any sampling rule allocating a fraction β to the optimal arm ($\Psi_{n,\mathrm{I}^\star}/n \to \beta$) satisfies $1 - a_{n,\mathrm{I}^\star} \geq e^{-n(\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}+o(1))}$ (a.s.) for large values of n. We define a *Bayesian β-optimal* sampling rule as a sampling rule matching this lower bound, i.e. satisfying $\Psi_{n,\mathrm{I}^\star}/n \to \beta$ and $1 - a_{n,\mathrm{I}^\star} \leq e^{-n(\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}+o(1))}$.

Russo [2016] proves that TTTS with parameter β is Bayesian β-optimal. However, the result is valid only under strong regularity assumptions, excluding the two practically important cases of Gaussian and Bernoulli bandits. In this paper, we complete the picture by establishing Bayesian β-optimality for those models in Section 3.5. For the Gaussian

bandit, Bayesian β-optimality was established for TTEI by Qin et al. [2017] with Gaussian priors, but this remained an open problem for TTTS.

A fundamental ingredient of these proofs is to establish the convergence of the allocation of measurement effort to the β-optimal allocation: $\Psi_{n,i}/n \to \omega_i^\beta$ for all i, which is equivalent to $T_{n,i}/n \to \omega_i^\beta$ (cf. Lemma 3.4).

β-optimality in the fixed-confidence setting. In the fixed confidence setting, the performance of an algorithm is evaluated in terms of sample complexity. A lower bound given by Garivier and Kaufmann [2016] states that any δ-correct strategy satisfies $\mathbb{E}[\tau_\delta] \geq T^\star(\boldsymbol{\mu}) \ln(1/(3\delta))$.

Observe that $T^\star(\boldsymbol{\mu})^{-1} = \max_{\beta\in[0,1]} T^\star_\beta(\boldsymbol{\mu})^{-1}$. Using the same lower bound techniques, one can also prove that under any δ-correct strategy satisfying $T_{n,I^\star}/n \to \beta$,

$$\liminf_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\ln(1/\delta)} \geq T^\star_\beta(\boldsymbol{\mu}).$$

This motivates the relaxed optimality notion that we introduce in this paper: A BAI strategy is called *asymptotically β-optimal* if it satisfies

$$\frac{T_{n,I^\star}}{n} \to \beta \quad \text{and} \quad \limsup_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\ln(1/\delta)} \leq T^\star_\beta(\boldsymbol{\mu}).$$

In the paper, we provide the first sample complexity analysis of a BAI algorithm based on TTTS (with the stopping and recommendation rules described in Section 3.2), establishing its asymptotic β-optimality.

As already observed by Qin et al. [2017], any sampling rule converging to the β-optimal allocation (i.e. satisfying $T_{n,i}/n \to w_i^\beta$ for all i) can be shown to satisfy

$$\limsup_{\delta\to 0} \tau_\delta / \ln(1/\delta) \leq T^\star_\beta(\boldsymbol{\mu})$$

almost surely, when coupled with the Chernoff stopping rule. The fixed confidence optimality that we define above is stronger as it provides guarantees on $\mathbb{E}[\tau_\delta]$.

3.4 Fixed-Confidence Analysis

In this section, we consider Gaussian bandits and the Bayesian rules using an improper prior on the means. We state our main result below, showing that TTTS and T3C are asymptotically β-optimal in the fixed confidence setting, when coupled with appropriate stopping and recommendation rules.

Theorem 3.1. *With $\mathcal{C}^{g_G}$ the function defined by Kaufmann and Koolen [2018], which satisfies $\mathcal{C}^{g_G}(x) \simeq x + \ln(x)$, we introduce the threshold*

$$d_{n,\delta} = 4\ln(4+\ln(n)) + 2\mathcal{C}^{g_G}\left(\frac{\ln((K-1)/\delta)}{2}\right). \tag{3.6}$$

The TTTS and T3C sampling rules coupled with either

- *the Bayesian stopping rule* (3.3) *with threshold*

$$c_{n,\delta} = 1 - \frac{1}{\sqrt{2\pi}} e^{-\left(\sqrt{d_{n,\delta}} + \frac{1}{\sqrt{2}}\right)^2}$$

  and the recommendation rule $J_t = \arg\max_i a_{n,i}$
- *or the Chernoff stopping rule* (3.4) *with threshold* $d_{n,\delta}$ *and recommendation rule* $J_t = \arg\max_i \mu_{n,i}$,

form a δ*-correct BAI strategy. Moreover, if all the arms means are distinct, it satisfies*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \leq T^\star_\beta(\boldsymbol{\mu}).$$

We now give the proof of Theorem 3.1, which is divided into three parts. The **first step** of the analysis is to prove the δ-correctness of the studied BAI strategies.

Theorem 3.2. *Regardless of the sampling rule, the stopping rule (3.3) with the threshold* $c_{n,\delta}$ *and the Chernoff stopping rule with threshold* $d_{n,\delta}$ *defined in Theorem 3.1 satisfy* $\mathbb{P}\left[\tau_\delta < \infty \wedge J_{\tau_\delta} \neq I^\star\right] \leq \delta$.

To prove that TTTS and T3C allow to reach a β-optimal sample complexity, one needs to quantify how fast the measurement effort for each arm is concentrating to its corresponding optimal weight. For this purpose, we introduce the random variable

$$T^\varepsilon_\beta \triangleq \inf\left\{N \in \mathbb{N} : \max_{i \in \mathscr{A}} |T_{n,i}/n - \omega^\beta_i| \leq \varepsilon, \forall n \geq N\right\}.$$

The **second step** of our analysis is a sufficient condition for β-optimality, stated in Lemma 3.1. Its proof is given in Appendix B.5. The same result was proven for the Chernoff stopping rule by Qin et al. [2017].

Lemma 3.1. *Let* $\delta, \beta \in (0,1)$. *For any sampling rule which satisfies* $\mathbb{E}\left[T^\varepsilon_\beta\right] < \infty$ *for all* $\varepsilon > 0$, *we have*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \leq T^\star_\beta(\boldsymbol{\mu}),$$

if the sampling rule is coupled with stopping rule (3.3),

Finally, it remains to show that TTTS and T3C meet the sufficient condition, and therefore the **last step**, which is the core component and the most technical part our analysis, consists of showing the following.

Theorem 3.3. *Under TTTS or T3C,* $\mathbb{E}\left[T^\varepsilon_\beta\right] < +\infty$.

In the rest of this section, we prove Theorem 3.2 and sketch the proof of Theorem 3.3. But we first highlight some important ingredients for these proofs.

3.4.1 Core ingredients

Our analysis hinges on properties of the Gaussian posteriors, in particular on the following tails bounds, which follow from Lemma 1 of Qin et al. [2017].

Lemma 3.2. *For any $i, j \in \mathscr{A}$, if $\mu_{n,i} \le \mu_{n,j}$*

$$\Pi_n\left[\theta_i \ge \theta_j\right] \le \frac{1}{2}\exp\left\{-\frac{\left(\mu_{n,j}-\mu_{n,i}\right)^2}{2\sigma^2_{n,i,j}}\right\}, \tag{3.7}$$

$$\Pi_n\left[\theta_i \ge \theta_j\right] \ge \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{\left(\mu_{n,j}-\mu_{n,i}+\sigma_{n,i,j}\right)^2}{2\sigma^2_{n,i,j}}\right\}, \tag{3.8}$$

where $\sigma^2_{n,i,j} \triangleq \sigma^2/\mathrm{T}_{n,i} + \sigma^2/\mathrm{T}_{n,j}$.

This lemma is crucial to control $a_{n,i}$ and $\psi_{n,i}$, the optimal action and selection probabilities.

3.4.2 Proof of Theorem 3.2

We upper bound the desired probability as follows

$$\begin{aligned}
\mathbb{P}\left[\tau_\delta < \infty \wedge \mathrm{J}_{\tau_\delta} \neq \mathrm{I}^\star\right] &\le \sum_{i\neq \mathrm{I}^\star} \mathbb{P}\left[\exists n \in \mathbb{N} : a_{n,i} > c_{n,\delta}\right] \\
&\le \sum_{i\neq \mathrm{I}^\star} \mathbb{P}\left[\exists n \in \mathbb{N} : \Pi_n(\theta_i \ge \theta_{\mathrm{I}_\star}) > c_{n,\delta}, \mu_{n,\mathrm{I}^\star} \le \mu_{n,i}\right] \\
&\le \sum_{i\neq \mathrm{I}^\star} \mathbb{P}\left[\exists n \in \mathbb{N} : 1-c_{n,\delta} > \Pi_n(\theta_{\mathrm{I}^\star} > \theta_i), \mu_{n,\mathrm{I}^\star} \le \mu_{n,i}\right].
\end{aligned}$$

The second step uses the fact that as $c_{n,\delta} \ge 1/2$, a necessary condition for $\Pi_n(\theta_i \ge \theta_{\mathrm{I}_\star}) \ge c_{n,\delta}$ is that $\mu_{n,i} \ge \mu_{n,\mathrm{I}_\star}$. Now using the lower bound (3.8), if $\mu_{n,\mathrm{I}^\star} \le \mu_{n,i}$, the inequality $1-c_{n,\delta} > \Pi_n(\theta_{\mathrm{I}^\star} > \theta_i)$ implies

$$\frac{(\mu_{n,i}-\mu_{n,\mathrm{I}^\star})^2}{2\sigma^2_{n,i,\mathrm{I}^\star}} \ge \left(\sqrt{\ln\frac{1}{\sqrt{2\pi}(1-c_{n,\delta})}} - \frac{1}{\sqrt{2}}\right)^2 = d_{n,\delta},$$

where the equality follows from the expression of $c_{n,\delta}$ as function of $d_{n,\delta}$. Hence to conclude the proof it remains to check that

$$\mathbb{P}\left[\exists n\in\mathbb{N} : \mu_{n,i} \ge \mu_{n,\mathrm{I}^\star}, \frac{(\mu_{n,i}-\mu_{n,\mathrm{I}^\star})^2}{2\sigma^2_{n,i,\mathrm{I}^\star}} \ge d_{n,\delta}\right] \le \frac{\delta}{\mathrm{K}-1}. \tag{3.9}$$

To prove this, we observe that for $\mu_{n,i} \ge \mu_{n,\mathrm{I}^\star}$,

$$\begin{aligned}
\frac{(\mu_{n,i}-\mu_{n,\mathrm{I}^\star})^2}{2\sigma^2_{n,i,\mathrm{I}^\star}} &= \inf_{\theta_i<\theta_{\mathrm{I}^\star}} \mathrm{T}_{n,i}\, d(\mu_{n,i};\theta_i) + \mathrm{T}_{n,\mathrm{I}^\star}\, d(\mu_{n,\mathrm{I}^\star};\theta_{\mathrm{I}^\star}) \\
&\le \mathrm{T}_{n,i}\, d(\mu_{n,i};\mu_i) + \mathrm{T}_{n,\mathrm{I}^\star}\, d(\mu_{n,\mathrm{I}^\star};\mu_{\mathrm{I}^\star}).
\end{aligned}$$

Corollary 10 of Kaufmann and Koolen [2018] then allows us to upper bound the probability

$$\mathbb{P}\left[\exists n \in \mathbb{N} : \mathrm{T}_{n,i}\, d(\mu_{n,i};\mu_i) + \mathrm{T}_{n,\mathrm{I}^\star}\, d(\mu_{n,\mathrm{I}^\star},\mu_{\mathrm{I}^\star}) \ge d_{n,\delta}\right]$$

by $\delta/(K-1)$ for the choice of threshold given in (3.6), which completes the proof that the stopping rule (3.3) is δ-correct. The fact that the Chernoff stopping rule with the above threshold $d_{n,\delta}$ given above is δ-correct straightforwardly follows from (3.9).

3.4.3 Sketch of the proof of Theorem 3.3

We present a unified proof sketch of Theorem 3.3 for TTTS and T3C. While the two analyses follow the same steps, some of the lemmas given below have different proofs for TTTS and T3C, which can be found in Appendix B.3 and Appendix B.4 respectively.

We first state two important concentration results, that hold under any sampling rule.

Lemma 3.3. *[Lemma 5 of Qin et al. 2017] There exists a random variable* W_1*, such that for all* $i \in \mathcal{A}$*,*

$$\forall n \in \mathbb{N}, \quad |\mu_{n,i} - \mu_i| \leq \sigma W_1 \sqrt{\frac{\log(e + T_{n,i})}{1 + T_{n,i}}} \ a.s.,$$

and $\mathbb{E}\left[e^{\lambda W_1}\right] < \infty$ *for all* $\lambda > 0$.

Lemma 3.4. *There exists a random variable* W_2*, such that for all* $i \in \mathcal{A}$*,*

$$\forall n \in \mathbb{N}, |T_{n,i} - \Psi_{n,i}| \leq W_2 \sqrt{(n+1)\log(e^2 + n)} \ a.s.,$$

and $\mathbb{E}\left[e^{\lambda W_2}\right] < \infty$ *for any* $\lambda > 0$.

Lemma 3.3 controls the concentration of the posterior means towards the true means and Lemma 3.4 establishes that $T_{n,i}$ and $\Psi_{n,i}$ are close. Both results rely on uniform deviation inequalities for martingales.

Our analysis uses the same principle as that of TTEI: We establish that T^{ε}_{β} is upper bounded by some random variable N which is a polynomial of the random variables W_1 and W_2 introduced in the above lemmas, denoted by $\mathrm{Poly}(W_1, W_2) \triangleq \mathcal{O}(W_1^{c_1} W_2^{c_2})$, where c_1 and c_2 are two constants (that may depend on arms' means and the constant hidden in the $\mathcal{O}$). As all exponential moments of W_1 and W_2 are finite, N has a finite expectation as well, which concludes the proof.

The first step to exhibit such an upper bound N is to establish that every arm is pulled sufficiently often.

Lemma 3.5. *Under TTTS or T3C, there exists* $N_1 = \mathrm{Poly}(W_1, W_2)$ *s.t.* $\forall n \geq N_1$*, for all* i*,* $T_{n,i} \geq \sqrt{n/K}$*, almost surely.*

Due to the randomized nature of TTTS and T3C, the proof of Lemma 3.5 is significantly more involved than for a deterministic rule like TTEI. Intuitively, the posterior of each arm would be well concentrated once the arm is sufficiently pulled. If the optimal arm is under-sampled, then it would be chosen as the first candidate with large probability. If a sub-optimal arm is under-sampled, then its posterior distribution would possess a

relatively wide tail that overlaps with or cover the somehow narrow tails of other overly-sampled arms. The probability of that sub-optimal arm being chosen as the challenger would be large enough then.

Combining Lemma 3.5 with Lemma 3.3 straightforwardly leads to the following result.

Lemma 3.6. *Under TTTS or T3C, fix a constant* $\varepsilon > 0$*, there exists* $\mathrm{N}_2 = \mathit{Poly}(1/\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$ *s.t.* $\forall n \geq \mathrm{N}_2$,

$$\forall i \in \mathscr{A}, \quad |\mu_{n,i} - \mu_i| \leq \varepsilon\,.$$

We can then deduce a very nice property about the optimal action probability for sub-optimal arms from the previous two lemmas. Indeed, we can show that

$$\forall i \neq \mathrm{I}^\star, \quad a_{n,i} \leq \exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\}$$

for n larger than some $\mathrm{Poly}(\mathrm{W}_1, \mathrm{W}_2)$. In the previous inequality, $\Delta_{\min}$ is the smallest mean difference among all the arms.

Plugging this in the expression of $\psi_{n,i}$, one can easily quantify how fast $\psi_{n,\mathrm{I}^\star}$ converges to β, which eventually yields the following result.

Lemma 3.7. *Under TTTS or T3C, fix* $\varepsilon > 0$*, then there exists* $\mathrm{N}_3 = \mathit{Poly}(1/\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$ *s.t.* $\forall n \geq \mathrm{N}_3$,

$$\left|\frac{\mathrm{T}_{n,\mathrm{I}^\star}}{n} - \beta\right| \leq \varepsilon\,.$$

The last, more involved, step is to establish that the fraction of measurement allocation to every sub-optimal arm i is indeed similarly close to its optimal proportion ω_i^β.

Lemma 3.8. *Under TTTS or T3C, fix a constant* $\varepsilon > 0$*, there exists* $\mathrm{N}_4 = \mathit{Poly}(1/\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$ *s.t.* $\forall n \geq \mathrm{N}_4$,

$$\forall i \neq \mathrm{I}^\star, \quad \left|\frac{\mathrm{T}_{n,i}}{n} - \omega_i^\beta\right| \leq \varepsilon\,.$$

The major step in the proof of Lemma 3.8 for each sampling rule, is to establish that if some arm is over-sampled, then its probability to be selected is exponentially small. Formally, we show that for n larger than some $\mathrm{Poly}(1/\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$,

$$\frac{\Psi_{n,i}}{n} \geq \omega_i^\beta + \xi \quad \Rightarrow \quad \psi_{n,i} \leq \exp\left\{-f(n,\xi)\right\},$$

for some function $f(n,\xi)$ to be specified for each sampling rule, satisfying $f(n) \geq \mathrm{C}_\xi\sqrt{n}$ (a.s.). This result leads to the concentration of $\Psi_{n,i}/n$, thus can be easily converted to the concentration of $\mathrm{T}_{n,i}/n$ by Lemma 3.4.

Finally, Lemma 3.7 and Lemma 3.8 show that $\mathrm{T}_\beta^\varepsilon$ is upper bounded by $\mathrm{N} \triangleq \max(\mathrm{N}_3, \mathrm{N}_4)$, which yields $\mathbb{E}[\mathrm{T}_\beta^\varepsilon] \leq \max(\mathbb{E}[\mathrm{N}_3], \mathbb{E}[\mathrm{N}_4]) < \infty$.

3.5 Optimal Posterior Convergence

Recall that $a_{n,\mathrm{I}^\star}$ denotes the posterior mass assigned to the event that action $\mathrm{I}^\star$ (i.e. the true optimal arm) is optimal at time n. As the number of observations tends to infinity, we desire that the posterior distribution converges to the truth. In this section we show equivalently that the posterior mass on the complementory event, $1-a_{n,\mathrm{I}^\star}$, the event that arm $\mathrm{I}^\star$ is not optimal, converges to zero at an exponential rate, and that it does so at optimal rate $\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}$.

Russo [2016] proves a similar theorem under three confining boundedness assumptions (cf. Russo 2016, Asssumption 1) on the parameter space, the prior density and the (first derivative of the) log-normalizer of the exponential family. Hence, the theorems in Russo [2016] do not apply to the two bandit models most used in practise, which we consider in this paper: the Gaussian and Bernoulli model.

In the first case, the parameter space is unbounded, in the latter model, the derivative of the log-normalizer (which is $e^\eta/(1+e^\eta)$) is unbounded. Here we provide two theorems, proving that under TTTS, the optimal, exponential posterior convergence rates are obtained for the Gaussian model with uninformative (improper) Gaussian priors (proof given in Appendix B.6), and the Bernoulli model with $\mathscr{B}eta(1,1)$ priors (proof given in Appendix B.7).

Theorem 3.4. *Under TTTS, for Gaussian bandits with improper Gaussian priors, it holds almost surely that*

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,\mathrm{I}^\star}) = \mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}.$$

Theorem 3.5. *Under TTTS, for Bernoulli bandits and uniform priors, it holds almost surely that*

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,\mathrm{I}^\star}) = \mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}.$$

3.6 Numerical Illustrations

3.6.1 Computation of the optimal error decay rate

We first describe how to approximate $\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}$ under any prior of 1-dimensional exponential family. We then also provide a way to compute numerically $\mathrm{T}^\star_\beta(\boldsymbol{\mu})^{-1}$ under Gaussian prior since it can be computed more explicitly.

General case. For any $i \neq \mathrm{I}^\star$, $\mathrm{C}_i(\beta,\omega_i)$ is defined as the output to a convex minimization problem for whom the unique solution has an analytic expression

$$\frac{\beta}{\beta+\omega_i}\mu_{\mathrm{I}^\star} + \frac{\omega_i}{\beta+\omega_i}\mu_i.$$

Next, we define for any $i \neq I^\star$, a function

$$g_i : [0,+\infty[\to [0,+\infty[$$
$$x \mapsto \beta d\left(\mu_{I^\star}; \frac{\beta}{\beta+\omega_i}\mu_{I^\star} + \frac{\omega_i}{\beta+\omega_i}\mu_i\right) + x d\left(\mu_i; \frac{\beta}{\beta+\omega_i}\mu_{I^\star} + \frac{\omega_i}{\beta+\omega_i}\mu_i\right).$$

In fact, g_i is a strictly increasing function (see Garivier and Kaufmann 2016, Appendix A.2 for more details), so does its inverse function $x_i \triangleq g_i^{-1}$ which is defined on $[0, \beta d(\mu_{I^\star}; \mu_i)[$ as g_i tends to 0 when x tends to 0 and tends to $\beta d(\mu_{I^\star}; \mu_i)$ when x tends to $+\infty$.

According to Proposition 3.1, the optimal proportion vector $\boldsymbol{\omega}^\beta$ that we are searching for satisfies the constraint that $\forall i, j \neq I^\star$,

$$g_i(\omega_i^\beta) = g_j(\omega_j^\beta) = T_\beta^\star(\boldsymbol{\mu})^{-1}.$$

Since $\omega_i^\beta = g_i^{-1}(\Gamma_\beta^\star) = x_i(\Gamma_\beta^\star)$, and we know that $\sum_{i \neq I^\star} \omega_i = 1-\beta$, thus the problem of computing $\Gamma_\beta^\star$ is equivalent to solve the following equation,

$$\sum_{i \neq I^\star} x_i(y) = 1-\beta.$$

This equation has a unique solution since $\sum_{i \neq I^\star}$ is a strictly increasing function valued in $[0,+\infty[$. We can thus apply a bisection method to this function whose evaluation require itself a bisection method applied on $K-1$ smooth scalar functions.

Gaussian case. In the context of this paper, we can do a more efficient approximation. In the Gaussian case, we know that for any $i, j \neq I^\star$,

$$\frac{1}{\omega_j^\beta} + \frac{1}{\beta} = \frac{(\mu_{I^\star} - \mu_j)^2}{(\mu_{I^\star} - \mu_i)^2}\left(\frac{1}{\omega_i^\beta} + \frac{1}{\beta}\right).$$

Denote $x_i \triangleq 1/\omega_i^\beta + 1/\beta$ and $a_{ji} \triangleq (\mu_{I^\star} - \mu_j)^2/(\mu_{I^\star} - \mu_i)^2$, fix some $i \neq I^\star$, then we have $\forall j \neq I^\star$, $x_j = a_{ji}x_i$. Since $\sum_{j \neq I^\star} \omega_j^\beta = 1-\beta$, we have

$$\sum_{j \neq I^\star} \frac{1}{x_j - 1/\beta} = \sum_{j \neq I^\star} \frac{1}{a_{ji}x_i - 1/\beta} = 1-\beta.$$

Thus we only need to find the unique solution to the equation

$$\sum_{j \neq I^\star} \frac{a_{ij}}{x - a_{ij}/\beta} = 1-\beta,$$

that requires only one shot bisection method.

3.6.2 Empirical vs. theoretical sample complexity

In Fig. 3.1, we plot expected stopping time of T3C for $\delta = 0.01$ as a function of $T_\beta^\star(\boldsymbol{\mu})$ on 100 randomly generated problem instances. We see on this plot that the empirical stopping time has the right linear scaling in $T_\beta^\star(\boldsymbol{\mu})$ (ignoring a few outliers).

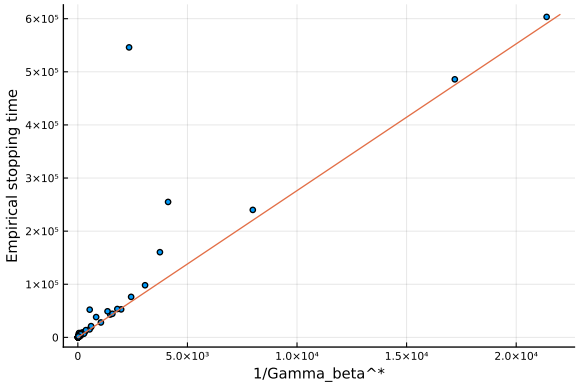


Figure 3.1: Dots: empirical sample complexity; Solid line: theoretical sample complexity.

3.6.3 Results

This section is aimed at illustrating our theoretical results and supporting the practical use of Bayesian sampling rules for fixed-confidence BAI.

We experiment with three different Bayesian sampling rules: T3C, TTTS and TTEI, and we also include the Direct Tracking (D-Tracking) rule of Garivier and Kaufmann [2016] (which is adaptive to β), the UGapE [Gabillon et al., 2012] algorithm, and a uniform baseline. In order to make a fair comparison, we use the Chernoff stopping rule (3.4) and associated recommendation rule for all of the sampling rules, including the uniform one, except for UGapE which has its own stopping rule. Furthermore, we include a top-two variant of the Best Challenger (BC) heuristic (see, e.g., Ménard, 2019).

BC selects the empirical best arm $\widehat{I}_n$ with probability β and the maximizer of $W_n(\widehat{I}_n, j)$ with probability $1-\beta$, but also performs forced exploration (selecting any arm sampled less than $\sqrt{n}$ times at round n). T3C can thus be viewed as a variant of BC in which no forced exploration is needed to converge to $\boldsymbol{\omega}^{\beta}$, due to the noise added by replacing $\widehat{I}_n$ with $I_n^{(1)}$.

We consider two simple instances with arms means given by

$$\boldsymbol{\mu}_1 = [0.5\ 0.9\ 0.4\ 0.45\ 0.44999] \text{ and } \boldsymbol{\mu}_2 = [1\ 0.8\ 0.75\ 0.7]$$

respectively. We run simulations for both Gaussian (with $\sigma = 1$) and Bernoulli bandits with a risk parameter $\delta = 0.01$. Figure 3.2 reports the empirical distribution of τ_δ under the different sampling rules, estimated over 1000 independent runs.

These figures provide several insights: (1) T3C is competitive with, and sometimes slightly better than TTTS and TTEI in terms of sample complexity. (2) The UGapE algorithm has a larger sample complexity than the uniform sampling rule, which highlights the importance of the stopping rule in the fixed-confidence setting. (3) The fact that D-Tracking performs best is not surprising, since it converges to $\boldsymbol{\omega}^{\beta^\star}$ and achieves minimal sample complexity. However, in terms of computation time, D-Tracking is much worse than other sampling rules, as can be seen in Table 3.1, which reports the average execution time of one step of each sampling rule for μ_1 in the Gaussian case. (4) TTTS also suffers from computational costs, whose origins are explained in Section 3.2, unlike T3C and TTEI. Although TTEI is already computationally more attractive than TTTS, its practical benefits are limited to the Gaussian case, since the *Expected Improvement* (EI) does not have a closed form beyond this case and its approximation would be costly. In contrast, T3C can be applied for other distributions.

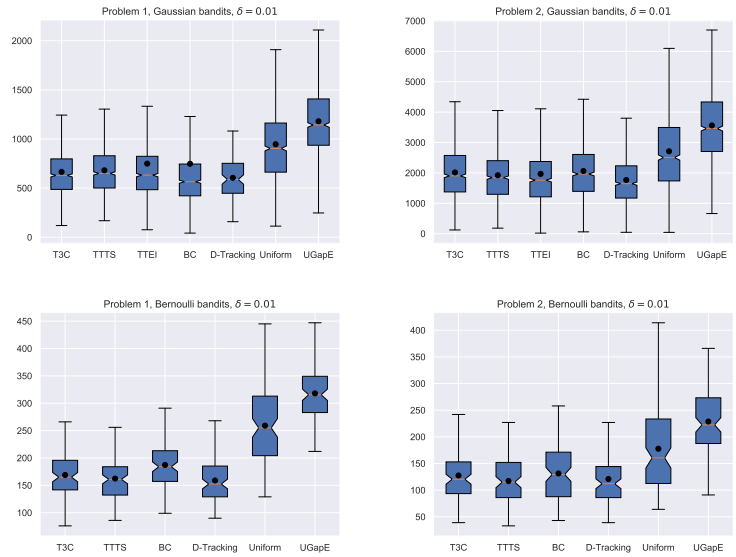


Figure 3.2: Sample complexity of different BAI sampling rules over some random problem instances. Black dots represent means and oranges lines represent medians.

Samp. rule	T3C	TTTS	TTEI	BC	D-T	Uniform	UGapE
Exec. time (s)	1.6×10^{-5}	2.3×10^{-4}	1×10^{-5}	1.4×10^{-5}	1.3×10^{-3}	6×10^{-6}	5×10^{-6}

Table 3.1: Average execution time in seconds for different BAI sampling rules.

3.7 Discussion

We have advocated the use of a Bayesian sampling rule for BAI. In particular, we proved that TTTS and a computationally advantageous approach T3C, are both β-optimal in the fixed-confidence setting, for Gaussian bandits. Our analysis applies to Gaussian bandits, but could be extended to more distributions for which posterior tails bounds are available.

We further extended the Bayesian optimality properties established by Russo [2016] to more practical choices of models and prior distributions.

For future work, it would also be meaningful to provide a fixed-budget analysis, in particular in some potential application scenario of TTTS. For example, pure-exploration bandit algorithms are widely used in HPO [Hoffman et al., 2014; Li et al., 2017] which is the topic of Chapter 6.

Another important unsolved open question comes to the tuning of β. Indeed, if β is set to $\beta^\star = \arg\max_{\beta \in [0,1]} T^\star_\beta(\boldsymbol{\mu})^{-1}$, a β-optimal strategy is also optimal. In practice of course $\beta^\star$ is unknown and so far TTTS cannot be asymptotically optimal without this knowledge. However, note that Russo [2016] shows that $\Gamma^\star_{1/2} \geq \Gamma^\star/2$, which provides a near-optimal tuning of TTTS. Obviously, proposing an satisfying online tuning of β (other than the one proposed in the TTTS paper) with provable fixed-confidence guarantees is another avenue for future work.

There is another line of research that leverages a game theoretic point of view on the pure exploration setting, that explores a statistic-computation trade-off (Degenne et al. 2019; Ménard 2019, see also Chapter 4). It is also interesting to investigate whether TS-based exploration can replace the current (complicated) optimistic approach.

Chapter 4

Optimal Algorithms for Linear Best-Arm Identification

" *Il n'y a pas de hors-texte.*

Jacques Derrida

Contents

4.1 Introduction 46
4.2 Problem Setting and Assumptions 47
4.3 Fixed-Confidence Optimality and Complexities 49
4.3.1 Lower bound 49
4.3.2 Best-arm identification complexity 51
4.3.3 Computation of different complexities 52
4.3.4 Some extensions 54
4.4 Related Work 55
4.5 Bayesian Algorithms for the Linear Case 56
4.5.1 Direct adaptation of TTTS and T3C 56
4.5.2 L-T3S and L-T3C can fail 58
4.5.3 A "greedy" fix of L-T3C 58
4.5.4 LinGapE versus L-T3C-Greedy: Is one of them optimal? 58
4.5.5 Empirical performance of L-T3C-Greedy 59
4.6 A Gamified Algorithm 62
4.6.1 Notation 62
4.6.2 The LinGame algorithm 62
4.6.3 Experiments 66
4.7 Other Saddle-Point Approaches 68
4.7.1 Linear Track-and-Stop 68
4.7.2 Saddle-point Frank-Wolfe 69
4.7.3 Experimental illustrations 70
4.8 Discussion 70

4.1 Introduction

Following the previous chapter, we study a natural extension of the vanilla BAI problem in this chapter, namely linear bandits BAI. As already stated in Chapter 2, linear bandits BAI studies the case where noisy linear payoffs depending on some *unknown* regression parameter $\boldsymbol{\theta}$ are assumed.

Bandits with linear payoffs (or more generally with contextual payoffs) are of great interest in many real-world applications. Typically, we can think of advertisement display optimization where an e-content provider seeks to identify the best-performing advertisement display design. Other relevant applications include recommender systems, path routing, power grid cost minimization, etc. It is arguable whether we need regret minimization or best-arm identification for those situations: a reasonable guess is that it is often subject to the real business needs.

Linear bandits were first investigated by Auer [2002] in the stochastic setting for regret minimization and later considered for BAI problems in the fixed-confidence setting by Soare et al. [2014]. In this chapter, we again focus on the fixed-confidence setting (see Definition 2.6). A quick reminder that for fixed-confidence BAI, we search for algorithms that are able to output the correct best arm with high confidence using as few samples as possible.

Recall that vanilla fixed-confidence BAI problems are often treated by arm eliminations such as `Successive-Elimination` [Karnin et al., 2013], or by confidence-based methods such as `UGapE` [Gabillon et al., 2012]. Those algorithms have naturally been extended to the linear setting (we survey existing methods for linear bandits BAI in Section 4.4). Before the work presented in this chapter, BAI for linear bandits has been previously studied by Fiez et al. [2019]; Kazerouni and Wein [2019]; Soare et al. [2014]; Tao et al. [2018]; Xu et al. [2018]; Zaki et al. [2019]. They all consider the fixed-confidence setting.

Beside studying fixed-confidence sample complexity, Garivier and Kaufmann [2016] and some subsequent works [Qin et al., 2017; Shang et al., 2020a] investigate a general criterion of judging the optimality of a BAI sampling rule: Algorithms that achieve the minimal sample complexity when δ tends to zero are called asymptotically optimal as elaborated on in Chapter 3. Previous work do not seem to satisfy this (asymptotic) optimality rule.

Since then, Ménard [2019] and Degenne et al. [2019] further study the problem from a game theoretical point of view, and extend the asymptotic optimality to the general pure exploration for structured bandits. Note that a naive adaptation of the algorithm proposed by Degenne et al. [2019] may not work smoothly in the linear setting. Algorithms that benefit better from the linear structure are needed.

The primary goal of this chapter is thus to investigate what is the key element that impacts the optimality of an algorithm and how to design (asymptotically) optimal algorithms for linear bandits BAI. A first set of candidates is the Bayesian algorithms presented in Chapter 3: can they be extended in order to achieve optimality in the linear case? Other potential options are considered as well, in particular methods that approach the lower bound step by step (e.g. inspired by Garivier and Kaufmann 2016 or by Degenne et al. [2019]).

Contributions. **1)** We provide new insights on the complexity of BAI for linear bandits. In particular, we relate the asymptotic complexity of the BAI problem and other measures of complexity inspired by optimal design theory, which were used in prior work. **2)** We propose extensions of the Bayesian algorithms studied in Chapter 3 to the linear setting,

and provide empirical evidence that they are not asymptotically optimal. **3)** We develop a saddle-point approach to the lower bound optimization problem, which also guides the design of an algorithm `LinGame` for linear bandits BAI in the fixed-confidence regime. Its sample complexity is asymptotically optimal and its empirical performance is competitive with the best existing algorithms.

☞ This chapter is partly based on some unpublished work, and partly base on Degenne et al. [2020a].

4.2 Problem Setting and Assumptions

In this section, we recall the problem setting of linear bandits as well as linear bandits BAI. In particular, we specify the assumptions used in this chapter.

Linear bandits. We consider a finitely-armed linear bandit problem, where the collection of arms $\mathcal{X} \triangleq \{\mathbf{x}_1, \cdots, \mathbf{x}_K\} \subset \mathbb{R}^d$ is given with $|\mathcal{X}| = K$, and spans $\mathbb{R}^d$. When there is no ambiguity, we can also use the index $i \in [K]$ to represent arm $\mathbf{x}_i$. The (unknown) mean of each arm is given by

$$\mu_i = \mathbf{x}_i^\top \boldsymbol{\theta}\,.$$

Assumption 4.1. *We assume that* $\exists L > 0$, $\forall \mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\| \leq L$, *where* $\|\mathbf{x}\|$ *denotes the Euclidean norm of the vector* $\mathbf{x}$.

The learning protocol (see also Definition 2.11) goes as follows: for each round n, the learner chooses an arm $\widehat{\mathbf{x}}_n \in \mathcal{X}$ and observes a noisy sample

$$r_n = \widehat{\mathbf{x}}_n^\top \boldsymbol{\theta} + \varepsilon_n\,,$$

where ε_n is the noise and $\boldsymbol{\theta}$ is the true regression parameter (unknown to the learner).

Assumption 4.2. *We assume that* $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$ *is conditionally independent from the past*.

For the sake of simplicity, we set $\sigma^2 = 1$ in the rest of this paper.

Best-arm identification for linear bandits. We assume that $\boldsymbol{\theta}$ belongs to some parameter set $\Theta \subset \mathbb{R}^d$ known to the learner. Recall that in a pure exploration game, given a parameter $\boldsymbol{\theta}$, the learner aims to find the correct answer $I^\star(\boldsymbol{\theta}) \in \mathcal{I}$ by interacting with the finite-armed linear bandit environment parameterized by $\boldsymbol{\theta}$ (see also Section 2.3.1).

In particular, we are interested in BAI for which the objective is to identify the arm with the largest mean. That is, the correct answer given $\boldsymbol{\theta}$ is given by

$$I^\star(\boldsymbol{\theta}) = \mathbf{x}^\star(\boldsymbol{\theta}) \triangleq \operatorname*{arg\,max}_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top \boldsymbol{\theta}$$

for $\boldsymbol{\theta} \in \Theta = \mathbb{R}^d$ and the set of possible correct answers is $\mathcal{I} = \mathcal{X}$. When clear from the context, we can simply denote $\mathbf{x}^\star(\boldsymbol{\theta})$ by $\mathbf{x}^\star$.

Algorithm. Let $\mathcal{F}_n = \sigma(\widehat{\mathbf{x}}_1, r_1, \cdots, \widehat{\mathbf{x}}_n, r_n)$ be the information available to the learner after n round. We restate the definition of a BAI algorithm under the fixed-confidence setting, which is given by three components: (1) a *sampling rule* $(\widehat{\mathbf{x}}_n)_{n\geq 1}$, where $\widehat{\mathbf{x}}_n \in \mathcal{X}$ is $\mathcal{F}_{n-1}$-measurable, (2) a *stopping rule* τ_δ, a stopping time for the filtration $(\mathcal{F}_n)_{n\geq 1}$, and (3) a *decision rule* $J_\tau \in \mathcal{X}$ which is $\mathcal{F}_{\tau_\delta}$-measurable.

δ-correctness and fixed-confidence objective. As already stated several times in the previous chapters, we say that an algorithm is δ-correct if it predicts the correct best arm with probability at least $1-\delta$, precisely if $\mathbb{P}_{\boldsymbol{\theta}}\left[\left(\mathbf{x}_{J_\tau} \neq I^\star(\boldsymbol{\theta})\right) \leq \delta\right]$ and $\tau_\delta < +\infty$ almost surely for all $\boldsymbol{\theta} \in \Theta$. Our goal is to find a δ-correct algorithm that minimizes the *sample complexity*, that is, $\mathbb{E}_{\boldsymbol{\theta}}[\tau_\delta]$ the expected number of sample needed to predict an answer.

Linear estimator. A crucial step in linear bandits is to estimate the regression parameter $\boldsymbol{\theta}$. Let $\mathbf{X}_n = (\widehat{\mathbf{x}}_1, \dots, \widehat{\mathbf{x}}_n)$ be a sequence of sampled arms, and $\mathbf{r}_n = (r_1, \dots, r_n)$ be the corresponding observations. To estimate $\boldsymbol{\theta}^\star$ based on the adaptive sequence of observations $\mathbf{r}_n$, one may use the *regularized least-square estimation*

$$\widehat{\boldsymbol{\theta}}_n^\lambda = (\lambda \mathbb{1}_d + \mathbf{A}_{\mathbf{X}_n})^{-1} \mathbf{b}_{\mathbf{X}_n}, \tag{4.1}$$

where $\mathbf{A}_{\mathbf{X}_n}$ and $\mathbf{b}_{\mathbf{X}_n}$ are the design matrix and the response vector respectively given by

$$\mathbf{A}_{\mathbf{X}_n} \triangleq \sum_{t=1}^{n} \widehat{\mathbf{x}}_t \widehat{\mathbf{x}}_t^\mathsf{T}, \quad \mathbf{b}_{\mathbf{X}_n} \triangleq \sum_{t=1}^{n} \widehat{\mathbf{x}}_t r_t,$$

and $\lambda \in \mathbb{R}$ is the regularization parameter. When clear from the context, we can simply denote $\mathbf{A}_{\mathbf{X}_n}$ by $\mathbf{A}_n$ and $\mathbf{b}_{\mathbf{X}_n}$ by $\mathbf{b}_n$.

Useful notation. The fixed-confidence optimality, as proved by Garivier and Kaufmann [2016]; Russo [2016], is related to the *proportion vector* of pulls of each arm that we denote by $\boldsymbol{\omega} = (\omega_1, \dots, \omega_K)$, where $\boldsymbol{\omega} \in \Sigma_K$. Given a vector of proportions $\boldsymbol{\omega}$, we can define a counterpart of the design matrix

$$\boldsymbol{\Lambda}_{\boldsymbol{\omega}} \triangleq \sum_{i=1}^{K} \omega_i \mathbf{x}_i \mathbf{x}_i^\mathsf{T}.$$

It is easy to switch between the design matrix and the proportion vector. Indeed, given a sequence of sampled arms $\mathbf{X}_n$, the corresponding proportion vector can be written as

$$\forall i \in [K], \quad \omega_{n+1,i} = \frac{T_{n+1,i}}{n},$$

where recall that $T_{n,i} \triangleq \sum_{t=1}^{n-1} \mathbb{1}\{\widehat{\mathbf{x}}_t = i\}$ is the number of pulls of arm i before round n. Therefore, the corresponding design matrix can be written as $\mathbf{A}_{\mathbf{X}_n} = n\boldsymbol{\Lambda}_{\boldsymbol{\omega}_{\mathbf{X}_n}}$.

Another important notation that we employ ceaselessly is the Mahalanobis norm which is defined, given a positive semi-definite matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, by

$$\forall \mathbf{x} \in \mathbb{R}^d, \quad \|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}}.$$

4.3 Fixed-Confidence Optimality and Complexities

Our primary goal is to propose a BAI strategy that outputs quickly and reliably a final guess of the best arm. Formally, given a risk level δ, we want to show that

$$\mathbb{P}\left[\mathbf{x}_{J_{\tau_\delta}} \neq \mathbf{x}^\star\right] \leq \delta,$$

while minimizing the expected number of samples $\mathbb{E}[\tau_\delta]$ that is required. To achieve this objective, we first investigate the lower bound of the sample complexity.

4.3.1 Lower bound

In this section we extend the lower bound of Garivier and Kaufmann [2016], to hold for *pure exploration in finitely-armed linear bandit* problems.

Alternative. Recall the notion of *alternative set* which is already defined in Section 2.5.2 for BAI. For the general pure-exploration problem, the definition is given below.

Definition 4.1 (alternative set (pure exploration))**.** *For any answer $i \in \mathscr{I}$ we define the* alternative set *of arm i, denoted by $\neg i$ the set of parameters where the answer i is not correct, i.e.*

$$\neg i \triangleq \{\boldsymbol{\theta} \in \Theta : \ i \neq \mathrm{I}^\star(\boldsymbol{\theta})\}.$$

Lower bound. A general result on the (non-asymptotic) sample-complexity lower bound in the fixed-confidence regime [Garivier and Kaufmann, 2016], which we reviewed in Section 2.5.2, states that for any δ-correct strategy, we have

$$\mathbb{E}[\tau_\delta] \geq \mathrm{T}^\star(\boldsymbol{\theta}) \log(\frac{1}{3\delta}), \tag{4.2}$$

for a given parameter $\boldsymbol{\theta}$ and a given confidence level δ. And the characteristic time $\mathrm{T}^\star(\boldsymbol{\theta})$ is written as

$$\mathrm{T}^\star(\boldsymbol{\theta})^{-1} \triangleq \max_{\boldsymbol{\omega} \in \Sigma_\mathrm{K}} \inf_{\boldsymbol{\theta}' \in \neg \mathrm{I}^\star(\boldsymbol{\theta})} \frac{1}{2} \left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}}, \tag{4.3}$$

and it can be further particularized into (4.4) defined in Proposition 4.1 for linear bandits.

Proposition 4.1. *In the linear case, the quantity $\mathrm{T}^\star(\boldsymbol{\theta})$ is written as*

$$\mathrm{T}^\star(\boldsymbol{\theta}) \triangleq \inf_{\boldsymbol{\omega} \in \Sigma_\mathrm{K}} \max_{\mathbf{x} \neq \mathbf{x}^\star} \frac{2\left\|\mathbf{x}^\star - \mathbf{x}\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}}{(\mathbf{x}^\mathsf{T}\boldsymbol{\theta} - (\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta})^2}. \tag{4.4}$$

Proof. Using the alternative set of $\mathrm{I}^\star(\boldsymbol{\theta})$, and by (4.3), we obtain

$$\begin{aligned}
\mathrm{T}^\star(\boldsymbol{\theta})^{-1} &= \max_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \inf_{\boldsymbol{\theta}'\in\neg\mathrm{I}^\star(\boldsymbol{\theta})} \frac{1}{2}\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}} \\
&= \max_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \inf_{\boldsymbol{\theta}'\in\neg\mathrm{I}^\star(\boldsymbol{\theta})} \sum_{i=1}^{\mathrm{K}} \omega_i d(\mu_i;\mu_i') \\
&= \max_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \min_{\mathbf{x}\neq\mathbf{x}^\star} \inf_{\mathbf{x}^\mathsf{T}\boldsymbol{\theta}'>(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta}'} \frac{\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}}}{2}.
\end{aligned}$$

Then we introduce the Lagrangian with η as the Lagrange multiplier, and it then becomes

$$\mathrm{T}^\star(\boldsymbol{\theta})^{-1} = \sup_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \min_{\mathbf{x}\neq\mathbf{x}^\star} \inf_{\boldsymbol{\theta}'} \sup_{\eta>0} \frac{\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}}}{2} - \eta(\mathbf{x}-\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta}',$$

and the inner expression attains its minimum when it comes

$$\boldsymbol{\Lambda}_\boldsymbol{\omega}(\boldsymbol{\theta}-\boldsymbol{\theta}') = \eta(\mathbf{x}^\star-\mathbf{x}),$$

which implies

$$\begin{aligned}
\mathrm{T}^\star(\boldsymbol{\theta})^{-1} &= \sup_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \min_{\mathbf{x}\neq\mathbf{x}^\star} \sup_{\eta>0} \eta(\mathbf{x}^\star-\mathbf{x})^\mathsf{T}\boldsymbol{\theta} - \frac{\eta^2\left\|\mathbf{x}-\mathbf{x}^\star\right\|_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}}{2\sigma^2} \\
&= \sup_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \min_{\mathbf{x}\neq\mathbf{x}^\star} \frac{\left(\mathbf{x}^\mathsf{T}\boldsymbol{\theta}-(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta}\right)^2}{2\left\|\mathbf{x}^\star-\mathbf{x}\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}}.
\end{aligned}$$

□

Asymptotic optimality. We can define the asymptotic optimality upon $\mathrm{T}^\star(\boldsymbol{\theta})$.

Definition 4.2. *A BAI strategy is called optimal in the fixed-confidence setting if it satisfies*

$$\limsup_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\ln(1/\delta)} \leq \mathrm{T}^\star(\boldsymbol{\theta}).$$

Remark 4.1. *Using the same lower-bound techniques, one can also prove that under any δ-correct strategy satisfying $\mathrm{T}_{n,\mathrm{I}^\star}/n \to \beta$ for a given β,*

$$\liminf_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\ln(1/\delta)} \geq \mathrm{T}^\star_\beta(\boldsymbol{\theta}),$$

where $\mathrm{T}^\star_\beta(\boldsymbol{\theta})$ is defined in the same way as $\mathrm{T}^\star(\boldsymbol{\theta})$, but restricted to the constraint $\omega_{\mathrm{I}^\star} = \beta$,

$$\mathrm{T}^\star_\beta(\boldsymbol{\theta}) \triangleq \inf_{\boldsymbol{\omega}\in\Sigma_\mathrm{K},\omega_{\mathrm{I}^\star}=\beta} \max_{\mathbf{x}\neq\mathbf{x}^\star} \frac{2\sigma^2\left\|\mathbf{x}^\star-\mathbf{x}\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}}{(\mathbf{x}^\mathsf{T}\boldsymbol{\theta}-(\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta})^2}.$$

Essentially, we can recover the β-optimality defined in Chapter 3.

4.3.2 Best-arm identification complexity

The inverse of the characteristic time of Proposition 4.1 can also be written as

$$\mathrm{T}^\star(\boldsymbol{\theta})^{-1} = \max_{\boldsymbol{\omega}\in\Sigma_K} \min_{\mathbf{x}\neq\mathbf{x}^\star} \frac{\left(\mathbf{x}^\mathsf{T}\boldsymbol{\theta} - (\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta}\right)^2}{2\left\|\mathbf{x}^\star - \mathbf{x}\right\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}}$$

for BAI.

Since the characteristic time involves many problem dependent quantities that are unknown to the learner, previous work target loose problem-independent upper bounds on the characteristic time. Soare et al. [2014] (see also Fiez et al. 2019; Tao et al. 2018) introduce the G-complexity (denoted by $\mathcal{XX}$) which coincides with the G-optimal design of experimental design theory (see Pukelsheim 2006) and the $\mathcal{XY}_{\mathtt{dir}}$-complexity (denoted by $\mathcal{XY}_{\mathtt{dir}}$) inspired by the transductive experimental design theory [Yu et al., 2006],

$$\mathcal{XX} = \min_{\boldsymbol{\omega}\in\Sigma_K} \max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}\,,$$
$$\mathcal{XY}_{\mathtt{dir}} = \min_{\boldsymbol{\omega}\in\Sigma_K} \max_{\mathbf{y}\in\mathcal{Y}_{\mathtt{dir}}} \|\mathbf{y}\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}\,,$$

where $\mathcal{Y}_{\mathtt{dir}}$ is the set of directions induced by $\mathcal{X}$:

$$\mathcal{Y}_{\mathtt{dir}} \triangleq \{\mathbf{x}-\mathbf{x}' : (\mathbf{x},\mathbf{x}') \in \mathcal{X}\times\mathcal{X}\}\,.$$

For the G-optimal complexity we seek for a proportion of pulls $\boldsymbol{\omega}$ that explores *uniformly* the means of the arms, since the statistical uncertainty for estimating $\mathbf{x}^\mathsf{T}\boldsymbol{\theta}$ scales roughly with $\|\mathbf{x}\|_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}$. In the $\mathcal{XY}_{\mathtt{dir}}$-complexity we try to estimate *uniformly* all the *directions* $\mathbf{x}-\mathbf{x}'$, while a potentially more plausible quantity to maximize would be the characteristic time itself. For the latter, we try to estimate all the *directions* $\mathbf{x}^\star - \mathbf{x}$ scaled by the squared gaps $(\mathbf{x}^\star - \mathbf{x})^\mathsf{T}\boldsymbol{\theta}$.

The fact that previous works maximize over loose upper bounds on the characteristic time is potentially the main reason that they cannot achieve optimality. We can see later (in Section 4.6) that directly maximizing the weighted gaps would indeed lead to an asymptotically optimal algorithm.

Note that the characteristic time can also be seen as a particular optimal transductive design. Indeed for

$$\mathcal{Y}^\star \triangleq \left\{ \frac{\mathbf{x}^\star(\boldsymbol{\theta}) - \mathbf{x}}{|(\mathbf{x}^\star(\boldsymbol{\theta}) - \mathbf{x})^\mathsf{T}\boldsymbol{\theta}|} : \mathbf{x}\in\mathcal{X}/\{\mathbf{x}^\star(\boldsymbol{\theta})\} \right\},$$

it holds

$$\mathrm{T}^\star(\boldsymbol{\theta}) = 2\mathcal{XY}^\star(\boldsymbol{\theta}) \triangleq 2\min_{\boldsymbol{\omega}\in\Sigma_K} \max_{\mathbf{y}\in\mathcal{Y}^\star(\boldsymbol{\theta})} \|\mathbf{y}\|^2_{\boldsymbol{\Lambda}_\boldsymbol{\omega}^{-1}}\,.$$

Besides, we have the following ordering on the complexities

$$\mathrm{T}^\star(\boldsymbol{\theta}) \leq 2\frac{\mathcal{XY}_{\mathtt{dir}}}{\Delta_{\min}(\boldsymbol{\theta})^2} \leq 8\frac{\mathcal{XX}}{\Delta_{\min}(\boldsymbol{\theta})^2} = \frac{8d}{\Delta_{\min}(\boldsymbol{\theta})^2}\,, \tag{4.5}$$

where $\Delta_{\min} = \min_{\mathbf{x}\neq\mathbf{x}^\star(\boldsymbol{\theta})} (\mathbf{x}^\star(\boldsymbol{\theta})-\mathbf{x})^\mathsf{T}\boldsymbol{\theta}$ and the last equality follows from the Kiefer-Wolfowitz equivalence theorem [Kiefer and Wolfowitz, 1959].

Remark 4.2. *In order to compute all these complexities, it is sufficient to solve the following generic optimal transductive design problem: for $\mathcal{Y}$ a finite set of elements in $\mathbb{R}^d$,*

$$\mathcal{XY}_{dir} = \min_{\boldsymbol{\omega} \in \Sigma_K} \max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{\omega}}^{-1}} .$$

When $\mathcal{Y} = \mathcal{X}$ we can use an algorithm inspired by Frank-Wolfe [Frank and Wolfe, 1956] which possesses convergence guarantees [Ahipasaoglu et al., 2008; Atwood, 1969]. But in the general case, up to our knowledge, there is no algorithm with the same kind of guarantees. Previous works used an heuristic based on a straightforward adaptation of the aforementioned algorithm for general sets $\mathcal{Y}$ but it seems to not converge on particular instances (see Section 4.3.3). We instead propose in the same section an algorithm based on saddle-point Frank-Wolfe algorithm that seems to converge on the different instances we tested.

Empirical evaluation. We use the following problem instance to illustrate how various complexities differ in practice. In this instance, contexts are the canonical basis $\mathbf{x}_1 = \mathbf{e}_1, \mathbf{x}_2 = \mathbf{e}_2, \cdots, \mathbf{x}_d = \mathbf{e}_d$, plus an additional disturbing context

$$\mathbf{x}_{d+1} = (\cos(\alpha), \sin(\alpha), 0, \ldots, 0)^\mathsf{T},$$

and a true regression parameter which is proportional to $\mathbf{e}_1$: $\boldsymbol{\theta}^\star = c\mathbf{e}_1$. This instance is frequently used in the literature of linear bandits BAI (see e.g. Soare et al. 2014; Xu et al. 2018) and is considered as a hard instance to test the performance of linear BAI algorithms. In this problem, the best arm is always $\mathbf{e}_1$, but when the angle α is small, the disturbing context is hard to discriminate from $\mathbf{e}_1$. In this section, we set $d = 2, c = 2, \delta = 0.01$ and $\alpha = 0.1$.

In Table 4.1 we compare the different complexities previously mentioned: the characteristic time $\mathrm{T}^\star(\boldsymbol{\theta})$ and its associated optimal weights $\boldsymbol{\omega}^\star_{\mathcal{X}\mathcal{Y}^\star(\boldsymbol{\theta})}$, the $\mathcal{XY}_{\mathtt{dir}}$-complexity and its associated optimal design $\boldsymbol{\omega}^\star_{\mathcal{X}\mathcal{Y}_{\mathtt{dir}}}$, the G-optimal complexity $\mathcal{XX}$ and its associated optimal design $\boldsymbol{\omega}^\star_{\mathcal{X}\mathcal{X}}$. For each weight vector

$$\boldsymbol{\omega} \in \left\{ \boldsymbol{\omega}^\star_{\mathcal{X}\mathcal{Y}^\star(\boldsymbol{\theta})}, \boldsymbol{\omega}_{\mathcal{X}\mathcal{Y}_{\mathtt{dir}}}, \boldsymbol{\omega}_{\mathcal{X}\mathcal{X}} \right\},$$

we also provide the lower bound $\mathrm{T}_{\boldsymbol{\omega}}$ given by (4.2), i.e.

$$\mathrm{T}_{\boldsymbol{\omega}} = \max_{\mathbf{x} \neq \mathbf{x}^\star(\boldsymbol{\theta})} \frac{\left((\mathbf{x}^\star(\boldsymbol{\theta}) - \mathbf{x})^\mathsf{T}\boldsymbol{\theta}\right)^2}{2\|\mathbf{x}^\star(\boldsymbol{\theta}) - \mathbf{x}\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{\omega}}^{-1}}} \log(1/\delta).$$

In particular we notice that targeting the proportions of pulls $\boldsymbol{\omega}_{\mathcal{X}\mathcal{Y}_{\mathtt{dir}}}, \boldsymbol{\omega}_{\mathcal{X}\mathcal{X}}$ leads to a much larger lower bound than the one obtained with the optimal weights.

4.3.3 Computation of different complexities

As mentioned in Section 4.3.1, computing the solution to a specified optimization problem is required in many existing linear BAI algorithms. We survey some methods that can potentially be useful to handle that issue.

We recall that the three notions of complexity $\mathcal{XX}, \mathcal{XY}_{\mathtt{dir}}, \mathcal{XY}^\star(\boldsymbol{\theta})$ can be written in a unified form,

$$\mathcal{XY} = \min_{\boldsymbol{\omega} \in \Sigma_K} \max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{\omega}}^{-1}}, \tag{4.6}$$

	$\omega^\star_{\mathcal{X}\mathcal{Y}^\star}$	$\omega^\star_{\mathcal{X}\mathcal{Y}_{\mathtt{dir}}}$	$\omega^\star_{\mathcal{X}\mathcal{X}}$
$\mathbf{x}_1$	0.047599	0.499983	0.499983
$\mathbf{x}_2$	0.952354	0.499983	0.499983
$\mathbf{x}_3$	0.000047	0.000033	0.000033
T_ω	369	2882	2882
	$T^\star(\theta)$	$2\mathcal{X}\mathcal{Y}_{\mathtt{dir}}/\Delta^2_{\min}$	$8\mathcal{X}\mathcal{X}/\Delta^2_{\min}$
Complexity	0.124607	32.0469	64.0939

Table 4.1: Optimal weights for various complexities with $\Delta_{\min} = 0.0049958$.

where $\mathcal{Y}$ is the transductive set, i.e. a finite set of elements in $\mathbb{R}^d$. Transductive sets corresponding to different complexity types mentioned in this paper can be found in Table 4.2.

Allocation type	Arm set	Transductive set
(1) $\mathcal{X}\mathcal{X}$-allocation	$\mathcal{X}$	$\mathcal{X}$
(2) $\mathcal{X}\mathcal{Y}_{\mathtt{dir}}$-allocation	$\mathcal{X}$	$\mathcal{Y}_{\mathtt{dir}} = \{\mathbf{x}-\mathbf{x}' : (\mathbf{x},\mathbf{x}') \in \mathcal{X}\times\mathcal{X}\}$
(3) $\mathcal{X}\mathcal{Y}^\star(\boldsymbol{\theta})$-allocation	$\mathcal{X}$	$\mathcal{Y}^\star(\boldsymbol{\theta}) = \left\{(\mathbf{x}^\star(\boldsymbol{\theta})-\mathbf{x})/\left\lvert(\mathbf{x}^\star(\boldsymbol{\theta})-\mathbf{x})^\mathsf{T}\boldsymbol{\theta}\right\rvert : \mathbf{x}\in\mathcal{X}/\{\mathbf{x}^\star(\boldsymbol{\theta})\}\right\}$

Table 4.2: Some examples of different transductive sets.

Frank-Wolfe. We can use a Frank-Wolfe heuristic to compute the optimizer of (4.6) shown in Algorithm 4.1. This heuristic is used for example by Fiez et al. [2019]. Note that it has been proved to have a linear convergence guarantee when $\mathcal{Y} = \mathcal{X}$ [Ahipasaoglu et al., 2008]. It is not clear, however, that the same guarantee holds for other transductive sets.

A simple sanity check to test whether a solver works smoothly is to solve $\mathcal{X}\mathcal{Y}^\star(\boldsymbol{\theta})$ for classical multi-armed bandits (i.e. when $\mathcal{X} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$), for which a solver with guarantee exists (see Garivier et al. 2018). In particular we found instances where Algorithm 4.1 does not converge toward the optimal weights, for example: $\mathcal{X} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}, \theta = (0.9, 0.5, 0.5)$.

Algorithm 4.1 Frank-Wolfe heuristic for computing $\mathcal{X}\mathcal{X}$-design

Input: arm set $\mathcal{X} \subset \mathbb{R}^d$, transductive set $\mathcal{Y} \subset \mathbb{R}^d$, maximum iterations N
Initialize: $\omega \leftarrow (1, 1, \ldots, 1) \in \mathbb{R}^A, \Lambda \leftarrow I_d, t \leftarrow 0$
while $n < N$ **do**
  $\widetilde{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{x}, \mathbf{y}\rangle^2_{\Lambda^{-1}}$
  $\Lambda \leftarrow \Lambda + \widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\mathsf{T}$
  $\omega \leftarrow \frac{n}{n+1}\omega + \frac{1}{n+1}\mathbf{e}_{\widetilde{\mathbf{x}}}$
  $n \leftarrow n+1$
end while
Return ω

We propose a variant of the previous heuristic that takes into account a count for each element in the transductive set $\mathcal{Y}$. The pseudo-code of our method is displayed in Algorithm 4.2. $N \in \mathbb{N}^{|\mathcal{Y}|}$ denotes the vector of counts for all $b \in \mathcal{Y}$. Sanity check on various MAB instances shows the correctness of our heuristic, its convergence guarantee remains for the future work.

Algorithm 4.2 Saddle Frank-Wolfe heuristic for computing generic $\mathcal{XY}$-design

Input: arm set $\mathcal{X} \subset \mathbb{R}^d$, transductive set $\mathcal{Y} \subset \mathbb{R}^d$, maximum iterations N
Initialize: $\boldsymbol{\omega} \leftarrow (1, 1, \dots, 1) \in \mathbb{R}^d, \widetilde{\Lambda} \leftarrow \mathrm{I}_d, \Lambda \leftarrow \mathrm{I}_d, n \leftarrow 0$
while $n < \mathrm{N}$ **do**
  $\widetilde{\mathbf{x}} \in \arg\max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|^2_{\Lambda^{-1}\widetilde{\Lambda}\Lambda^{-1}}$
  $\widetilde{\mathbf{y}} \in \arg\max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|^2_{\Lambda^{-1}}$
  $\Lambda \leftarrow \Lambda + \widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^\mathsf{T}$
  $\widetilde{\Lambda} \leftarrow \widetilde{\Lambda} + \widetilde{\mathbf{y}}\widetilde{\mathbf{y}}^\mathsf{T}$
  $\boldsymbol{\omega} \leftarrow \frac{n}{n+1}\boldsymbol{\omega} + \frac{1}{n+1}\mathbf{e}_{\widetilde{\mathbf{x}}}$
  $n \leftarrow n + 1$
end while
Return $\boldsymbol{\omega}$

Entropic mirror descent. An entropic mirror descent alternative is used by Tao et al. [2018] to compute $\mathcal{XX}$. The entropic mirror descent approach requires the knowldge of the Lipschitz constant of $\log\det \boldsymbol{\Lambda}_{\boldsymbol{\omega}}$. Unfortunately, that Lipschitzness property does not seem to hold. Lu et al. [2018] propose a solution to overcome the Lipschitz issue, but only for $\mathcal{XX}$-design. Whether it still works for general $\mathcal{XY}$-design remains an open question.

4.3.4 Some extensions

In this section, we present two relevant extensions of linear bandits BAI. We introduce the settings from a pure-exploration point of view.

Bounded BAI. One straightforward extension is to consider the *bounded* BAI. In this case, the set of parameters is

$$\Theta \triangleq \{\boldsymbol{\theta} \in \mathbb{R}^d : \ |\arg\max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\mathsf{T}\boldsymbol{\theta}| = 1 \text{ and } \|\boldsymbol{\theta}\| \leq \mathrm{M}\}$$

for some $\mathrm{M} > 0$. The set of possible answers is $\mathcal{I} = \mathcal{X}$ and the correct answer is given by

$$\mathrm{I}^\star(\theta) = \mathbf{x}^\star(\theta) \triangleq \arg\max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\mathsf{T}\boldsymbol{\theta}.$$

This additional assumption reduces the characteristic time to

$$\mathrm{T}^\star(\theta)^{-1} = \max_{\boldsymbol{\omega} \in \Sigma_\mathrm{K}} \min_{\mathbf{x} \neq \mathbf{x}^\star(\boldsymbol{\theta})} \inf_{\substack{(\mathbf{x}-\mathbf{x}^\star)^\mathsf{T}\boldsymbol{\theta}' > 0 \\ \|\boldsymbol{\theta}'\| \leq \mathrm{M}}} \left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{\omega}}}.$$

Transductive BAI. Another very closely-related setting is the transductive BAI [Fiez et al., 2019] where the learner wants to find the best arm of a different set $\mathcal{Y}$ that the one they are allowed to pull. Precisely the set of parameters is

$$\Theta \triangleq \mathbb{R}^d / \{\boldsymbol{\theta} \in \mathbb{R}^d : |\arg\max_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}^\mathsf{T}\boldsymbol{\theta}| > 1\}.$$

The set of possible answers is $\mathcal{I} = \mathcal{Y}$ and the correct answer is given by

$$\mathrm{I}^\star(\boldsymbol{\theta}) = \mathbf{y}^\star(\boldsymbol{\theta}) \triangleq \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathbf{y}^\mathsf{T}\boldsymbol{\theta}.$$

The characteristic time in this case is

$$\mathrm{T}^\star(\boldsymbol{\theta})^{-1} = \max_{\boldsymbol{\omega}\in\Sigma_K} \min_{\mathbf{y}\neq\mathbf{y}^\star(\boldsymbol{\theta})} \frac{(\mathbf{y}^\mathsf{T}\boldsymbol{\theta} - (\mathbf{y}^\star)^\mathsf{T}\boldsymbol{\theta})^2}{2\left\|\mathbf{y}^\star - \mathbf{y}\right\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}}.$$

Note that the dependency on the arm set $\mathcal{X}$ here only appears through the matrix $\boldsymbol{\Lambda}_\boldsymbol{\omega}$.

4.4 Related Work

We survey previous work on linear BAI. The major focus is put on sampling rules in this section. We stress that all the stopping rules employed in the linear BAI literature are equivalent up to the choice of their exploration rate (More discussion in Appendix C.4). As aforementioned, existing sampling rules for fixed-confidence linear BAI are either elimination-based or gap-based. Elimination-based sampling rules usually operate in phases and progressively discard sub-optimal directions. Gap-based sampling rules always play the most informative arm that reduces the uncertainty of the gaps between the empirical best arm and the others.

$\mathcal{XY}$-Static and $\mathcal{XY}$-Adaptive. Soare et al. [2014] first propose a static allocation design $\mathcal{XY}$-Static that aims at reducing the uncertainty of the gaps of all arms. More precisely, it requires to either solve the $\mathcal{XY}_{\mathtt{dir}}$-complexity or use a *greedy* version that pulls the arm

$$\arg\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}_{\mathtt{dir}}} \|\mathbf{y}\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}$$

at the cost of having no guarantees. An elimination-like alternative called $\mathcal{XY}$-Adaptive is proposed then to overcome that issue. We say elimination-like since $\mathcal{XY}$-Adaptive does not discard arms once and for all, but reset the active arm set at each phase. These two algorithms are the first one being linked to $\mathcal{XX}$-optimality, but are not asymptotically optimal.

ALBA. ALBA is also an eliminations-based algorithm designed by Tao et al. [2018] that improves over $\mathcal{XY}$-Adaptive by a factor of d in the sample complexity using a tighter elimination criterion.

RAGE. Fiez et al. [2019] extend $\mathcal{XY}$-Static and $\mathcal{XY}$-Adaptive to a more general transductive bandits setting. RAGE is also elimination-based and requires the computation of $\mathcal{XY}_{\mathtt{dir}}$-complexity at each phase.

LinGapE and variants. LinGapE [Xu et al., 2018] is the first gap-based sampling rule for linear BAI. LinGapE is inspired by UGapE [Gabillon et al., 2012]. It is, however, not clear whether LinGapE is asymptotically optimal or not. Similar to $\mathcal{XY}$-Static, LinGapE either requires to solve a time-consuming optimization problem at each step, or can use a greedy version that pulls arm

$$\arg\min_{\mathbf{x}\in\mathcal{X}} \left\|\mathbf{x}_{i_n} - \mathbf{x}_{j_n}\right\|^2_{(\mathbf{A}_n+\mathbf{x}\mathbf{x}^\mathsf{T})^{-1}}$$

instead, again at the cost of losing guarantees. Here $i_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n)$ and $\widehat{\mathbf{x}}_{j_n}$ is the most ambiguous arm w.r.t. $\widehat{\mathbf{x}}_{i_n}$, i.e.

$$\arg\max_{j\neq i_n} (\mathbf{x}_j - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}^\lambda_n))^\mathsf{T}\widehat{\boldsymbol{\theta}}^\lambda_n + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}^\lambda_n) - \mathbf{x}_{j_n}\right\|_{\mathbf{A}_n^{-1}} \sqrt{2d_{n,\delta}},$$

with $d_{n,\delta}$ the stopping rule threshold.

On the other hand, Zaki et al. [2019] propose a new algorithm based on LUCB. With a careful examination, we note that the sampling rule of GLUCB is equivalent to that of the greedy LinGapE using the Sherman-Morrison formula. Later, Kazerouni and Wein [2019] provide a natural extension of LinGapE to the *generalized linear bandits* setting, where the rewards depend on a strictly increasing *inverse link function*. GLGapE reduces to LinGapE when the inverse link function is the identity function.

Summary. It is worth noting that all the sampling rules presented here depend on δ (except $\mathcal{XY}$-Static), while we aim to design sampling rules that are δ-free which is appealing for applications as argued by Jun and Nowak [2016]. Also all the guarantees in the literature are of the form $C\log(1/\delta) + O\big(\log(1/\delta)\big)$ for a constant C that is strictly larger than $T^\star(\theta)^{-1}$.

In the next, we present a set of algorithms using different design patterns that aim to address linear bandits BAI in a (near) optimal way.

4.5 Bayesian Algorithms for the Linear Case

We first investigate a natural extension of the Bayesian algorithms from Chapter 3 to the linear setting.

4.5.1 Direct adaptation of TTTS and T3C

We consider two Bayesian sampling rules inspired by TTTS and T3C called Linear-Top-Two Thompson Sampling (L-T3S) and Linear-Top-Two Transportation Cost (L-T3C) respectively. Both sampling rules make use of a prior distribution Π_1 over a set of parameters Θ, that contains the unknown true regression parameter $\boldsymbol{\theta}$. Upon observing a sequence of payoffs $(r_1,,\dots,r_{n-1})$, we update our beliefs over the regression parameter and obtain a posterior distribution Π_n whose density w.r.t. the Lebesgue measure is denoted by π_n.

L-T3S/L-T3C differ from TTTS/T3C in the choice of prior distribution and consequently the deduction of posterior distribution. Indeed, in the linear case, we assume that $\boldsymbol{\theta}$ is sampled from $\mathcal{N}(0,\kappa^2\mathbb{1}_d)$ with κ^2 to be precised below. The posterior distribution Π_n, given the sequence of sampled arms $\mathbf{X}_n$, can be written as $\mathcal{N}(\widehat{\boldsymbol{\theta}}_n^\lambda, \widehat{\boldsymbol{\Sigma}}_n)$ with

$$(\widehat{\boldsymbol{\Sigma}}_n)^{-1} = \frac{1}{\kappa^2}\mathbb{1}_d + \frac{1}{\sigma^2}\sum_{t=1}^{n}\widehat{\mathbf{x}}_t\widehat{\mathbf{x}}_t^\top \quad \text{and} \quad \widehat{\boldsymbol{\theta}}_n^\lambda = \frac{1}{\sigma^2}\widehat{\boldsymbol{\Sigma}}_n\mathbf{b}_{\mathbf{X}_n}. \tag{4.7}$$

Combining (4.7) and (4.1), we obtain $\kappa^2 = \sigma^2/\lambda$. One can also write $\widehat{\boldsymbol{\Sigma}}_n = \sigma^2(\mathbf{B}_n^\lambda)^{-1}$ with $\mathbf{B}_n^\lambda = \lambda\mathbb{1}_d + \sum_{t=1}^n \widehat{\mathbf{x}}_t\widehat{\mathbf{x}}_t^\top$.

Description of L-T3S. At each time step n, L-T3S has two potential actions: (1) with probability β, a parameter vector $\boldsymbol{\theta}_1$ is sampled from Π_n, and L-T3S chooses to play $\widehat{\mathbf{x}}_n^{(1)} \triangleq \arg\max_{\mathbf{x}\in\mathcal{X}}\mathbf{x}^\top\boldsymbol{\theta}_1$, whose index is denoted by $I_n^{(1)}$, (2) and with probability $1-\beta$, the algorithm continues sampling new $\boldsymbol{\theta}_2$ until we obtain a *challenger* $\widehat{\mathbf{x}}_n^{(2)} \triangleq \arg\max_{\mathbf{x}\in\mathcal{X}}\mathbf{x}^\top\boldsymbol{\theta}_2$ indexed by $I_n^{(2)}$ that is different from $I_n^{(1)}$, and L-T3S then selects the challenger.

Description of L-T3C. We can also extend T3C, the computational-lightweight variant of TTTS to the linear case which we call L-T3C. Instead of re-sampling from the posterior until a different candidate appears, we define the challenger as the arm that has the lowest *transportation cost* $W_n(I_n^{(1)}, i)$ with respect to the first candidate (with ties broken uniformly at random). The transportation cost is defined as

$$W_n(i,j) = \frac{(\mathbf{x}_i^\top \widehat{\boldsymbol{\theta}}_n^\lambda - \mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda)^2}{2\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_{\widehat{\boldsymbol{\Sigma}}_n}^2} \mathbb{1}\left\{\mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda < \mathbf{x}_i^\top \widehat{\boldsymbol{\theta}}_n^\lambda\right\}. \tag{4.8}$$

The pseudo-code of the two sampling rules are given in Algorithm 4.3.

Algorithm 4.3 Sampling rule of L-T3S/L-T3C

Input: β
for $n \leftarrow 1, 2, \cdots$ **do**
  Sample $\boldsymbol{\theta}_1 \sim \Pi_n$
  $\widehat{\mathbf{x}}^{(1)} \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} \mathbf{x}^\top \boldsymbol{\theta}_1$ (indexed by $I^{(1)}$)
  Sample $b \sim \mathcal{B}er n(\beta)$
  if $b = 1$ **then**
    Evaluate arm $I^{(1)}$
  else
    Repeat sample $\boldsymbol{\theta}_2 \sim \Pi_n$
    $\widehat{\mathbf{x}}^{(2)} \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} \mathbf{x}^\top \boldsymbol{\theta}_2$ (indexed by $I^{(2)}$) ▷ L-T3S
    `until` $I^{(2)} \neq I^{(1)}$
    $I^{(2)} \leftarrow \arg\min_{i\neq I^{(1)}} W_n(I^{(1)}, i)$, (see (4.8) for the definition) ▷ L-T3C
    Evaluate arm $I^{(2)}$
  end if
  Update mean and variance according to (4.1) and (4.7)
  $n = n + 1$
end for

Optimal action probability. The optimal action probability $\alpha_{n,i}$ is defined as the posterior probability that arm i is optimal. Formally, denote Θ_i as the subset of Θ where arm i is the optimal arm, we have

$$\Theta_i \triangleq \left\{\boldsymbol{\theta} \in \Theta \;\middle|\; \mathbf{x}_i^\top \boldsymbol{\theta} > \max_{j\neq i} \mathbf{x}_j^\top \boldsymbol{\theta}\right\},$$

then we define

$$\alpha_{n,i} \triangleq \Pi_n(\Theta_i) = \int_{\Theta_i} \pi_n(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}.$$

Stopping rule and decision rule. As argued in Section 3.4, it is reasonable to use the Chernoff stopping rule formalized by Garivier and Kaufmann [2016] in practice. Using the transportation cost $W_n(i,j)$ defined in (4.8), the Chernoff stopping rule can be written as

$$\tau_\delta \triangleq \inf\left\{n \in \mathbb{N} : \max_{i\in[K]} \min_{j\neq i} W_n(i,j) > d_{n,\delta}\right\}, \tag{4.9}$$

with $d_{n,\delta}$ the threshold to be chosen neatly in practice.

This stopping rule is coupled with the decision rule

$$J_n = \arg\max_j \mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda. \tag{4.10}$$

4.5.2 L-T3S and L-T3C can fail

L-T3S and L-T3C may not be optimal for the linear case actually. To understand that, we run some simulations with the two sampling rules on the problem instance of Section 4.3.2 with $d = 2, c = 2$, and $\alpha = 0.01$. Both algorithms appear to be alternating between sampling $\mathbf{x}_1$ and the disturbing context $\mathbf{x}_2$ and take very long time before stopping. Note that in this instance, it would be more informative to select $\mathbf{x}_2$ a lot in order to lead how to discriminate between $\mathbf{x}_1$ and $\mathbf{x}_3$ (which is what our competitor LinGapE is doing). To explain why this happens, we display in Fig. 4.1 the confidence ellipsoid of the posterior after how many 10000 iterations of L-T3C as a blue dot region. We can see that the confidence region of the posterior is around the axe $x = 2$, thus a vector sampled from the posterior will most of the time have a larger dot product with $\mathbf{x}_1$ and arm $\mathbf{x}_3$, and $\mathbf{x}_2$ will seldom be chosen as the leader or the challenger.

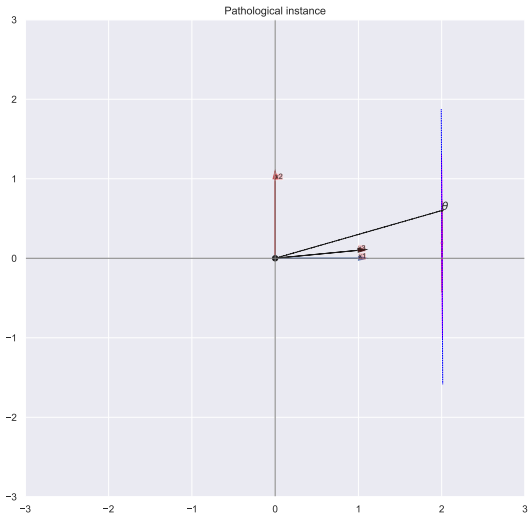


Figure 4.1: A hard problem instance for linear best-arm identification.

4.5.3 A "greedy" fix of L-T3C

A potential fix of the previous issue is the greedy version of L-T3C whose pseudo-code is displayed in Algorithm 4.4. It is inspired by the *greedy* rule already used in (a heuristic version of) LinGapE [Xu et al., 2018], and is motivated by the following observation: in order to learn to discriminate between some arm $\mathrm{I}^{(1)}$ and a challenger $\mathrm{I}^{(2)}$, it may be more informative to select another arm. More specifically, the arm from which a new pull would reduce the most the variance in the estimation of $\mathbf{x}_{\mathrm{I}^{(1)}} - \mathbf{x}_{\mathrm{I}^{(2)}}$. In a standard bandit, this is simply the least pulled arm between $\mathrm{I}^{(1)}$ and $\mathrm{I}^{(2)}$, but in the linear case it may be another arm!

4.5.4 LinGapE versus L-T3C-Greedy: Is one of them optimal?

Upon close examination, LinGapE and L-T3C-Greedy are very similar: they both rely on the computation of a leader and a challenger followed by the greedy rule to decide which

Algorithm 4.4 Sampling rule of L-T3C-Greedy

1: **for** $n \leftarrow 1, 2, \cdots$ **do**
2: Sample $\boldsymbol{\theta}_1 \sim \Pi_n$
3: $\widehat{\mathbf{x}}^{(1)} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top \boldsymbol{\theta}_1$ (indexed by $\mathrm{I}^{(1)}$)
4: $\mathrm{I}^{(2)} \leftarrow \arg\min_{i \neq \mathrm{I}^{(1)}} \mathrm{W}_n(\mathrm{I}^{(1)}, i)$ (see (4.8) for the definition) ▷ $\mathrm{I}^{(2)} \leftarrow \widehat{\mathbf{x}}^{(2)}$
5: Evaluate arm $\widehat{\mathbf{x}} \triangleq \arg\min_{\mathbf{x} \in \mathcal{X}} \left\| \widehat{\mathbf{x}}^{(1)} - \widehat{\mathbf{x}}^{(2)} \right\|_{(\mathbf{A}_{\mathbf{X}_n} + \mathbf{x}\mathbf{x}^\top)^{-1}}$
6: Update mean and variance according to (4.1) and (4.7)
7: $n = n + 1$
8: **end for**

arm to explore/play, and they actually use the same stopping rule (up to possible tuning of the threshold). The differences are:

- how they define the challenger once the leader is chosen;
- how they perform exploration: LinGapE performs exploration in the choice of the challenger, which depends on some confidence bounds. L-T3C-Greedy performs exploration in the choice of the leader, which is determined by Thompson sampling.

Table 4.3 provides a more detailed comparison of the two algorithms for linear bandits. We see in particular that the stopping rule coincide up to the choice $\mathrm{C}_n = \sqrt{2 d_{n,\delta}}$.

	LinGapE	L-T3C-Greedy
Leader	$\mathrm{I}_n^{(1)} = \arg\max_i \mathbf{x}_i^\top \widehat{\boldsymbol{\theta}}_n^\lambda$ with $\widehat{\boldsymbol{\theta}}_n^\lambda$ the least square estimate	$\mathrm{I}_n^{(1)} = \arg\max_i \mathbf{x}_i^\top \widetilde{\boldsymbol{\theta}}_n$ with $\widetilde{\boldsymbol{\theta}}_n \sim \mathcal{N}(\widehat{\boldsymbol{\theta}}_n^\lambda, \widehat{\boldsymbol{\Sigma}}_n)$
Challenger	$\mathrm{I}_n^{(2)} = \arg\max_{j \neq \mathrm{I}_n^{(1)}} \left(\mathbf{x}_j - \mathbf{x}_{\mathrm{I}_n^{(1)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda + \left\| \mathbf{x}_j - \mathbf{x}_{\mathrm{I}_n^{(1)}} \right\|_{\widehat{\boldsymbol{\Sigma}}_n} \mathrm{C}_n$	$\mathrm{I}_n^{(2)} = \arg\min_{j \neq \mathrm{I}_n^{(1)}} \frac{\left(\left(\mathbf{x}_j - \mathbf{x}_{\mathrm{I}_n^{(1)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda\right)^2}{2 \left\| \mathbf{x}_j - \mathbf{x}_{\mathrm{I}_n^{(1)}} \right\|^2_{\widehat{\boldsymbol{\Sigma}}_n}} \mathbb{1}\left\{ \mathbf{x}_{\mathrm{I}_n^{(1)}}^\top \widehat{\boldsymbol{\theta}}_n^\lambda \geq \mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda \right\}$
Stopping	$\left(\mathbf{x}_{\mathrm{I}_n^{(2)}} - \mathbf{x}_{\mathrm{I}_n^{(1)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda + \left\| \mathbf{x}_{\mathrm{I}_n^{(2)}} - \mathbf{x}_{\mathrm{I}_n^{(1)}} \right\|_{\widehat{\boldsymbol{\Sigma}}_n} \mathrm{C}_n < 0$ $\Leftrightarrow \frac{\left(\left(\mathbf{x}_{\mathrm{I}_n^{(1)}} - \mathbf{x}_{\mathrm{I}_n^{(2)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda\right)^2}{2 \left\| \mathbf{x}_{\mathrm{I}_n^{(2)}} - \mathbf{x}_{\mathrm{I}_n^{(1)}} \right\|^2_{\widehat{\boldsymbol{\Sigma}}_n}} > \mathrm{C}_n^2/2$	$\min_{j \neq \mathrm{J}_n^{(1)}} \frac{\left(\left(\mathbf{x}_j - \mathbf{x}_{\mathrm{J}_n^{(1)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda\right)^2}{2 \left\| \mathbf{x}_j - \mathbf{x}_{\mathrm{J}_n^{(1)}} \right\|^2_{\widehat{\boldsymbol{\Sigma}}_n}} \mathbb{1}\left\{ \mathbf{x}_{\mathrm{J}_n^{(1)}}^\top \widehat{\boldsymbol{\theta}}_n^\lambda \geq \mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda \right\} > d_{n,\delta}$ $\Leftrightarrow \mathrm{J}_n^{(1)} = \arg\max_j \mathbf{x}_j^\top \widehat{\boldsymbol{\theta}}_n^\lambda, \frac{\left(\left(\mathbf{x}_{\mathrm{J}_n^{(1)}} - \mathbf{x}_{\mathrm{J}_n^{(2)}}\right)^\top \widehat{\boldsymbol{\theta}}_n^\lambda\right)^2}{2 \left\| \mathbf{x}_{\mathrm{J}_n^{(2)}} - \mathbf{x}_{\mathrm{J}_n^{(1)}} \right\|^2_{\widehat{\boldsymbol{\Sigma}}_n}} > d_{n,\delta}$

Table 4.3: Comparison between the two algorithms.

Experiments for classical bandits. L-T3C-Greedy can be particularized to the classic BAI setting (that corresponds to choosing the canonical basis as contexts). In that case, the selection rule in Line 6 of Algorithm 4.4 corresponds to choosing the least pulled arm between the two candidates.

We investigate whether L-T3C-Greedy could be optimal in the classical bandit setting with some experiments. In particular, we call the derived algorithm T3C-Greedy. See Fig. 4.2 for a comparison against the asymptotically optimal D-Tracking rule and the oracle. It is seemingly that L-T3C-Greedy is not very promising for being asymptotically optimal.

4.5.5 Empirical performance of L-T3C-Greedy

The usual hard instance. We compare the performance of L-T3C-Greedy to LinGapE over the aforementioned hard instance with $d = 2$, $c = 2$ and two values of α. More pre-

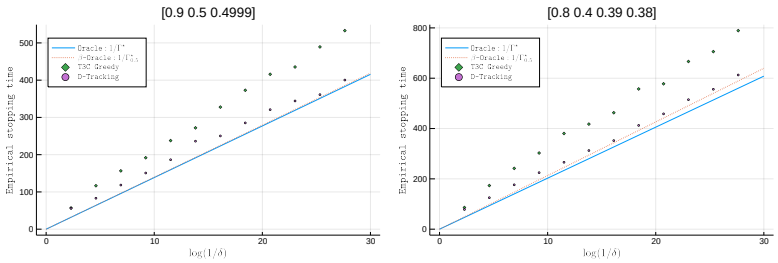


Figure 4.2: T3C-Greedy vs. Track-and-Stop for different value of δ.

cisely, we report in Table 4.4 the total number of pulls and also the number of pulls allocated to each arm for each of the sampling rules. The results confirm our intuition in the previous section.

	L-T3C-Greedy	LinGapE
$\mathbf{x}_1 = (1,0)^\mathsf{T}$	1783.1	5844.4
$\mathbf{x}_2 = (0,1)^\mathsf{T}$	357009.0	1169260.5
$\mathbf{x}_3 = (\cos(0.01), \sin(0.01)^\mathsf{T})$	1.0	1.0
Total	**358793.1**	1175105.9

Table 4.4: Average number of pulls of each arm ($d = 2, \delta = 0.1$).

	L-T3C ($\beta = 1/2$)	L-T3C-Greedy	LinGapE
$\mathbf{x}_1 = (1,0)^\mathsf{T}$	131.3	24.9	26.3
$\mathbf{x}_2 = (0,1)^\mathsf{T}$	3.3	60.8	63.4
$\mathbf{x}_3 = (\cos(\pi/4), \sin(\pi/4)^\mathsf{T})$	133.4	1.0	1.0
Total	268.0	**86.7**	89.7

Table 4.5: Average number of pulls of each arm ($d = 2, \delta = 0.1$).

Arms with mild gaps. We can also provide more experimental illustrations on different types of problem.

We construct a set of K arms proposed by Fiez et al. [2019]:

$$\mathbf{x}_1 = (1,0)^\mathsf{T}, \mathbf{x}_2 = (\cos(3\pi/4), \sin(3\pi/4))^\mathsf{T},$$

and for $k = 3, \ldots, K$,

$$\mathbf{x}_k = (\sin(\pi/4 + \phi_k), \cos(\pi/4 + \phi_k))^\mathsf{T}$$

where $\phi_i \sim \mathcal{N}(0, 0.09)$. We fix the true regression parameter θ to be $\mathbf{e}_1$. This set of arms has some nice properties, that $\mathbf{x}_1$ is the optimal arm, and $\mathbf{x}_2$ is the arm that gives more information on identifying the best arm. We first report results with a moderate K = 6 as an example where the generated expected means are $[1.0, -0.71, 0.84, -0.95, 0.93, 0.99]$.

Impact of the dimension. We also compare the impact of the dimension d over the performance of our sampling rule and LinGapE. We run experiments on the same instance with a value of the angle set to $\alpha = 0.1$. This time we let the dimension d varying from 2 to 6. L-T3C is always better and the performance gap increases with the dimension. Thus, our algorithm seems to be more robust to the dimension.

	L-T3C	L-T3C-Greedy	LinGapE
$\mathbf{x}_1 = (1,0)^\mathsf{T}$	170872.19	1.06	1.13
$\mathbf{x}_2 = (\cos(3\pi/4), \sin(3\pi/4))^\mathsf{T}$	1.25	1.0	1.0
$\mathbf{x}_3 = (\sin(\pi/4+\phi_3), \cos(\pi/4+\phi_3))^\mathsf{T}$	92.18	28693.26	120657.6
$\mathbf{x}_4 = (\sin(\pi/4+\phi_4), \cos(\pi/4+\phi_4))^\mathsf{T}$	1.12	26197.03	110157.63
$\mathbf{x}_5 = (\sin(\pi/4+\phi_5), \cos(\pi/4+\phi_5))^\mathsf{T}$	77.04	1.0	1.0
$\mathbf{x}_6 = (\sin(\pi/4+\phi_6), \cos(\pi/4+\phi_6))^\mathsf{T}$	170892.82	1.36	1.83
Total	341936.6	**54894.71**	230820.19

Table 4.6: average number of pulls of each arm ($d = 2, \delta = 0.1$).

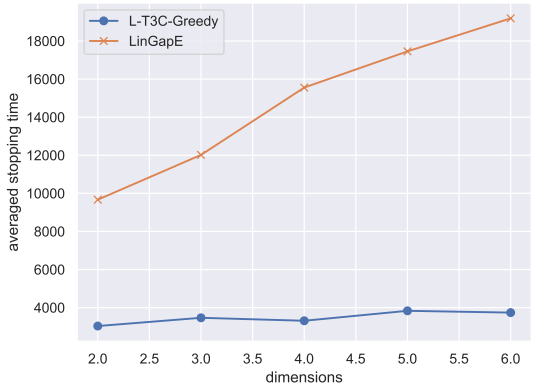


Figure 4.3: Comparison of the sample complexity of L-T3C-Greedy and LinGapE on the pathological instance in $\mathbb{R}^d$ for different values of d.

Rank-1 multivariate normal distribution sampling. A technical point for the experiments in this section is to sample from a multivariate normal distribution (e.g. for L-T3C). We can proceed by the following.

Indeed, a random Gaussian vector

$$\mathbf{X} = (X_1, X_2, \ldots, X_d)^\mathsf{T}$$

of mean vector $\overline{\boldsymbol{\mu}}$ and covariance matrix $\boldsymbol{\Sigma}$ can be formally defined as following,

$$\mathbf{X} \sim \mathcal{N}(\overline{\boldsymbol{\mu}}, \boldsymbol{\Sigma}) \iff \exists \overline{\boldsymbol{\mu}} \in \mathbb{R}^d, \mathbf{A} \in \mathbb{R}^{d \times d'} \text{ s.t. } \mathbf{X} = \mathbf{A}\mathbf{Z} + \overline{\boldsymbol{\mu}},$$

for $Z_i \sim \mathcal{N}(0,1)$ i.i.d. with $i \in \{1, \ldots, d'\}$, and here $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\mathsf{T}$.

To draw a sample from a multivariate normal distribution, according to the previous definition, one can first find any real matrix $\mathbf{A}$ such that $\mathbf{A}\mathbf{A}^\mathsf{T} = \boldsymbol{\Sigma}$. Then draw a vector $\mathbf{Z}$ whose components independently follow standard normal distribution. Finally $\mathbf{X} = \mathbf{A}\mathbf{Z} + \overline{\boldsymbol{\mu}}$ forms a valid sample. The main issue is thus how to find an appropriate matrix $\mathbf{A}$.

In this section, we need to sample from $\mathcal{N}(\widehat{\boldsymbol{\theta}}_n^\lambda, \widehat{\boldsymbol{\Sigma}}_n)$, where the covariance matrix $\widehat{\boldsymbol{\Sigma}}_n$ is a positive-definite matrix. A usual way is to apply the Cholesky decomposition, which is computationally inefficient if it were to be applied at each time step. Fortunately, we can apply rank-1 Cholesky decomposition in our case.

4.6 A Gamified Algorithm

The first attempt using Bayesian machinery does not end up with a satisfying output. In this section, we turn our thoughts to another idea and take inspiration from the *zero-sum game* (see e.g. Degenne and Koolen 2019). We describe Linear Game (LinGame), detailed in Algorithm 4.5. As noted in the seminal work of Chernoff [1959], the complexity $T^\star(\boldsymbol{\theta})^{-1}$ is the value of a fictitious zero-sum game between the learner choosing an optimal proportion of allocation of pulls $\boldsymbol{\omega}$ and a second player, the nature, that tries to fool the agent by responding with the most confusing alternative $\boldsymbol{\theta}'$ leading to an incorrect answer. LinGame is an asymptotically optimal algorithm for linear bandits BAI. Note that the present algorithm is asymptotically optimal for the general pure exploration game, which is however not the main focus of this thesis. Lecturers can refer to Degenne et al. [2020a] for more details.

In this section, we make the following extra assumption.

Assumption 4.3. *We assume that* $\exists M > 0$, *s.t.* $\forall \boldsymbol{\theta} \in \Theta$, $\|\boldsymbol{\theta}\| \leq M$, *where* $\|\boldsymbol{\theta}\|$ *denotes the Euclidean norm of the vector* $\boldsymbol{\theta}$.

4.6.1 Notation

In this section, besides the usual learner, we include an extra fictive player – the nature – and we thus introduce some specific notation of counts for this section.

At each round n the algorithm to be presented will play an arm $\widehat{\mathbf{x}}_n$ and choose (fictitiously) an answer $i_n \in \mathscr{I}$. We denote by $T_n^{\mathbf{x},i} \triangleq \sum_{t=1}^n \mathbb{1}_{\{(\widehat{\mathbf{x}}_n, i_n)=(\mathbf{x},i)\}}$ the number of times the pair $(\mathbf{x}, i) \in \mathscr{X} \times \mathscr{I}$ is chosen up to and including time n, and by $T_n^{\mathbf{x}} = \sum_{i\in\mathscr{I}} T_n^{\mathbf{x},i}$ and $T_n^i = \sum_{\mathbf{x}\in\mathscr{X}} T_n^{\mathbf{x},i}$ the partial sums[1]. The vectors of counts at time n is denoted by $\mathbf{T}_n \triangleq (T_n^{\mathbf{x}})_{\mathbf{x}\in\mathscr{X}}$ and when it is clear from the context we will also denote by $\mathbf{T}_n^{\mathbf{x}} = (T_n^{\mathbf{x},i})_{i\in\mathscr{I}}$ and $\mathbf{T}_n^i = (T_n^{\mathbf{x},i})_{\mathbf{x}\in\mathscr{X}}$ the vectors of partial counts. Recall that in the case of BAI, $\mathscr{X} = \mathscr{I}$.

4.6.2 The LinGame algorithm

The pseudo-code of LinGame is provided in Algorithm 4.5. We explain how it works in detail in the next.

Stopping rule and decision rule. We follow the same stopping rule and decision rule as those of Section 4.5[2]: LinGame stops if a generalized likelihood ratio exceeds a threshold $d_{n,\delta}$. With the notation of this section, the stopping time can be written as

$$\max_{i\in\mathscr{I}} \inf_{\boldsymbol{\theta}'\in\neg i} \frac{1}{2}\|\widehat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\theta}'\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n}} > d_{n,\delta}\,, \tag{4.11}$$

and the decision rule is

$$J_n = \operatorname*{arg\,max}_{i\in\mathscr{I}} \inf_{\boldsymbol{\theta}'\in\neg i} \|\widehat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\theta}'\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n}}/2\,. \tag{4.12}$$

[1] Note that if i is the index of arm $\mathbf{x}$, then $T_n^{\mathbf{x}}$ is simply $T_{n,i}$ as defined in (2.1).

[2] It is easy to check that (4.11) and (4.12) are equivalent to (4.9) and (4.10) respectively.

Algorithm 4.5 Algorithm of LinGame

Input: Learners for each answer $(\mathscr{L}_{\boldsymbol{\omega}}^{i})_{i\in\mathscr{I}}$, threshold d
for n = 1 ... **do**
  // Stopping rule
  if $\max_{i\in\mathscr{I}} \inf_{\boldsymbol{\theta}'\in\neg i} \frac{1}{2}\left\|\widehat{\boldsymbol{\theta}}_{n-1}^{\lambda}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_{n-1}}} \geq d_{n-1,\delta}$ **then**
    Stop
    Return $\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_{n-1}^{\lambda})$
  end if
  // Empirical best guess
  $i_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_{n-1}^{\lambda})$
  // Learner plays first
  Get $\boldsymbol{\omega}_n$ from $\mathscr{L}_{\boldsymbol{\omega}}^{i_n}$
  Update $\mathbf{W}_n = \mathbf{W}_{n-1} + \boldsymbol{\omega}_n$
  // Best response of the nature
  $\boldsymbol{\theta}_n^{i_n} \in \arg\min_{\boldsymbol{\theta}'\in\neg i_n}\left\|\widehat{\boldsymbol{\theta}}_{n-1}^{\lambda}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{\omega}_n}}$
  // Feed optimistic gains
  Feed learner $\mathscr{L}_{\boldsymbol{\omega}}^{i_n}$ with $g_n(\boldsymbol{\omega}) = \sum_{\mathbf{x}\in\mathscr{X}} \omega_{\mathbf{x}} \mathrm{U}_n^{\mathbf{x},i_n}/2$
  // Track the weights
  Pull $\widehat{\mathbf{x}}_n \in \arg\min_{\mathbf{x}\in\mathscr{X}} \mathrm{T}_{n-1}^{\mathbf{x}} - \mathrm{W}_{n,\mathbf{x}}$
end for

Similarly to T3C/TTTS of Chapter 3, these stopping and decision rules ensure that the LinGame is δ-correct regardless of the sampling rule used, see lemma below[3] proved in Appendix C.2.

Lemma 4.1. *Regardless of the sampling rule, the stopping rule* (4.11) *with the threshold*

$$d_{n,\delta} = \left(\sqrt{\log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1+\frac{n\mathrm{L}^2}{\lambda d}\right)} + \sqrt{\frac{\lambda}{2}}\mathrm{M}\right)^2, \tag{4.13}$$

satisfy

$$\mathbb{P}_{\boldsymbol{\theta}}\left[\left(\tau_\delta < \infty \wedge \mathrm{J}_\tau \neq \mathrm{I}^\star(\boldsymbol{\theta})\right)\right] \leq \delta.$$

Our contribution is a sampling rule that minimizes the sample complexity when combined with these stopping and decision rules. We now explain our sampling strategy to ensure that the stopping threshold is reached as soon as possible.

Saddle-point computation. Suppose in this paragraph, for simplicity, that the regression parameter $\boldsymbol{\theta}$ is known to the learner. By the definition of stopping rule and generalized likelihood ratio, as long as the algorithm does not stop, we have

$$d_{n,\delta} \geq \inf_{\boldsymbol{\theta}'\in\neg\mathrm{I}^\star(\boldsymbol{\theta})} \sum_{\mathbf{x}\in\mathscr{X}} \mathrm{T}_n^{\mathbf{x}} \|\boldsymbol{\theta}-\boldsymbol{\theta}'\|^2_{\mathbf{x}\mathbf{x}^\intercal}/2.$$

[3] The fact that $\tau_\delta < +\infty$ is a consequence of our analysis, see Appendix C.3.

Now, let $\boldsymbol{\omega}^\star(\boldsymbol{\theta})$ be the optimal pulling proportions given $\boldsymbol{\theta}$. If we manage to have $\mathbf{T}_n \approx n\boldsymbol{\omega}^\star(\boldsymbol{\theta})$, then it follows $d_{n,\delta} \geq n\mathrm{T}^\star(\boldsymbol{\theta})^{-1}$ and, solving that equation would lead to the asymptotic optimality.

At each time step, `LinGame` produces a guess i_n for $\mathrm{I}^\star(\boldsymbol{\theta})$ and its analysis involves proving that the guess is wrong only finitely-many times in expectation.

The sampling rule implements a lower-bound game between a learner, playing at each stage n a pull-proportion/weight vector $\boldsymbol{\omega}_n$ in the probability simplex Σ_K, and nature, who computes at each stage a response $\boldsymbol{\theta}_n \in \neg i_n$. We additionally ensure that $\mathrm{T}_n^{\mathbf{x}} \approx \sum_{t=1}^n \omega_{t,\mathbf{x}}$, where $\omega_{n,\mathbf{x}}$ denotes the weight of $\mathbf{x}$ at stage n. The goal of the sampling rule is to ensure a ε-approximation of the saddle point of the lower-bound game.

To achieve that, we implement the saddle-point algorithm by using `AdaHedge` for the learner – a regret-minimizing algorithm of the exponential family – and using best-response for the nature, which plays after the agent. Precisely `LinGame` uses $|\mathscr{I}|$ (= K for BAI) learners $\mathscr{L}^i_{\boldsymbol{\omega}}$, one for each possible guess of $\mathrm{I}^\star(\boldsymbol{\theta})$ with the gains. For $i \in \mathscr{I}$, the learner $\mathscr{L}^i_{\boldsymbol{\omega}}$ is an `AdaHedge` on the probability simplex Σ_K with the gains (when the guess is i)

$$g_n^{\boldsymbol{\theta}}(\boldsymbol{\omega}) = \frac{1}{2}\sum_{\mathbf{x}\in\mathscr{X}} \omega_{\mathbf{x}} \|\boldsymbol{\theta} - \boldsymbol{\theta}_n^i\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}}\,.$$

ε is then the sum of the regrets of the two players. Best-response has regret 0, while the regret of `AdaHedge` is $\mathrm{O}(\sqrt{n})$ for bounded gains, as seen in the following lemma, taken from de Rooij et al. [2014].

Lemma 4.2. *On the online learning problem with* K *arms and gains* $g_t(\boldsymbol{\omega}) = \sum_{k\in[\mathrm{K}]} \omega_k \mathrm{U}_t^k$ *for* $t \in [n]$, *`AdaHedge`, predicting* $(\boldsymbol{\omega}_t)_{t\in[n]}$, *has regret*

$$\begin{aligned}\mathrm{R}_n &\triangleq \max_{\boldsymbol{\omega}\in\Sigma_\mathrm{K}} \sum_{t=1}^n g_t(\boldsymbol{\omega}) - g_t(\boldsymbol{\omega}_t) \\ &\leq 2\eta\sqrt{n\log(\mathrm{K})} + 16\eta(2+\log(\mathrm{K})/3)\,,\end{aligned}$$

where $\eta = \max_{t\leq n}(\max_{k\in[\mathrm{K}]} \mathrm{U}_t^k - \min_{k\in[\mathrm{K}]} \mathrm{U}_t^k)$.

Other combinations of learners are possible, as long as the sum of their regrets is sufficiently small. At each stage $n \in \mathbb{N}$, both learners advance only by one iteration and as time progresses, the quality of the saddle-point approximation improves. This is in contrast with `C-Tracking` and `D-Tracking` of Garivier and Kaufmann [2016], in which an exact saddle point is computed at each stage, at a potentially much greater computational cost.

Optimism. The above saddle-point argument would be correct for a known game, while our algorithm is confronted to a game depending on the unknown parameter $\boldsymbol{\theta}$. Following a long tradition of stochastic bandit algorithms, we use the principle of OFU. Given an estimate $\widehat{\boldsymbol{\theta}}_{n-1}$, we compute upper bounds for the gain of the real learner at $\boldsymbol{\theta}$, and feed these optimistic gains to them. Precisely, given the best response $\boldsymbol{\theta}_n^i \in \neg i$, we define,

$$\mathrm{U}_n^{\mathbf{x},i} = \left\{\begin{array}{ll} \max_\xi & \min\left(\|\boldsymbol{\xi} - \boldsymbol{\theta}_n^i\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}}, 4\mathrm{L}^2\mathrm{M}^2\right) \\ \text{s.t.} & \|\widehat{\boldsymbol{\theta}}^\lambda_{n-1} - \boldsymbol{\xi}\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_{n-1}}+\lambda \mathrm{I}_d} \leq 2h(n) \end{array}\right.,$$

where $h(n) = \beta(n, 1/n^3)$ is some exploration function.

We clipped the values, using Assumption 4.1 and Assumption 4.3 to ensure bounded gains for the learners (see Section 4.3.4 for description of bounded BAI).

Under the event that the true parameter verifies $\|\widehat{\boldsymbol{\theta}}^{\lambda}_{n-1} - \boldsymbol{\theta}\|^2 \leq 2h(n)$, this is indeed an optimistic estimate of $\|\boldsymbol{\theta} - \boldsymbol{\theta}^i_n\|^2_{\mathbf{x}\mathbf{x}^\intercal}$. Note that $\mathrm{U}^{\mathbf{x},i}_n$ has a closed form expression, see Appendix C.3. The optimistic gain is then

$$g_n(\boldsymbol{\omega}) = \frac{1}{2}\sum_{\mathbf{x}\in\mathscr{X}} \omega_{\mathbf{x}} \mathrm{U}^{\mathbf{x},i_n}_n .$$

Tracking. In Algorithm 4.5, the learner plays weight vectors in a simplex. Since the bandit procedure allows only to pull one arm at each stage, our algorithm needs a procedure to transcribe weights into pulls. This is what we call tracking, following Garivier and Kaufmann [2016]. The choice of arm (or arm and answer) is

$$\widehat{\mathbf{x}}_{n+1} \in \operatorname*{arg\,min}_{\mathbf{x}\in\mathscr{X}} \mathrm{T}^{\mathbf{x}}_n - \mathrm{W}_{n+1,\mathbf{x}} .$$

This procedure guarantees that for all $n \in \mathbb{N}, \mathbf{x} \in \mathscr{X}$, we have $-\log(|\mathscr{X}|) \leq \mathrm{T}^{\mathbf{x}}_n - \mathrm{W}_{n,\mathbf{x}} \leq 1$. This result is due to Degenne et al. [2020b].

Theorem 4.3. *For a regularization parameter*[4] $\lambda \geq 2(1+\log(\mathrm{K}))\mathrm{K}\mathrm{L}^2 + \mathrm{M}^2$*, for the threshold* $d_{n,\delta}$ *given by* (4.13)*, for an exploration function* $h(n) = d_{n,1/n^3}$*,* `LinGame` *is* δ*-correct and asymptotically optimal. That is, it verifies for all* $\boldsymbol{\theta} \in \Theta$*,*

$$\limsup_{\delta\to 0} \frac{\mathbb{E}_{\boldsymbol{\theta}}[\tau_\delta]}{\log 1/\delta} \leq \mathrm{T}^\star(\boldsymbol{\theta}) .$$

On the boundedness assumption. The boundedness assumption on the parameter set is shared by many works on linear bandits (not necessarily for BAI, but for regret minimization as well, see e.g. Abbasi-Yadkori et al. 2011; Soare et al. 2014).

In Section 4.3.4 we show that adding a bound constraint on the parameter reduces the characteristic time $\mathrm{T}^\star(\boldsymbol{\theta})$. This is not surprising since we add a new constraint in the optimization problem, which would lead to an earlier stop of the algorithm. The counterpart of this improvement is that it is often difficult to compute the best response for nature. Indeed, for example, in BAI, there is an explicit expression of the best response, While it is not the case for the bounded case and one needs to solve an uni-dimensional optimization problem (see Lemma C.1). To devise an asymptotically optimal algorithm without the boundedness assumption remains an open problem.

A convexified variant. Degenne et al. [2020a] present another sampling rule `LinGame-C` that is also asymptotically optimal in the fixed-confidence regime. The idea is to introduce a convex formulation of the problem, which leads to an algorithm with a more direct analysis than previous lower-bound inspired methods. The full description and analysis of `LinGame-C` is omitted as the primary goal here is just to show a feasible way of designing optimal sampling rules from a game-theoretical point of view. Lecturers can refer to Degenne et al. [2020a] for more details on `LinGame-C`. Nonetheless, we still include `LinGame-C` in the coming experiments.

4.6.3 Experiments

We provide experimental illustrations of LinGame. In addition to our algorithms, we also implement the following algorithms, all using the same stopping rule (more discussion given in Appendix C.4): uniform sampling, the greedy version of $\mathcal{XY}$-Static (including $\mathcal{XX}$-allocation and $\mathcal{XY}_{\texttt{dir}}$-allocation), $\mathcal{XY}$-Adaptive, and the greedy version of LinGapE. We skip GLUCB/GLGapE since they are more or less equivalent to LinGapE in the scope of this paper.

Implementation details. We give some details about each individual algorithm implemented to ensure reproducibility.

- For our algorithms LinGame and LinGame-C, we implemented the version with the boundedness assumption.
- For LinGapE We implemented the greedy version, that is, pull the arm

$$\arg\min_{\mathbf{x}\in\mathcal{X}} \left\|\mathbf{x}_{i_n} - \mathbf{x}_{j_n}\right\|^2_{(\mathbf{\Lambda}_{\mathbf{T}_n}+\mathbf{x}\mathbf{x}^\mathsf{T})^{-1}}$$

  with $i_t = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}^\lambda_n)$ and

$$j_n = \arg\max_{j\neq i_n} (\mathbf{x}_j - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}^\lambda_n))^\mathsf{T}\widehat{\boldsymbol{\theta}}^\lambda_n + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}^\lambda_n) - \mathbf{x}_{j_n}\right\|_{\mathbf{\Lambda}^{-1}_{\mathbf{T}_n}} \sqrt{2d_{n,\delta}}\,.$$

  Note that this version does not have a theoretical guarantee in the general case. However, as we stated in Section 4.4, the GLUCB proposed by Zaki et al. [2019] is equivalent to this greedy version of LinGapE, and they provided an analysis for the 2-arm and 3-arm case. LinGapE is designed for ε-best-arm identification, we set $\varepsilon = 0$ in our experiments to make sure that it outputs the optimal one.
- For $\mathcal{XY}$-Static, we implemented the greedy incremental version for both $\mathcal{XX}$-allocation and $\mathcal{XY}_{\texttt{dir}}$-allocation, that allows us to avoid the step of computing optimal design. To implement the non-greedy version, readers are invited to look at next Section 4.3.3 where we discuss in detail the computation of $\mathcal{XY}$-optimal design.
- For $\mathcal{XY}$-Adaptive, it requires a hyper-parameter that characterizes the length of each phase. We set that hyper-parameter to 0.1 as done by Soare et al. [2014].

Implementation trick. Matrix inversion is a costly step that should be avoided at best. For linear bandits, in particular, we need to inverse the (regularized) design matrix $\mathbf{B}^\lambda_n$, which is renewed with a rank-1 update at each time step. Applying Sherman-Morrison formula allows us thus to only update its inverse incrementally, that releases a huge burden of computation.

Indeed, beginning with $\mathbf{B}^\lambda_0 \triangleq \lambda \mathbb{1}_d$, we have

$$\forall\, t \geq 0, \quad \mathbf{B}^\lambda_{t+1} = \mathbf{B}^\lambda_t + \widehat{\mathbf{x}}_{t+1}\widehat{\mathbf{x}}^\mathsf{T}_{t+1},$$

thus using Sherman-Morrison formula we have

$$\forall\, t \geq 0, \quad (\mathbf{B}^\lambda_{t+1})^{-1} = (\mathbf{B}^\lambda_t)^{-1} - \frac{(\mathbf{B}^\lambda_t)^{-1}\widehat{\mathbf{x}}_{t+1}\widehat{\mathbf{x}}^\mathsf{T}_{t+1}(\mathbf{B}^\lambda_t)^{-1}}{1 + \|\widehat{\mathbf{x}}_{t+1}\|^2_{(\mathbf{B}^\lambda_t)^{-1}}}.$$

The posterior mean vector and covariance matrix can then be easily expressed in terms of $(\mathbf{B}^\lambda_t)^{-1}$. Let $\mathbf{z}_t \triangleq \sum_{s=1}^t y_s \widehat{\mathbf{x}}_s$, we obtain

$$\widehat{\boldsymbol{\theta}}^\lambda_n = (\mathbf{B}^\lambda_t)^{-1}\mathbf{z}_t \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}}_n = \sigma^2(\mathbf{B}^\lambda_t)^{-1}.$$

Experimental results. Sampling rules for classical BAI without any adaptation may not work for the linear case. This can be understood, again, on the well-studied hard instance mentioned in Section 4.3.2, which encapsulates the difficulty of BAI in a linear bandit, and thus is the first instance on which we test our algorithms.

As already argued by Soare et al. [2014], an efficient sampling rule for this problem instance would rather pull $\mathbf{x}_2$ in order to reduce the uncertainty in the direction $\mathbf{x}_1 - \mathbf{x}_{d+1}$. Naive application of classical BAI algorithms cannot deal with that situation naturally. We further use a simple set of experiments to justify that intuition. We run `LinGame` (along with `LinGame-C`) and the one of Degenne et al. [2019] that we call `DKM` over the problem instance whence $d = 2, c = 2, \delta = 0.01$ and $\alpha = 0.1$. We show the number of pulls for each arm averaged over 100 replications of experiments in Table 4.7. We see that, indeed, `DKM` pulls too much $\mathbf{x}_3$, while our `LinGame` focuses mostly on $\mathbf{x}_2$.

	`LinGame`	`LinGame-C`	`DKM`
a_1	1912	1959	1943
a_2	5119	4818	4987
a_3	104	77	1775
Total	7135	**6854**	8705

Table 4.7: Average number of pulls of `LinGame` and `LinGame-C` (against `DKM`) for each arm.

Next, we benchmark our sampling rules against others from the literature. We test over two synthetic problem instances, with the first being the previous instance. We set $d = 2, c = 2, \alpha = \pi/6$. Fig. 4.4 shows the empirical stopping time of each algorithms averaged over 100 runs, with a confidence level $\delta = 0.1, 0.01, 0.0001$ from left to right. Our two algorithms show competitive performance (the two leftmost boxes on each plot), and are only slightly worse than `LinGapE`.

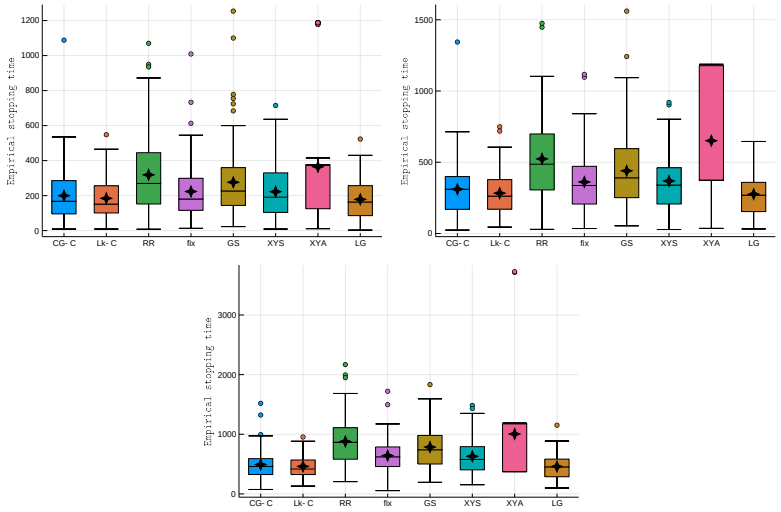


Figure 4.4: Sample complexity of different linear BAI sampling rules over the usual counter-example with $\delta = 0.1, 0.01, 0.0001$ respectively. CG = `LinGame-C`, Lk = `LinGame`, RR = uniform sampling, fix = tracking the fixed weights, GS = $\mathcal{XY}$`-Static` with $\mathcal{XX}$-allocation, XYS = $\mathcal{XY}$`-Static` with $\mathcal{XY}_{\texttt{dir}}$-allocation, LG = `LinGapE`. The mean stopping time is represented by a black cross.

For the second instance, we consider 20 arms randomly generated from the unit sphere $\mathbb{S}^{d-1} \triangleq \{a \in \mathbb{R}^d; \|a\|_2 = 1\}$. We choose the two closest arms a, a' and we set $\theta = a + 0.01(a' - a)$ so that a is the best arm. This setting has already been considered by Tao et al. [2018].

We report the same box plots over 100 replications as before with increasing dimension in Fig. 4.5. More precisely, we set $d = 6, 8, 10, 12$ respectively, and always keep a same $\delta = 0.01$. Our algorithms consistently show strong performances compared to other algorithms apart from `LinGapE`.

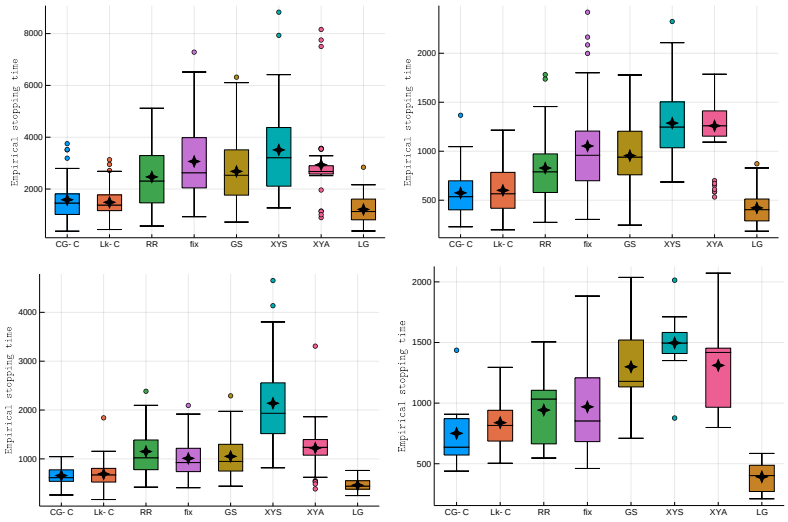


Figure 4.5: Sample complexity of different linear BAI sampling rules over random unit sphere vectors with $d = 6, 8, 10, 12$ from left to right.

4.7 Other Saddle-Point Approaches

Methods based on approaching the saddle point of the lower bound look promising, one concern about `LinGame` (or `LinGame-C`) could be its computational complexity though. In BAI, the one step complexity of `LinGame` is dominated by the computation of the best response for nature, which requires a full matrix inversion. Alternatives that involve rank-1 updates should be considered.

We end this chapter by some more discussions on other possible saddle-point methods.

4.7.1 Linear `Track-and-Stop`

The linear version of `Track-and-Stop` seems to be another plausible candidate of being asymptotically optimal provided that a numerical solver for computing the optimal weights exists (proven recently by Jedra and Proutière 2020). Fortunately, the Algorithm 4.2 proposed in Section 4.3.2 seems to be one.

`Track-and-Stop` remains unchanged in the linear case, since it only cares about tracking the optimal weights. We hereby recall that the `C-Tracking` rule consists of playing

$$\widehat{\mathbf{x}} \in \operatorname*{arg\,max}_{i \in [K]} \sum_{t=0}^{n} \omega_i^{\varepsilon_t}(\widehat{\boldsymbol{\mu}}_t) - T_{t,i}\,,$$

where $\boldsymbol{\omega}^{\varepsilon}$ is a L^{∞} projection of $\boldsymbol{\omega}^{\star}$ onto the simplex $\Sigma_K^{\varepsilon} \triangleq \{\boldsymbol{\omega} \in [\varepsilon, 1]^K : \sum_{i=1}^{K} \omega_i = 1\}$. The draw back of `Track-and-Stop` is that we need to compute a plug-in estimate of the optimal weights at each stage, which is computationally unfavorable.

4.7.2 Saddle-point Frank-Wolfe

On the other hand, the Frank-Wolfe heuristic in Algorithm 4.2 is an efficient rank-1 solver. It is thus natural to investigate if it can be incorporated into existing algorithms.

In particular, we can propose two new algorithms by adding the solver on top of `LinGapE` and `L-T3C` that we call Saddle-Point Linear Gap-Based Exploration (`SLinGapE`) and Saddle-Point Linear-Top-Two Transportation Cost (`SL-T3C`) respectively, as shown in Algorithm 4.6 and Algorithm 4.7. We define a so called *active transductive set* as

$$\widehat{\mathscr{Y}}(\mathbf{x},\boldsymbol{\theta}) \triangleq \left\{ \frac{(\mathbf{x}-\mathbf{x}')}{|(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\theta}|} : \mathbf{x}' \in \mathscr{X} / \{\mathbf{x}\} \right\}. \tag{4.14}$$

Algorithm 4.6 Algorithm of `SLinGapE`

Input: δ
Initialize: $\widetilde{\Lambda} \leftarrow \mathrm{I}_d, \Lambda \leftarrow \mathrm{I}_d$
for $n \leftarrow 1, 2, \cdots$ **do**
  $\widehat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in\mathscr{X}} \|\mathbf{x}\|^2_{\Lambda^{-1}\widetilde{\Lambda}\Lambda^{-1}}$
  $\widehat{\mathbf{x}}^{(1)} \leftarrow \arg\max_{\mathbf{x}\in\mathscr{X}} \mathbf{x}^\top \widehat{\boldsymbol{\theta}}^{\lambda}_n$
  $\widehat{\mathbf{x}}^{(2)} \leftarrow \arg\max_{\mathbf{x}\neq\widehat{\mathbf{x}}^{(1)}} (\mathbf{x}-\widehat{\mathbf{x}}^{(1)})^\top \widehat{\boldsymbol{\theta}}^{\lambda}_n + \sqrt{2\beta(n,\delta)} \left\|\widehat{\mathbf{x}}^{(1)}-\mathbf{x}\right\|_{\boldsymbol{\Lambda}^{-1}}$
  $\mathrm{B}_n \leftarrow \max_{\mathbf{x}\neq\widehat{\mathbf{x}}^{(1)}} (\mathbf{x}-\widehat{\mathbf{x}}^{(1)})^\top \widehat{\boldsymbol{\theta}}^{\lambda}_n + \sqrt{2\beta(n,\delta)} \left\|\widehat{\mathbf{x}}^{(1)}-\mathbf{x}\right\|_{\boldsymbol{\Lambda}^{-1}}$
  if $\mathrm{B}_n \leq 0$ **then**
    return $\widehat{\mathbf{x}}^{(1)}$
  end if
  $\widehat{\mathbf{y}} \leftarrow \dfrac{(\widehat{\mathbf{x}}^{(1)}-\widehat{\mathbf{x}}^{(2)})}{(\widehat{\mathbf{x}}^{(1)}-\widehat{\mathbf{x}}^{(2)})^\top \boldsymbol{\theta}}$
  $\boldsymbol{\Lambda} \leftarrow \Lambda + \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top$
  $\widetilde{\Lambda} \leftarrow \widetilde{\Lambda} + \widehat{\mathbf{y}}\widehat{\mathbf{y}}^\top$
  Evaluate arm $\widehat{\mathbf{x}}$
  Update mean and variance according to (4.1) and (4.7)
  $n = n+1$
end for

Algorithm 4.7 Sampling rule of `SL-T3C`

Initialize: $\widetilde{\Lambda} \leftarrow \mathrm{I}_d, \Lambda \leftarrow \mathrm{I}_d$
for $n \leftarrow 1, 2, \cdots$ **do**
  Sample $\boldsymbol{\theta} \sim \Pi_n$
  $\widehat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in\mathscr{X}} \|\mathbf{x}\|^2_{\Lambda^{-1}\widetilde{\Lambda}\Lambda^{-1}}$
  $\widehat{\mathbf{y}} \in \arg\max_{\mathbf{y}\in\widehat{\mathscr{Y}}(\widehat{\mathbf{x}},\boldsymbol{\theta})} \|\mathbf{y}\|^2_{\Lambda^{-1}}$
  $\Lambda \leftarrow \Lambda + \widehat{\mathbf{x}}\widehat{\mathbf{x}}^\top$
  $\widetilde{\Lambda} \leftarrow \widetilde{\Lambda} + \widehat{\mathbf{y}}\widehat{\mathbf{y}}^\top$
  Evaluate arm $\widehat{\mathbf{x}}$
  Update mean and variance according to (4.1) and (4.7)
  $n = n+1$
end for

4.7.3 Experimental illustrations

We compare our saddle-point-based algorithms against LinGapE. To make a fair comparison, we use always the same exploration rate for all the stopping rules. Indeed the stopping rules are equivalent if they keep the same exploration rate as argued in Appendix C.4. We use the previous hard instance with $c = 1$, the results are reported as box plots of average stopping time in Figure 4.6.

It seems that they can achieve the same level of empirical performance as LinGapE, their theoretical behaviour thus turns out to be an interesting research direction for the future.

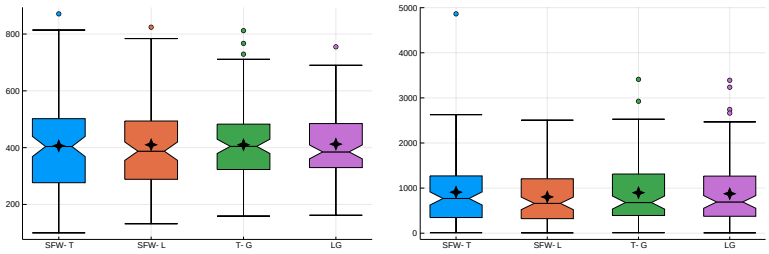


Figure 4.6: Average stopping time (Left: $d = 2, \alpha = \pi/4, \delta = 0.0000001$, Right: $d = 2, \alpha = 0.1, \delta = 0.1$), with SFW-T = SL-T3C, SFW-L = SLinGapE, T-G = L-T3C, LG = LinGapE.

4.8 Discussion

In this chapter, we designed the first practically usable asymptotically optimal sampling rules for the pure exploration game for finite-arm linear bandits. Whether the boundedness assumption is necessary to obtain optimal algorithms remains an open question.

Note that since the publication of our work, several other algorithms have been proposed [Jedra and Proutière, 2020; Katz-samuels et al., 2020; Zaki et al., 2020]. Particularly, Jedra and Proutière [2020] provide a first analysis of linear bandits BAI with a specific continuous set of arms; Katz-samuels et al. [2020] also study the fixed-budget setting for linear bandits BAI and propose a novel lower bound in terms of an experimental-design objective based on the Gaussian-width of the underlying arm set; while Zaki et al. [2020] follows more or less the same approach as ours. Later, Yang and Tan [2021] propose a minimax optimal algorithm for fixed-budget BAI.

More generally, however, the part of fixed-confidence pure exploration algorithms that needs an improvement the most is the stopping rule. While the one we used guarantees δ-correctness, it is very conservative. Indeed, the experimental error rates of algorithms using that stopping rule are orders of magnitude below δ. This means that the concentration inequality does not reflect the query we seek to answer. It quantifies deviations of the d-dimensional estimate in all directions (morally, along 2^d directions). However, for the usual BAI setting with d arms in an orthogonal basis, it would be sufficient to control the deviation of that estimator in $d-1$ directions to make sure that $\mathrm{I}^\star(\boldsymbol{\theta}) = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}^{\lambda}_n)$.

Finally, the good performance of LinGapE raises the natural question of whether it could be proven to have similar asymptotic optimality.

Chapter 5

Hierarchical Bandits for Black-Box Optimization

" *By heavens! there is something after all in the world allowing one man to steal a horse while another must not look at a halter. Steal a horse straight out. Very well. He has done it. Perhaps he can ride. But there is a way of looking at a halter that would provoke the most charitable of saints into a kick.*

Joseph Conrad

Contents

5.1 Introduction 72

5.2 Required Assumptions 73

5.2.1 General assumptions 73

5.2.2 Covering tree that guides the optimization 74

5.3 General Parallel Optimization 75

5.3.1 Generic parallel optimistic optimization 75

5.3.2 A more general wrapper 76

5.4 HCT under Local Smoothness w.r.t. $\mathscr{P}$ 78

5.4.1 Description of HCT 79

5.4.2 Analysis of HCT **under a local *metricless* assumption** 80

5.4.3 Upper bound on the simple regret of PCT 83

5.5 Experimental Illustrations 83

5.6 Discussion 85

5.1 Introduction

In this chapter, we adopt another perspective on sequential optimization. We are still interested in identifying the best arm, but this time among an infinite number of candidates. The search space can be countably infinite or even continuous. That is the global optimization.

GO has applications in several domains including hyper-parameter tuning [Jamieson and Talwalkar, 2016; Li et al., 2017; Samothrakis et al., 2013] which is the main topic of Chapter 6. GO usually consists of a data-driven optimization process over an expensive-to-evaluate function. It is also known as BBO since the inner behavior of a function is often unknown.

Contrary to Chapter 3 and Chapter 4, we are interested in a budgeted setting in this Chapter as we are subject to a high resource-consuming target, hence a pre-defined budget limit can be favored. In budgeted function optimization, a learner optimizes a function $f : \mathcal{X} \to \mathbb{R}$ depending on a number of function evaluations limited by N which are sequentially selected. For each of the N evaluations, at round n, the learner picks an element $x_n \in \mathcal{X}$ and observes a real number r_n, where

$$r_n = f(x_n) + \varepsilon_n,$$

with ε_n the noise. At the end, the learner is supposed to output a guess of the optimum and their performance is assessed by the simple regret (see Definition 2.16). The true function value $f(x)$ can thus be interpreted as the arm mean of x using MAB terminology.

Based on ε_n, we can distinguish *deterministic* feedback setting and *stochastic* feed setting. For deterministic feedback, the function evaluations are noiseless (see e.g. de Freitas et al. 2012 for motivation and applications). We pay our attention to the stochastic setting where the noise is assumed to be independent from past observations: $\mathbb{E}[r_n|x_n] = f(x_n)$.

Treating the problem without any further assumption would be a *mission impossible*. However, the setting gets easier if we assume a global smoothness of the reward function [Agrawal, 1995; Auer et al., 2007; Cope, 2009; Kleinberg, 2004; Kleinberg et al., 2008, 2013; Slivkins, 2011]. A weaker condition is some *local* smoothness where only neighborhoods around the maximum are required to be smooth. In fact, local smoothness is sufficient for achieving near-optimality [Azar et al., 2014; Bull, 2015; Grill et al., 2015; Valko et al., 2013]. This is the *continuum-armed bandits* setting.

We base our work on optimistic tree-based optimization algorithms [Azar et al., 2014; Munos, 2011; Preux et al., 2014; Valko et al., 2013] that approach the problem with a hierarchical partitioning of the arm space and take the *optimistic principle*. This idea comes from *planning* in MDP [Grill et al., 2016; Kocsis and Szepesvári, 2006; Munos, 2014].

Our work is motivated by the Parallel Optimistic Optimization (POO) approach proposed by Grill et al. [2015], that *adapts to the smoothness* without the knowledge of it. POO is a *meta-algorithm* which can be used on top of any hierarchical optimization algorithm that *knows the smoothness*, that we call a subroutine. Not only does POO require only the mildest local regularity conditions, but it also gets rid of the unnecessary metric assumption that is often required. Local smoothness naturally covers a larger class of functions than global smoothness, yet still assures that the function does not decrease too fast around the maximum. We highlight that the analysis of POO is modular: Assuming the subroutine has a **cumulative** regret of order R_N *under a local smoothness assumption with respect to a fixed partitioning* (Grill et al. 2015, Assumption 5.2, formally introduced in Section 5.2), POO run with such subroutine has a **simple** regret bounded by $R_N\sqrt{\log N}$.

The analysis of POO heavily relies on the subroutine having guarantees on the cumulative regret. In the context of optimization where sometimes only simple regret guarantees

are available, this is not always desirable as requirements. In this chapter, we provide a more general wrapper for algorithms that only have guarantees on their simple regret, called General Parallel Optimization (GPO). We show that with a cross-validation scheme instead of the original recommendation strategy, any hierarchical bandit algorithm with simple regret guarantee can be plugged into GPO with only a tiny increase in the resulting simple regret. Note that any subroutine that is able to underpin POO can do the same for GPO while the converse is not necessarily true. This can be explained by 2.6 as good cumulative regret guarantees imply good simple regret guarantees but not the converse.

To validate GPO, it is necessary to find a subroutine that achieves meaningful regret bound (can be either cumulative regret or simple regret) under Assumption 5.2. A natural candidate can refer to the subroutine of POO. POO was originally analyzed using Hierachical Optimistic Optimization (HOO) as its base algorithm. However, unlike what Grill et al. [2015] hypothesize, it is non-trivial to provide a regret bound for HOO under Assumption 5.2. We elaborate on that in Section 5.4. In order to validate POO as well as GPO, there needs to exist a subroutine with a regret guarantee that is provable under Assumption 5.2. This is another message that we deliver.

In particular, we prove that HCT-iid, denoted by High Confidence Tree (HCT) in the rest of the paper since we do not consider the correlated feedback setting, of Azar et al. [2014] satisfies the required regret guarantee, and is, therefore, a desirable subroutine to be plugged in POO and GPO. Similar to HOO, HCT is a hierarchical optimization algorithm based on confidence intervals. However, unlike HOO, these confidence intervals are obtained by repeatedly sampling a representative point of each cell in the partitioning before splitting the cell. This yields partition trees that have a *controlled depth*, which are easier to analyze under a local smoothness assumption with respect to the partitioning. Whether HOO has similar regret guarantees under the desired local metricless assumption remains an open question.

Contributions. **1)** We propose to use a cross-validation scheme to wrap up algorithms that only possess simple regret guarantees at the cost of a slight loss in the final regret bound. **2)** We show that HCT can serve as a valid subroutine under desired assumptions. **3)** We further provide numerical illustrations to show that HCT is empirically comparable to HOO as a subroutine.

☞ This chapter is based on Shang et al. [2018, 2019a].

5.2 Required Assumptions

5.2.1 General assumptions

Let $\mathcal{X}$ be a measurable space. Our goal is to find the maximum of an unknown noisy function $f : \mathcal{X} \to \mathbb{R}$ of which the cost of evaluation is high, given a total budget of N evaluations. At each round n, a learner selects a point $x_n \in \mathcal{X}$ and observes a reward $r_n \triangleq f(x_n) + \varepsilon_n$. We make the following assumption on the noise in this thesis. More discussions on the role and impact of noise are provided by Bartlett et al. [2019].

Assumption 5.1. *[bounded, independent and conditionally centered noise] We assume that the noise ε_n is bounded by $[0,1]$, independent from previous observations and such*

that

$$\mathbb{E}[\varepsilon_n | x_n] = 0\,.$$

After N evaluations, the algorithm outputs a guess for the maximizer, denoted by $x(\mathrm{N})$. $x(\mathrm{N})$ is indeed the decision rule $\mathrm{J_N}$ in the learning protocol of a general pure-exploration problem. We assume that there exists at least one $x^\star \in \mathcal{X}$ s.t. $f(x^\star) \triangleq \sup_{x\in\mathcal{X}} f(x)$, denoted by $f^\star$ in the following. We measure the performance by the simple regret. We can particularize Definition 2.16 of simple regret into the setting of this chapter:

$$\mathrm{S_N} \triangleq f^\star - f(x(\mathrm{N}))\,.$$

Likewise, we can also particularize the notion of cumulative regret into our setting:

$$\mathrm{R_N} \triangleq \mathrm{N}f^\star - \sum_{n=1}^{\mathrm{N}} f(x_n)\,.$$

5.2.2 Covering tree that guides the optimization

Hierarchical bandits rely on the existence of hierarchical partitioning $\mathcal{P} \triangleq \{\mathcal{P}_{h,i}\}_{h,i}$ defined recursively, where

$$\mathcal{P}_{0,1} = \mathcal{X}\,, \quad \mathcal{P}_{h,i} = \bigcup_{j=0}^{\mathrm{K}-1} \mathcal{P}_{h+1,\mathrm{K}i-j}\,.$$

Such a partition can be naturally represented by a tree, where K denotes the maximum number of children of a node in that tree. Many of known algorithms depend on a metric/dissimilarity over the search space to define the regularity assumptions that link the partitioning to some *near-optimality dimension*, that is independent of the partitioning. However, this was shown to be artificial [Grill et al., 2015], since (i) the metric is not fully exploited by the algorithms and (ii) the notion of near-optimality dimension independent of partitioning is ill-defined. Hence, it is natural to make smoothness assumptions directly related only to the partitioning.

We now present *the only regularity assumption* on the target function f that is expressed in terms of the partitioning $\mathcal{P}$ given in Assumption 5.2.

Assumption 5.2. *[local smoothness w.r.t. $\mathcal{P}$] For $x^\star$ be a global maximizer, we denote by $i_h^\star$ be the index of the only cell at depth h that contains $x^\star$. Then, there exist a global maximizer $x^\star$ and two constants $\nu > 0$, $\rho \in (0,1)$ s.t.,*

$$\forall h \geq 0, \forall x \in \mathcal{P}_{h,i_h^\star},\ \ f(x) \geq f^\star - \nu\rho^h.$$

Note that this assumption is the same as the one of Grill et al. [2015]. Multiple maximizers may exist, but this assumption needs to be satisfied only by one of them.

We stress again that requiring only a **local** smoothness assumption is an improvement since (i) it is a one-side local Lipschitz-type of assumption that naturally covers a larger class of functions, (ii) it only constrains f along the optimal path of the covering tree which is a plausible property in an optimization scenario, and (iii) shows that the optimization is actually easier than it was previously believed.

Besides, in this chapter, we aim to design algorithms that does not rely on any metric. Previous methods all depend on a metric, and their smoothness assumptions are summarized in Table 5.1.

	global smoothness	**local** smoothness
known smoothness	`Zooming`, `HOO`	`DOO`, `HCT`
unknown smoothness	`TaxonomyZoom`	`SOO`, `StoSOO`, `ATB`

Table 5.1: Smoothness assumptions for hierarchical bandits algorithms.

As first observed by Auer et al. [2007], the difficulty of a GO should depend on the size of near-optimal regions and on how fast they shrink. Auer et al. [2007] use a margin condition that quantifies this difficulty by the volume of near-optimal regions. In this work, we use a similar notion of *near-optimality dimension*[1] instead. This notion is directly related to the partitioning.

Definition 5.1. *[near-optimality dimension w.r.t. $\mathscr{P}$] For any $\nu > 0, C > 1$, and $\rho \in (0,1)$, we define the near-optimality dimension of f with respect to $\mathscr{P}$ as*

$$d(\nu, C, \rho) \triangleq \inf\left\{ d' \in \mathbb{R}^+ : \forall h \geq 0, \mathcal{N}_h(3\nu\rho^h) \leq C\rho^{-d'h} \right\},$$

where $\mathcal{N}_h(\varepsilon)$ is the number of cells $\mathscr{P}_{h,i}$ such that $\sup_{x \in \mathscr{P}_{h,i}} f(x) \geq f^\star - \varepsilon$.

$\mathcal{N}_h(3\nu\rho^h)$ can be thought as the number of cells that any algorithm needs to sample in order to find the maximum. A smaller $d(\nu, C, \rho)$ implies an easier optimization problem.

5.3 General Parallel Optimization

We present our new wrapper in this section. In order to get a better understanding, we start with a brief introduction of `POO`.

5.3.1 Generic parallel optimistic optimization

We introduce `POO`($\mathscr{A}$) as a generic algorithm, taking as input *any* hierarchical optimization algorithm $\mathscr{A} = \mathscr{A}(\nu, \rho)$ requiring the smoothness parameters.

`POO`($\mathscr{A}$) is a meta-algorithm that uses $\mathscr{A}$ that knows the smoothness as a subroutine, originally proposed by Grill et al. [2015] for $\mathscr{A} =$ `HOO`. In this algorithm, several instances of $\mathscr{A}$ are run in parallel, each one using a different pair of parameters (ν, ρ) in a well-chosen grid $\mathscr{G}$ (defined in Line 4 of Algorithm 5.1). In the end, `POO`($\mathscr{A}$) chooses the instance that has the largest empirical mean reward and returns one of the points evaluated by this instance, chosen uniformly at random.

The pseudo-code of `POO`($\mathscr{A}$) is shown in Algorithm 5.1. Additionally to the base algorithm itself, it requires two parameters $\rho_{\max}$ and $\nu_{\max}$ that determine the range of instances $\mathscr{A}(\nu, \rho)$ that we can compete with. However, these parameters can be set as a function of the number of evaluations as explained in details in Appendix C of Grill et al.

[1] The present definition is slightly different from the original `POO` paper, where a coefficient 3 is present instead of 2 due to a technical detail.

[2015], hence not mandatory in practice. An important remark is that given a budget N of function evaluations, the number of instances M run by POO($\mathscr{A}$) depends on N, and each instance is run for $\lfloor \mathrm{N}/\mathrm{M}(\mathrm{N}) \rfloor$ times. Due to the doubling scheme used in Lines 2-10 of Algorithm 5.2, note however that POO($\mathscr{A}$) does not need to know this total number of function evaluations. Hence, if the base algorithm $\mathscr{A}$ is anytime, so is POO($\mathscr{A}$).

Algorithm 5.1 Algorithm of POO($\mathscr{A}$) with base algorithm $\mathscr{A}$

1: **Input:** base algorithm $\mathscr{A}$, $\nu_{\max}$,$\rho_{\max}$, branching factor of the partitioning K
2: **Initialization:** $\mathrm{D}_{\max} \leftarrow \ln \mathrm{K} / \ln\left(1/\rho_{\max}\right)$, number of function evaluations $n \leftarrow 0$, current number of instances of $\mathscr{A}$: $\mathrm{N} \leftarrow 1$, $\mathscr{S} \leftarrow \{(\nu_{\max}, \rho_{\max})\}$
3: **while** budget still available **do**
4: **while** $\mathrm{N} \leq \frac{1}{2}\mathrm{D}_{\max} \log\left(\mathrm{N}/(\log \mathrm{N})\right)$ **do**
5: **for** $i \leftarrow 1, \ldots, \mathrm{N}$ **do**
6: $s \leftarrow \left(\nu_{\max}, {\rho_{\max}}^{2\mathrm{M}/(2i+1)}\right)$
7: Initialize $\mathscr{A}(s)$ (if not already done before)
8: Continue running $\mathscr{A}(s)$ until it has given $\frac{\mathrm{N}}{\mathrm{M}}$ rewards $r_{s,1}, \ldots, r_{s,\mathrm{N}/\mathrm{M}}$.
9: Compute $\widehat{\mu}[s] = \frac{\mathrm{M}}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}/\mathrm{M}} r_{s,i}$.
10: **end for**
11: $\mathrm{N} \leftarrow 2\mathrm{N}$
12: $\mathrm{M} \leftarrow 2\mathrm{M}$
13: **end while**
14: Perform each $\mathscr{A}(s)$ once
15: Update $\widehat{\mu}[s]$
16: $\mathrm{N} \leftarrow \mathrm{N} + \mathrm{M}$
17: **end while**
18: $s^\star \leftarrow \arg\max_{s \in \mathscr{S}} \widehat{\mu}[s]$
19: **Return** A point sampled u.a.r. from the points evaluated by $\mathscr{A}(s^\star)$

5.3.2 A more general wrapper

The analysis of POO($\mathscr{A}$) heavily relies on the fact that we control the *cumulative regret* of algorithm $\mathscr{A}$ (see Appendix D.2 for details). POO indeed exploits this property when selecting $s^\star$ as the instance with largest empirical cumulative rewards. In this section, we propose a simple modification of POO($\mathscr{A}$) that allows using as base algorithms any hierarchical optimization algorithms that would only have *simple regret* guarantees.

The GPO($\mathscr{A}$) algorithm, whose pseudo-code is shown in Algorithm 5.2, mostly needs to modify the model selection strategy of POO. There are two natural candidates: (i) Lepski's method which is a nested aggregation scheme [Lepski, 1992; Lepski and Spokoiny, 1997; Locatelli and Carpentier, 2018; Locatelli et al., 2017] that requires a single optimum, thus not directly applicable to our case, and (ii) a cross-validation scheme that we use and detail in the next. Given a total budget of n function evaluations, GPO($\mathscr{A}$) runs several instances of $\mathscr{A}$ in parallel with parameters chosen in the same grid as that used by POO, each using the same number of evaluations to output a recommendation $\widetilde{x}_i$. One half of the budget is then dedicated to estimating the function values at those points, and the one with the highest estimated value is kept.

In Theorem 5.1, we provide a general analysis of the GPO algorithm, showing that it attains an (order)-optimal simple regret without knowing the parameter triple $(\nu^\star, \mathrm{C}^\star, \rho^\star)$ provided that its base algorithm does.

Algorithm 5.2 Algorithm of GPO

Input: base algorithm $\mathscr{A}$, budget n, $\rho_{\max}, \nu_{\max}$, K
Initialization: $D_{\max} \leftarrow \ln K / \ln\left(1/\rho_{\max}\right)$, number of function evaluations $N \leftarrow 0$, current number of HCT instances $M \leftarrow 1$, $\mathscr{S} \leftarrow \{(\nu_{\max}, \rho_{\max})\}$
Compute $M = \lceil (1/2) D_{\max} \ln((M/2)/\ln(M/2)) \rceil$ (the number of instances)
for $i \leftarrow 1, \dots, M$ **do**
    $s \leftarrow \left(\nu_{\max}, \rho_{\max}^{2M/(2i+1)}\right)$
    Run $\mathscr{A}(s)$ for $\lfloor N/(2M) \rfloor$ time steps
    Recommend $\widetilde{x}_s$
    Get $\lfloor N/(2M) \rfloor$ noisy evaluations of $f(\widetilde{x}_s)$
    Compute their average $V[s]$
end for
$s^\star \leftarrow \arg\max_s V[s]$
Return $\widetilde{x}_{s^\star}$

Theorem 5.1. *If for all* (ν, ρ) *the* $\mathscr{A}(\nu, \rho)$ *algorithm has its* simple regret *bounded as*

$$\mathbb{E}\left[S_N^{\mathscr{A}(\nu,\rho)}\right] \le \alpha C \left((\log N/N)^{1/(d(\nu,C,\rho)+2)}\right), \tag{5.1}$$

for any function f *satisfying Assumption 5.2 with parameters* (ν, ρ)*, then there exists a constant* β *that is independent of* $\nu_{\max}$ *and* $\rho_{\max}$ *such that*

$$\mathbb{E}\left[S_N^{GPO(\mathscr{A})}\right] \le \beta D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left((\log^2 N)/N\right)^{1/(d(\nu^\star,C^\star,\rho^\star)+2)},$$

for any function f *satisfying Assumption 5.2 with parameters* $\nu^\star \le \nu_{\max}$ *and* $\rho^\star \le \rho_{\max}$.

Proof. We start by fixing some notation. Recall that M (that depends on N) is the number of instances run in parallel. For $j \in \{1, \dots, M\}$, we let $\widetilde{x}_j$ denote the point recommended by the instance $\mathscr{A}(\nu_{\max}, \rho_j)$ with $\rho_j = \rho_{\max}^{2M/(2j+1)}$. Let $(r_{i,j})_{1 \le i \le N^+}$ be the i.i.d. evaluations of $f(\widetilde{x}_j)$ used during the validation phase, with $N^+ \triangleq \lfloor N/(2M) \rfloor$ and $\widehat{\mu}_{N^+,j} = \frac{1}{N^+} \sum_{i=1}^{N^+} r_{i,j}$ be the estimated value of $f(\widetilde{x}_j)$ computed by the algorithm. We let

$$\widehat{j} = \arg\max_j \widehat{\mu}_{N^+,j} \quad \text{and} \quad \widetilde{j} = \arg\max_j f(\widetilde{x}_j)$$

be the index of the empirical best and true best among the recommended point. We notice that for any j, $\{r_{i,j} - f(\widetilde{x}_j)\}_{i=1}^{N^+}$ is a bounded i.i.d. sequence with zero mean (conditionally to $\widetilde{x}_j$) thus using Hoeffding's inequality one can show that for all $\Delta > 0$,

$$\mathbb{P}\left[\sum_{i=1}^{N^+} (r_{i,j} - f(\widetilde{x_j})) > N^+ \Delta\right] \le \exp\left(-2N^+\Delta^2\right),$$

therefore,

$$\mathbb{P}\left[\widehat{\mu}_{N^+,j} - f(\widetilde{x_j}) > \Delta\right] \le \exp\left(-2N^+\Delta^2\right),$$

and we have immediately

$$\mathbb{P}\left[|\widehat{\mu}_{N^+,j} - f(\widetilde{x}_j)| > \Delta\right] \le 2\exp\left(-2N^+\Delta^2\right).$$

By integrating over $\Delta \in [0,1]$, we get

$$\forall\, j \in \{1,\dots,M\},\ \mathbb{E}\left[|\widehat{\mu}_{N^+,j} - f(\widetilde{x}_j)|\right] \leq \frac{\sqrt{\pi/2}}{\sqrt{N^+}}. \tag{5.2}$$

As in the analysis of POO, the instance $\overline{\jmath}$ defined as

$$\overline{\jmath} \triangleq \operatorname*{argmin}_{j \leq M : \rho_j \geq \rho^\star} \left[d(\nu_{\max}, C^\star, \rho_j) - d(\nu^\star, C^\star, \rho^\star)\right]$$

shall play a crucial role. Indeed, inequality (5.1) is exactly what is needed in Appendix B.2 and Appendix B.3 of Grill et al. [2015] to control the simple regret of that instance in terms of $(\nu^\star, C^\star, \rho^\star)$. Following the exact same steps, we can show that for some constant α,

$$\mathbb{E}\left[S_{(N/2M)}^{\mathscr{A}(\nu_{\max},\rho_{\overline{\jmath}})}\right] \leq \alpha D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left((\log^2 N)/N\right)^{1/(d(\nu^\star,C^\star,\rho^\star)+2)}. \tag{5.3}$$

We now turn our attention to the simple regret of GPO($\mathscr{A}$) after n function evaluations.

$$\mathbb{E}\left[S_N^{\mathrm{GPO}}\right] = \mathbb{E}\left[f^\star - f(\widetilde{x}_{\widehat{\jmath}})\right] = \mathbb{E}\left[f^\star - f(\widetilde{x}_{\overline{\jmath}})\right] + \mathbb{E}\left[f(\widetilde{x}_{\overline{\jmath}}) - f(\widetilde{x}_{\widetilde{\jmath}})\right] + \mathbb{E}\left[f(\widetilde{x}_{\widetilde{\jmath}}) - f(\widetilde{x}_{\widehat{\jmath}})\right]. \tag{5.4}$$

The first term in (5.4) is equal to the simple regret of the instance $\overline{\jmath}$ that uses n/N samples, which is upper bounded in (5.3). The second term in (5.4) is always negative by definition of $\widetilde{\jmath}$ and the third term can be rewritten as

$$\mathbb{E}\left[f(\widetilde{x}_{\widetilde{\jmath}}) - f(\widetilde{x}_{\widehat{\jmath}})\right] = \mathbb{E}\left[f(\widetilde{x}_{\widetilde{\jmath}}) - \widehat{\mu}_{N^+,\widetilde{\jmath}}\right] + \mathbb{E}\left[\widehat{\mu}_{N^+,\widetilde{\jmath}} - \widehat{\mu}_{N^+,\widehat{\jmath}}\right] + \mathbb{E}\left[\widehat{\mu}_{N^+,\widehat{\jmath}} - f(\widetilde{x}_{\widehat{\jmath}})\right]. \tag{5.5}$$

where the first and the third term of (5.5) are both upper bounded by $(\sqrt{\pi/2})/\sqrt{N^+}$ using (5.2), and the second term is always negative by definition of $\widehat{\jmath}$. Putting things together yields

$$\mathbb{E}\left[S_N^{\mathrm{GPO}}\right] \leq \alpha D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left((\log^2 N)/N\right)^{1/(d(\nu^\star,C^\star,\rho^\star)+2)} + O\left(\frac{\sqrt{M}}{\sqrt{N}}\right).$$

The conclusion follows by observing that the second term in the right-hand side is negligible with respect to the first. □

5.4 HCT under Local Smoothness w.r.t. $\mathscr{P}$

Not let us turn our attention to finding a valid base algorithm for GPO. The first idea is to refer to the original base algorithm of POO, namely HOO.

Analyzing HOO under Assumption 5.2, however, is not trivial. A key lemma in the analysis of HOO (Lemma 3 by Bubeck et al. 2011) that controls the variance of near-optimal cells *is not true* under local smoothness assumptions as Assumption 5.2. Indeed, HOO could induce a very deep covering tree, while producing too many nodes that are neither near-optimal nor sub-optimal. The concept of near-optimal and sub-optimal nodes is then characterized by the *sub-optimality gap* of each node which measures the distance between the local maximum of the node and the global maximum. Intuitively, nodes that are neither near-optimal nor sub-optimal represent the nodes of whom the sub-optimality gap is neither too large nor too small.

To control the regret due to these nodes, Bubeck et al. [2011] use global smoothness (weakly Lipschitz) assumption. Assumption 5.2 is weaker, only local, and does not offer

such comfort. If we want to control the regret due to these nodes without Lemma 3 of Bubeck et al. [2011], one possible way is to control the depth of the covering tree to ensure that we do not have too many of them. In particular, another algorithm known as HCT proposed by Azar et al. [2014] implies a controlled depth of the tree which allows it to be analyzed under Assumption 5.2 as opposed to HOO. We now give a brief description of HCT and present a new analysis of it.

5.4.1 Description of HCT

Algorithm 5.3 Algorithm of HCT

Input: K, $\nu > 0$, $\rho \in (0,1)$, $c > 0$, tree partitioning $\{\mathscr{P}_{h,i}\}$, confidence δ
Initialization: $\mathscr{T}_1 \leftarrow \{(0,1),(1,1),\dots,(1,K)\}$, $U_{1,1}(1) \leftarrow \cdots \leftarrow U_{1,K}(1) \leftarrow +\infty$
for $n \leftarrow 1 \dots N$ **do**
  if $n = n^+$ **then**
    for $(h,i) \in \mathscr{T}_n$ **do**
      $U_{h,i}(n) \leftarrow \widehat{\mu}_{h,i}(n) + \nu\rho^h + c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{T_{h,i}(n)}}$
    end for
    UpdateBackward($\mathscr{T}_n, n$)
  end if
  $(h_n, i_n), P_n \leftarrow$ OptTraverse($\mathscr{T}_n, n$)
  Evaluate x_{h_n,i_n} and obtain r_n
  $T_{h_n,i_n}(n) \leftarrow T_{h_n,i_n}(n) + 1$
  Update $\widehat{\mu}_{h_n,i_n}(n)$
  $U_{h_n,i_n}(n) \leftarrow \widehat{\mu}_{h_n,i_n}(n) + \nu\rho^{h_n} + c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{T_{h_n,i_n}(n)}}$
  UpdateBackward(P_n, n)
  $\tau_{h_n}(n) \leftarrow \lceil \frac{c^2 \log(1/\widetilde{\delta}(n^+))}{\nu^2} \rho^{-2h_n} \rceil$
  if $T_{h_n,i_n}(n) \geq \tau_{h_n}(n)$ and (h_n, i_n) is a leaf **then**
    Expand the node (h_n, i_n)
  end if
end for

Algorithm 5.4 Snippet OptTraverse of HCT

Input: a tree $\mathscr{T}$, round n
Initialization: $(h,i) \leftarrow (0,1)$; $P \leftarrow \{(0,1)\}$; $T_{0,1}(n) = \tau_0(n) = 1$
while (h,i) is not a leaf of $\mathscr{T}$ and $T_{h,i}(n) \geq \tau_h(n)$ **do**
  $j \leftarrow \underset{j \in \{0,\dots,K-1\}}{\operatorname{argmax}} \{B_{h+1,Ki-j}(n)\}$
  $(h,i) \leftarrow (h+1, Ki-j)$
  $P \leftarrow P \cup \{(h,i)\}$
end while
Return (h,i) and P

The pseudo-code of HCT (Algorithm 5.3) and two detailed snippets (Algorithm 5.4 and Algorithm 5.5) describe the process of traversing the covering tree. The algorithm stores a finite subtree $\mathscr{T}_n$ at each round t which is initialized by $\mathscr{T}_0 = \{(0,1)\}$. Each cell is associated with a representative point $x_{h,i}$ and the algorithm keeps track of some statistics regarding

Algorithm 5.5 Snippet `UpdateBackward` of `HCT`

Input: a tree $\mathcal{T}$, round n
    note that P_n *can also be considered as a tree, thus input of this function*
for $(h,i) \in \mathcal{T}$ backward from each leaf of $\mathcal{T}$ **do**
    if (h,i) is a leaf of $\mathcal{T}$ **then**
        $\mathrm{B}_{h,i}(n) \leftarrow \mathrm{U}_{h,i}(n)$
    else
        $\mathrm{B}_{h,i}(n) \leftarrow \min\left\{\mathrm{U}_{h,i}(n), \max_{j\in\{0,\dots,\mathrm{K}-1\}} \{\mathrm{B}_{h+1,\mathrm{K}i-j}(n)\}\right\}$
    end if
end for

this point. One of these statistics is the empirical mean reward $\widehat{\mu}_{h,i}(n)$ which is the average on the first $\mathrm{T}_{h,i}(n)$ rewards received when querying $x_{h,i}$. The `HCT` algorithm also keeps track of an upper confidence bound U-value for the cell (h,i),

$$\mathrm{U}_{h,i}(n) \triangleq \widehat{\mu}_{h,i}(n) + \nu\rho^h + c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h,i}(n)}},$$

where $n^+ \triangleq 2^{\lceil \log_2(n) \rceil}$, $\widetilde{\delta}(n) \triangleq \min\{c_1\delta/t, 1/2\}$, and its corresponding B-value,

$$\mathrm{B}_{h,i}(n) \triangleq \begin{cases} \min\left\{\mathrm{U}_{h,i}(n), \max_{j\in\{0,\dots,\mathrm{K}-1\}} \{\mathrm{B}_{h+1,\mathrm{K}i-j}(n)\}\right\} & \text{if } (h,i) \text{ is an internal node,} \\ \mathrm{U}_{h,i}(n) & \text{otherwise,} \end{cases}$$

which is designed to be a tighter upper confidence bound than the U-value. Here, c and c_1 are two constants, and $\nu\rho^h$ represents the *resolution*[2] of the region $\mathcal{P}_{h,i}$. Observe that $\mathrm{U}_{h,i}(n)$ and $\mathrm{B}_{h,i}(n)$ are not updated at every round, but are constant on time intervals of the form $[2^k, 2^{k+1})$.

At each round n, the algorithm traverses the current covering tree along an *optimistic path* P_n before choosing a point (`OptTraverse` function). This optimistic path P_n is obtained by repeatedly selecting cells that have a larger B-value until a leaf or a node that is sampled less than a certain number of times is reached. If a leaf is reached, then this leaf is sampled and expanded (i.e., we split the leaf into K equal-sized regions and initialize their U-values to $+\infty$); otherwise, the node that is not sampled enough is re-sampled. All the B-values along the optimistic path are then updated backwardly from the current node to the root (`UpdateBackward` function). More precisely, `HCT` samples one node a certain number of times $\tau_h(n)$ in order to sufficiently reduce the uncertainty before expanding it. Hence, $\tau_h(n)$ is defined such that the uncertainty over the rewards in $\mathcal{P}_{h,i}$ is roughly equal to the resolution of the node,

$$\tau_h(n) \triangleq \lceil \frac{c^2 \log(1/\widetilde{\delta}(n^+))}{\nu^2} \rho^{-2h} \rceil.$$

5.4.2 Analysis of `HCT` under a local *metricless* assumption

We now show that `HCT` is indeed a valid candidate underlying algorithm for `GPO`.

[2] The term *resolution* refers to the maximum variation in the cell. If it is too large, then we need to shrink the volume, thus increase the resolution.

We state our main result in Theorem 5.2. We prove that HCT achieves an expected cumulative regret bound under Assumption 5.2 which matches the regret bound given by Azar et al. [2014] up to constants.

Moreover, compared to that result, the *near-optimality dimension* d featured in Theorem 5.2 is the one of Definition 5.1 that is defined with respect to the partitioning and not with respect to a metric. For a fixed budget N, we introduce the notation HCT(ν,ρ) to refer to the instance of HCT parameterized by ν, ρ, $c = 2\sqrt{1/(1-\rho)}$ and $\delta = 1/N$.

Theorem 5.2. *Assume that function f satisfies Assumption 5.2. Then, setting $\delta \triangleq 1/N$, the cumulative regret of HCT(ν,ρ) after N function evaluations is upper bounded as*

$$\mathbb{E}[R_N^{HCT(\nu,\rho)}] \leq \alpha C (\log N)^{1/(d(\nu,C,\rho)+2)} N^{(d(\nu,C,\rho)+1)/(d(\nu,C,\rho)+2)},$$

where α is a numerical constant and C is the constant associated to $d(\nu, C, \rho)$.

As a consequence, according to Remark 2.6, we get the following simple-regret bound.

Corollary 5.3. *The simple regret of HCT after N function evaluations under Assumption 5.2 satisfies*

$$\mathbb{E}[S_N^{HCT(\nu,\rho)}] \leq \alpha C (\log N)^{1/(d(\nu,C,\rho)+2)} N^{-1/(d(\nu,C,\rho)+2)}.$$

Remark 5.4. *One may notice that to validate GPO, we only need to bound the simple regret of HCT. The reason that we provide a cumulative regret bound is that we can show that HCT is also a valid base algorithm for POO, thus validate POO as well. As indeed the problem raised in the analysis of HOO under Assumption 5.2 does make the validity of POO questionable.*

We now sketch the proof. The full proof is detailed in Appendix D.2. As mentioned above, HCT has a controlled depth. Indeed, given the threshold $\tau_h(n)$ required at depth h, in Section D.2.1, we prove that the depth of the covering tree is bounded as stated in the following lemma:

Lemma 5.1. *The depth of the covering tree produced by HCT after N function evaluations satisfies*

$$H(N) \leq H_{\max}(N) \triangleq \left\lceil \frac{1}{2(1-\rho)} \log\left(\frac{N\nu^2}{c^2\rho^2}\right) \right\rceil.$$

Defining the mean reward $\mu_{h,i} \triangleq f(x_{h,i})$, we introduce a favorable event under which the mean reward of all expanded nodes is within a confidence interval,

$$\xi_n \triangleq \left\{ \forall (h,i) \in \mathscr{L}_n, |\widehat{\mu}_{h,i}(n) - \mu_{h,i}| \leq c\sqrt{\log(1/\widetilde{\delta}(n))/T_{h,i}(n)} \right\},$$

where $\mathscr{L}_n$ is the set of all possible nodes in trees of maximum depth $\mathrm{H}_{\max}(n)$.

We split the regret into two parts depending on whether ξ_n holds or not. We prove in Appendix D.2.3 that the failing confidence term is with high probability bounded by $\sqrt{n}$. In the case when ξ_n holds, we bound the regret in Appendix D.2.4 by treating separately the two parts, Δ_{h_n,i_n} and $\widehat{\Delta}_n$, of the instantaneous regret Δ_n,

$$\Delta_n \triangleq f^\star - r_n = f^\star - f(x_{h_n,i_n}) + f(x_{h_n,i_n}) - r_n = \Delta_{h_n,i_n} + \widehat{\Delta}_n.$$

Next, we bound $\widehat{\Delta}_n$ by Azuma-Hoeffding concentration inequality [Azuma, 1967]. Then, we bound Δ_{h_n,i_n} with the help of the following lemma, which is the major difference compared to the original HCT analysis by Azar et al. [2014]. In particular, the lemma states that if Assumption 5.2 is verified then $f^\star$ is upper-bounded by the U-value of any optimal node.

Lemma 5.2. *Under Assumption 5.2 and under event ξ_n, we have that for any optimal node $(h^\star, i^\star)$, $\mathrm{U}_{h^\star,i^\star}(n)$ is an upper bound on $f^\star$.*

Proof. Since $n^+ \geq n$, we have

$$\mathrm{U}_{h^\star,i^\star}(n) \triangleq \widehat{\mu}_{h^\star,i^\star}(n) + \nu\rho^{h^\star} + c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h^\star,i^\star}(n)}} \geq \widehat{\mu}_{h^\star,i^\star}(n) + \nu\rho^{h^\star} + c\sqrt{\frac{\log(1/\widetilde{\delta}(n))}{\mathrm{T}_{h^\star,i^\star}(n)}}.$$

Moreover, as we are under event ξ_n, we also have

$$\widehat{\mu}_{h^\star,i^\star}(n) + c\sqrt{\frac{\log(1/\widetilde{\delta}(n))}{\mathrm{T}_{h^\star,i^\star}(n)}} \geq f\left(x_{h^\star,i^\star}\right).$$

Therefore, $\mathrm{U}_{h^\star,i^\star}(n) \geq f(x_{h^\star,i^\star}) + \nu\rho^{h^\star} \geq f^\star$. □

With the help of Lemma 5.2 (see Step 2 in Appendix D.2.4), we can then upper bound Δ_{h_n,i_n} as

$$\Delta_{h_n,i_n} \leq 3c\sqrt{\frac{\log(2/\widetilde{\delta}(n))}{\mathrm{T}_{h_n,i_n}(n)}}.$$

To bound the total regret of the all nodes selected, we divide them into two categories, depending on whether their depth is smaller or equal than $\overline{\mathrm{H}}$ (to be optimized later) or not.

For the nodes in depths $h \leq \overline{\mathrm{H}}$, we use Lemma 5.2 again, now to show that `OptTraverse` only selects nodes that have a parent which is $(3\nu\rho^{h_n-1})$-optimal. For the nodes for which $h > \overline{\mathrm{H}}$, we bound the regret using the selection rule of HCT.

The sums of the regrets from the two categories are proportional and inversely proportional to an increasing function of $\overline{\mathrm{H}}$. By finding the value of $\overline{\mathrm{H}}$ for which the sum of the two terms reaches its minimum and adding the regret coming from the situations where the favorable event does not hold, gives us the following cumulative regret for HCT: With probability $1-\delta$,

$$\mathrm{R}_{\mathrm{N}}^{\mathrm{HCT}(\nu,\rho)} \leq \mathscr{O}\left((\log(\mathrm{N}/\delta))^{1/(d(\nu,\mathrm{C},\rho)+2)}\mathrm{N}^{(d(\nu,\mathrm{C},\rho)+1)/(d(\nu,\mathrm{C},\rho)+2)}\right).$$

However, the analysis of POO requires a bound on the expected regret of the underlying subroutine. For that purpose, we simply set $\delta \triangleq 1/\mathrm{N}$ and that gives us the statement of Theorem 5.2, and consequently Corollary 5.3.

5.4.3 Upper bound on the simple regret of PCT

Building on our new analysis of the HCT algorithm, we are able to provide theoretical guarantees for a new algorithm instance of POO, namely the POO(HCT) algorithm, as a side result. We refer to the new algorithm instance as Parallel Confidence Tree (PCT). More precisely we define PCT(δ) as POO run on top of HCT using confidence parameter δ.

Let $(\nu^\star, C^\star, \rho^\star)$ be a triple of parameters for which Assumption 5.2 is true, we prove that PCT achieves a regret that is comparable to the one obtained by HCT.

Theorem 5.5. *Assume that the target function f satisfies Assumption 5.2 and $\nu^\star \le \nu_{\max}$ and $\rho^\star \le \rho_{\max}$. For $\delta = M(N)/N$ with $M(N) = \lceil (1/2) D_{\max} \log(N/\log N) \rceil / N$, the simple regret of PCT(δ) after N function evaluations is bounded as*

$$\mathbb{E}[S_N^{PCT(\delta)}] \le \beta D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left(((\log^2 N)/N)^{1/(d(\nu^\star, C^\star, \rho^\star)+2)} \right),$$

where β is a constant independent of $\nu_{\max}$ and $\rho_{\max}$.[3]

By Corollary 5.3, we know that the simple regret of HCT after N function evaluations run with $(\nu^\star, C^\star, \rho^\star)$ is of order $\mathcal{O}\left((\log N/N)^{1/(d(\nu^\star, C^\star, \rho^\star)+2)} \right)$. As a consequence, the performance of PCT is at most a $\sqrt{\log n}$ factor away from that of the best HCT instance.

Theorem 5.5 follows from Corollary 5.3 and Proposition 5.1 below. This wrapper result highlights how cumulative regret guarantees for *any* base algorithm translate into simple regret guarantees for the corresponding POO($\mathscr{A}$) algorithm. Its proof almost replicates the analysis of POO(HOO) by Grill et al. [2015] and we provide it in Appendix D.3 for the sake of completeness.

Proposition 5.1. *If for all (ν, ρ) the $\mathscr{A}(\nu, \rho)$ algorithm has its* cumulative regret *bounded as*

$$\mathbb{E}\left[R_n^{\mathscr{A}(\nu,\rho)} \right] \le \alpha C (\log n)^{1/(d(\nu,C,\rho)+2)} n^{(d(\nu,C,\rho)+1)/(c+2)}, \tag{5.6}$$

for any function f satisfying Assumption 5.2 with parameters (ν, C, ρ), then there exists a constant β that is independent of $\nu_{\max}$ and $\rho_{\max}$ such that

$$\mathbb{E}\left[S_n^{POO(\mathscr{A})} \right] \le \beta D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left((\log^2 n)/n \right)^{1/(d(\nu^\star, C^\star, \rho^\star)+2)},$$

for any function f satisfying Assumption 5.2 with parameters $\nu^\star \le \nu_{\max}$ and $\rho^\star \le \rho_{\max}$.

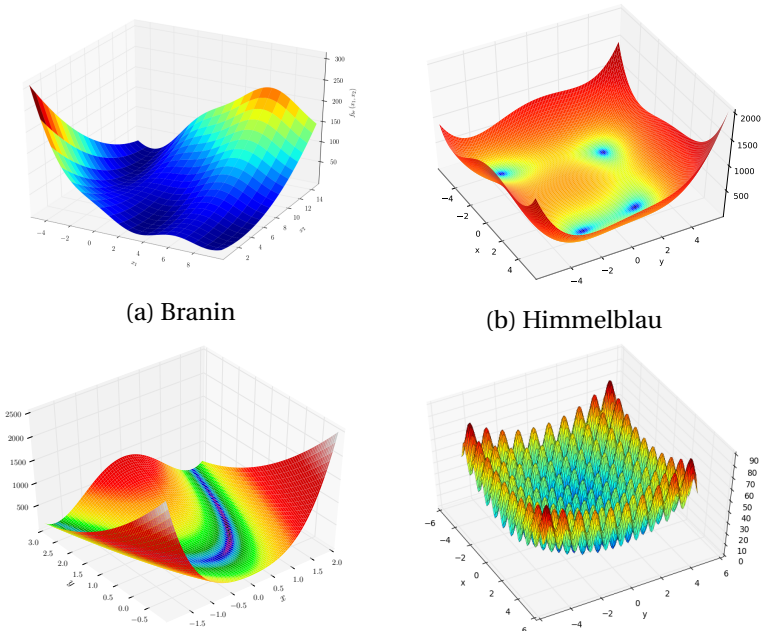
In Theorem 5.1, we provide a general analysis of the GPO algorithm, showing that it attains an (order)- optimal simple regret without knowing the parameter triple $(\nu^\star, C^\star, \rho^\star)$ provided that its base algorithm does. As a consequence GPO(HCT) is an alternative to PCT with similar simple regret guarantees.

5.5 Experimental Illustrations

In this section, we provide some simple numerical illustrations that aim to compare the performance of HCT and HOO as subroutines. We run experiments on several test functions

comparing the original POO(HOO) against our new algorithm instance PCT with different ρ values. In these experiments, we set $\rho_{\max} = 0.9$, and we add Gaussian noise to the function evaluations with a relatively small variance ($\sigma = 0.1$).

Artificial landscapes. We test the algorithms on some functions from the *artificial landscapes*[4], including (i) two functions with many local minima: Himmelblau function and Rastrigin function, (ii) one valley-shaped function: Rosenbrock function, and (iii) Branin function (see Figure 5.1). Note that the Rastrigin function shown is its 2D version. In our experiments, we use a Rastrigin function in 5D.



(a) Branin (b) Himmelblau

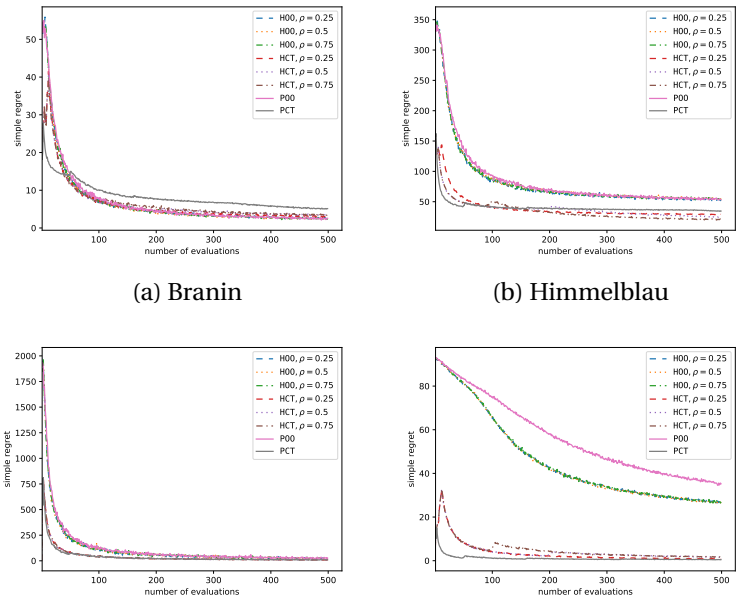
(c) Rosenbrock (d) Rastrigin in 2D

Figure 5.1: Benchmark functions for testing black-box optimization algorithms.

In Figure 5.2, we plot the simple regret of the algorithms as a function of the number of evaluations. All the results are averaged over 5000 runs and we plot the simple regret after 500 function evaluations. Each instance of HOO or HCT would recommend a point picked uniformly at random among those evaluated so that we have the same recommendation strategy as POO and PCT.

The first observation is that PCT does match the performance of some single HCT instances as expected. We also notice that PCT has comparable performance w.r.t. POO in these plots, which justifies the choice of using HCT as a subroutine for the POO meta-algorithm.

[4]Source: https://en.wikipedia.org/wiki/Test_functions_for_optimization



(a) Branin (b) Himmelblau

(c) Rosenbrock (d) Rastrigin in 5D

Figure 5.2: Simple regret of POO and PCT run for different ρ values.

5.6 Discussion

We proposed GPO, a general framework for making any hierarchical bandit algorithm that only has a simple regret guarantee adaptive to unknown smoothness. This improves over the previous framework POO that requires cumulative regret guarantee for its subroutine.

Besides, we also studied PCT, a new implementation of POO on top of HCT. We proved that HCT is a plausible subroutine for POO by adapting the analysis of HCT under a new assumption w.r.t. a fixed partitioning, and is also a valid underlying subroutine for GPO by consequence. However, whether it is possible to weaken the assumptions of HOO in the same way as HCT while keeping similar regret guarantees remains open.

A subsequent work [Bartlett et al., 2019] further proposes new algorithms that adapts to the noise. However, tree-based algorithms are well-known to suffer from high-dimensional search spaces, which impedes quite a lot the using of hierarchical bandits in practice. An important but yet unsolved problem is thus to investigate how to be adaptive to the dimension.

Chapter 6

Bandits and Hyper-Parameter Optimization

" *True optimization is the revolutionary contribution of modern search to decision processes.*

George Dantzig

Contents

6.1 Introduction 88
6.2 A Brief Survey of Automated Machine Learning 89
6.2.1 The AutoML taxonomy 89
6.2.2 Hyper-parameter optimization 91
6.3 Hyper-Parameter Optimization Framework 91
6.4 Best-Arm Identification for Hyper-Parameter Tuning 92
6.5 Active TTTS for Hyper-Parameter Optimization 93
6.6 Experiments 97
6.6.1 Some synthetic results 97
6.6.2 Experiments on real datasets 98
6.7 Adaptivity to $\mu^\star$ 100
6.8 Discussion 104

6.1 Introduction

Training a machine learning algorithm often requires to specify several parameters. For instance, for neural networks, it is the architecture of the network and also the parameters of the gradient algorithm used or the choice of regularization. These *hyper-parameters* are difficult to learn through the standard training process and are often manually specified.

When it is not feasible to design algorithms with a few hyper-parameters, we opt for HPO. HPO is a crucial component of modern machine learning and *automated machine learning* (AutoML). Recall that HPO can be viewed as a BBO/GO problem (see Chapter 1.2.4) where the evaluation of the objective function is expensive as it is the accuracy of a learning algorithm for a given configuration of hyper-parameters. Indeed, a typical function evaluation involves training the primary machine learning algorithm to completion on a dataset, which often takes a considerable amount of time or resources, in particular for large DL models. For example, the training of language representation model `BERT-Large` [Devlin et al., 2019] was performed on 16 Cloud TPUs (64 TPU chips in total), and each pre-training took 4 days to complete. This vastly limits the number of evaluations that can be carried out, which calls for a design of efficient high-level algorithms that automate the tuning procedure.

In this chapter, we are interested in exploring how MAB, or more precisely BAI, can guide the design of efficient HPO. Indeed, some bandit tools have already been employed for GO (see Chapter 5) and HPO: First, in the field of Bayesian optimization, the `GP-UCB` algorithm [Srinivas et al., 2010] is a Gaussian process extension of the classical `UCB` bandit algorithm [Auer et al., 2002a]. Later, Hoffman et al. [2014] proposed to use BAI tools – still with a Bayesian flavor – for automated machine learning, where the goal is to smartly try hyper-parameters from a pre-specified *finite* grid.

However, in most cases, the number of hyper-parameter configurations to explore is infinite. In this chapter, we investigate the use of bandit tools suited for an *infinite* number of arms. There are two lines of work for tackling a very large or infinite number of configurations (arms). The first is the continuum-armed bandits discussed in Chapter 5 (see also Bartlett et al. 2019; Bubeck et al. 2010; Grill et al. 2015; Shang et al. 2019a). It makes use of hierarchical bandit tools and aims at exploiting the (possibly unknown) smoothness of the black-box function to optimize. To the best of our knowledge, these methods have never been extensively tested in practice for HPO.

The second line of work does not assume any smoothness: At each round, the learner may ask for a new arm from a *reservoir* distribution ν_0 (pick randomly a new hyper-parameter configuration) and add it to the current arm pool $\mathscr{X}$, or re-sample one of the previous arms (evaluate configuration already included in $\mathscr{X}$), in order to find an arm with a good mean reward (i.e., a hyper-parameter configuration with a good validation accuracy). It is the *infinitely-armed bandits* setting. In particular, we study the stochastic case in which observations are assumed to be independent. The *stochastic infinitely-armed bandits* (SIAB) is studied by Berry et al. [1997]; Wang et al. [2008] for the rewards maximization problem while Aziz et al. [2018a]; Carpentier and Valko [2015] study the simple regret problem, which is related to BAI. While most proposed algorithms consist of querying an *adequate* number of arms from the reservoir before running a standard BAI algorithm, Li et al. [2017] propose a more robust approach called `Hyperband` that uses several such phases.

Contributions. **1)** In this chapter, we go even further and propose the first *dynamic* algorithm for BAI in SIAB, that at each round, may either query a new arm from the reservoir or

re-sample arms previously queried. Our algorithm leverages a Bayesian model and builds on TTTS. **2)** We also introduce a variant of `Hyperband` where the `Sequential-Halving` subroutine [Karnin et al., 2013] is replaced by TTTS. **3)** Numerical studies are presented to show the competitiveness and robustness of the proposed dynamic algorithm with respect to state-of-the-art HPO methods.

☞ This chapter is mainly based on Shang et al. [2019b] with some additional discussions from Shang et al. [2020b].

6.2 A Brief Survey of Automated Machine Learning

Although the focus of this chapter is HPO, it is worth mentioning that HPO has gradually extended to more general AutoML, for whom the goal is to optimize the entire machine learning pipeline from data preparation to model learning (see e.g. Feurer et al. 2015). This effort has led to the development of a wide variety of efficient AutoML systems in the past few years [Kotthoff et al., 2017; Mohr et al., 2018; Olson and Moore, 2019; Rakotoarison et al., 2019; Thornton et al., 2013].

In this section, we give a brief introduction of AutoML and we pay particular attention, of course, to HPO.

We first provide a full-stack pipeline for an AutoML procedure as displayed in Figure 6.1 followed by a general taxonomy on different components of AutoML [He et al., 2019; Hutter et al., 2019; Zöller and Huber, 2019].

6.2.1 The AutoML taxonomy

Data preparation. The very first step of a machine learning pipeline consists of collecting and preparing clean data.

Data collection often relies on web searching. However, data could be ill-labeled or inaccurate, thus semi-supervised or self-labeling methods are required. When not enough data are available, data synthesis is needed, mainly with the help of data augmentation.

On the other hand, data cleaning is more or less standardized. Typical tricks include normalization, scaling, binarization of numerical (discrete or continuous) attributes, one-hot encoding for categorical attributes, imputation (e.g. with mean values), etc.

Feature engineering. Once the data are collected and tidied, the next step is dealing with the features. Depending on the task, feature manipulation can contain three different aspects.

Sometimes we need to reduce irrelevant or redundant features, thus builds a more compact feature subset. This is feature selection. A typical feature selection process starts with subset generation using some (random) search strategy like simulated annealing, genetic algorithms, etc; followed by subset evaluation with filter methods, wrapper methods or embedded methods such as deep neural networks, decision trees, etc; and finally ends up with a validation step.

Another similar but not completely same trick is feature extraction. Feature extraction aims at extracting more informative features using dimensionality reduction techniques like `PCA`, `ICA`, `LDA`, etc. Recent autoencoder-based methods can be used as well.

Finally, we can also construct new features from the raw data to enhance the robustness and generalizability of the model. Typical methods of feature construction include

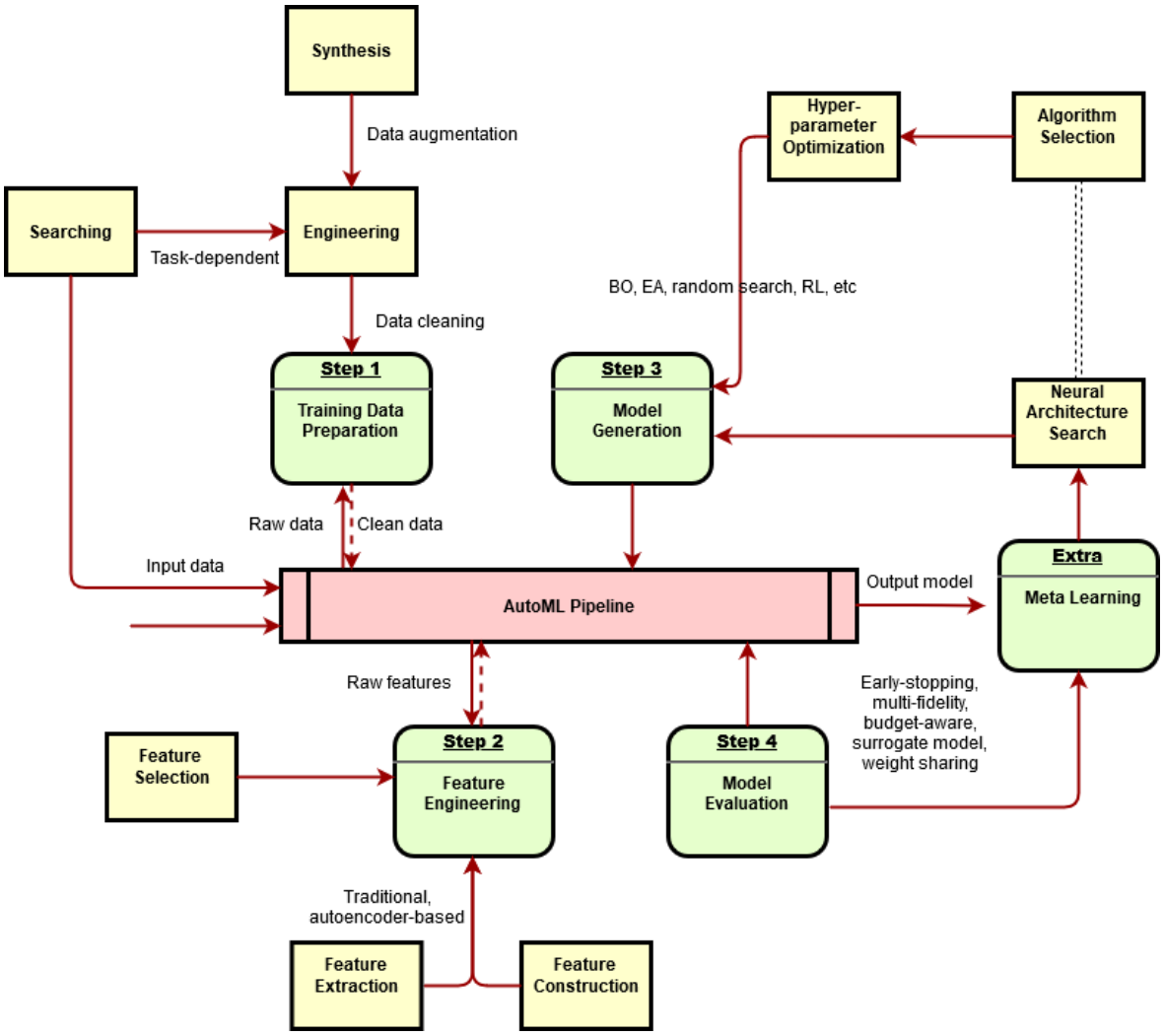


Figure 6.1: The full-stack pipeline of a machine learning task.

searching methods such as tree-based approaches, genetic algorithms; and annotation-based approaches.

Pipeline generation. The previous two parts are not the focus of this chapter. We are more interested in the third step, namely pipeline generation. Indeed HPO is one of the main research topics in this domain.

The pipeline generation is sometimes modeled as a *full model selection* (FMS) or *combined algorithm selection and hyper-parameter optimization* (CASH) problem, that is typically composed of a model/algorithm selection process and a HPO process. We focus on HPO in this thesis. It is worth noting, however, that many HPO algorithms can also be applied on the full model selection problem as a whole.

A particular instance of FMS or CASH is the *neural architecture search* (NAS) problem. NAS is specific to large modern neural-network models and has become a very hot topic recently (see e.g. Elsken et al. 2019; Kandasamy et al. 2018; Liu et al. 2019; Zoph et al. 2018). NAS can also be regarded as a HPO problem sometimes, but more often, we can make use of extra information from its inner structure. Anyhow, it is also out of the scope of this manuscript.

Model evaluation and estimation. Last but not least, an evaluation of the model is needed at the end. The classic way of evaluating a model is to wait until the completion of the training before assessing its performance on a validation set. Recent work (see e.g. Li et al. 2017) suggests that using a multi-fidelity estimation [Huang et al., 2006; Peherstorfer et al., 2018; Wu et al., 2019] can tremendously reduce the time and resource consumption. Multi-fidelity methods estimate the target function by low-resolution approximations (using only a subset of data for example).

Other methods exist as well that aim at increasing the efficacy of model evaluation and estimation, such as early-stopping, surrogate model, weight sharing, etc. They are mostly specifically designed for DL though.

6.2.2 Hyper-parameter optimization

We stress again that we only study HPO in this chapter. In particular, we do not take care of the model evaluation part, but rather focus on designing efficient model selection algorithms. That being said, we always consider complete training in this thesis, and we do not restrict ourselves to DL.

Two naive but daily-used HPO methods are `Grid-Search` and `Random-Search`. More sophisticated model-free methods address HPO as a sequential resource allocation problem, by adaptively choosing the next hyper-parameter(s) to explore, based on the results obtained previously. For example, evolutionary optimization follows a process inspired by the biological concept of evolution, which repeatedly replaces the worst-performing hyper-parameter configurations from a randomly initialized population of configurations (see e.g. Loshchilov and Hutter 2016) for an example of using `CMA-ES` for hyper-parameter tuning. A major drawback of evolutionary optimization is its lack of theoretical understanding.

Model-based approaches also exist. For example, Bayesian optimization is an approach that leverages the sequential nature of the setting. BO depends on a prior belief for the target function, typically a Gaussian process. This prior distribution can be updated to a posterior given a sequence of observations. Several algorithms exploiting this posterior distribution to decide where to sample next have been given (see e.g. Shahriari et al. 2016, for a survey). Snoek et al. [2012] and Klein et al. [2017] provide Python packages called `Spearmint` and `RoBO` to perform hyper-parameter tuning with BO methods. Similar packages are available for `PyTorch` (`BoTorch`[1]) and `TensorFlow` (`GPflowOpt` by Knudde et al. 2017). Among BO algorithms, `TPE` [Bergstra et al., 2011] and `SMAC` [Hutter et al., 2011] were specifically proposed for HPO. A shortcoming of BO is that most algorithms select where to sample next based on optimizing some *acquisition function* computed from the posterior, e.g., the expected improvement [Jones et al., 1998]. This auxiliary task cannot be solved analytically but needs to be performed itself by optimization procedures as `L-BFGS` that make the process slow.

6.3 Hyper-Parameter Optimization Framework

In this chapter, we view HPO as a particular GO setting, for which the target function f is a mapping from a hyper-parameter configuration to some measure of failure for the machine learning algorithm trained with these hyper-parameters. Formally, we aim at

[1] `https://botorch.org/`

solving an optimization problem of the form

$$f^\star = \min\left\{f(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \Omega\right\},$$

where $\boldsymbol{\lambda}$ denotes a configuration of hyper-parameters chosen from a configuration space Ω. A hyper-parameter optimizer is a sequential procedure, that at each round n, selects a configuration $\boldsymbol{\lambda}_n$ to evaluate using some sampling rule, after which a (costly and *noisy*) evaluation of $f(\boldsymbol{\lambda}_n)$ is observed. Besides, a hyper-parameter configuration $\widehat{\boldsymbol{\lambda}}^\star$ is recommended as a guess for a close-to-optimal configuration at the end. The hope is that $f(\widehat{\boldsymbol{\lambda}}^\star)$ is not far from $f^\star$.

We restrict our attention to hyper-parameter tuning for supervised learning algorithms. Given a training dataset $\mathscr{D}_{\texttt{train}}$ containing m labeled examples in $\mathscr{X} \times \mathscr{Y}$ and a choice of hyper-parameter configuration $\boldsymbol{\lambda}$, a supervised learning algorithm (neural network, SVM, gradient boosting, ...) produces a predictor $\widehat{g}_{\boldsymbol{\lambda}}^{(m)} : \mathscr{X} \to \mathscr{Y}$. Note that there can be some randomness in the training process (e.g., if stochastic gradient descent is used) so that $\widehat{g}_{\boldsymbol{\lambda}}^{(m)}$ may still be random for a given training set and hyper-parameters. The goal is to build a predictor that generalizes well. If we had access to the distribution $\mathbf{P}$ that generated the data (i.e., assuming that data points in $\mathscr{D}_{\texttt{train}}$ are i.i.d. from $\mathbf{P}$), this generalization power would be measured by the risk $f(\boldsymbol{\lambda}) \triangleq \mathbb{E}\left[\ell\left(\mathbf{Y}, \widehat{g}_{\boldsymbol{\lambda}}^{(m)}(\mathbf{X})\right)\right]$, where ℓ is some loss function measuring the distance between two predictions and the expectation is taken on $(\mathbf{X}, \mathbf{Y}) \sim \mathbf{P}$ and the possible randomness in the training process.

In practice, however, the explicit evaluation of f is impossible, but there are several methods for *noisy evaluations*. We can either compute the validation error of $\widehat{g}_{\boldsymbol{\lambda}}^{(n)}$ on a held-out validation set,

$$\frac{1}{|\mathscr{D}_{\texttt{valid}}|} \sum_{i=1}^{|\mathscr{D}_{\texttt{valid}}|} \ell(\widehat{g}_{\boldsymbol{\lambda}}^{(m)}(\mathbf{x}_i), \mathbf{y}_i),$$

or a cross validation error over the training set as an approximation of the objective.

6.4 Best-Arm Identification for Hyper-Parameter Tuning

HPO with pre-defined set of hyper-parameter configurations can be modeled as a BAI game. Given a finite set of arms $\mathscr{X} \triangleq \{1, \ldots, \mathrm{K}\}$, when we select arm i, we get an independent observation from some unknown distribution ν_i with mean μ_i. A BAI algorithm sequentially selects arms in order to identify the arm with the largest mean[2], $\mathrm{I}^\star \triangleq \arg\max_{i \in \mathscr{A}} \mu_i$.

In the context of HPO, each arm models the quality of a given hyper-parameter configuration $\boldsymbol{\lambda}$. When the arm is sampled, a noisy evaluation of $f(\boldsymbol{\lambda})$ is received, which is the mean reward of that arm.

As stated in Section 6.1, standard BAI algorithms are not straightforwardly applicable to HPO when the search space can be infinite and is often continuous. To handle such cases, we rather turn our attention to SIAB. In this context, there is an infinite pool of arms, whose means are assumed to be drawn from some *reservoir* distribution ν_0. In such a model, an algorithm maintains a list of arms that have been tried before. At each round it can either query a new arm from the reservoir, add it to the list and sample it, or sample an arm already in the list.

[2]Here we present BAI problems in a standard way for which we search for an arm with the largest mean. For HPO, however, it is important to mention that we are searching for a hyper-parameter configuration that minimizes the validation error. One can easily see that it does not change the problem in principle.

A natural way to perform BAI in an infinite-many armed bandit model consists of first querying a *well-chosen* number of arms from the reservoir and then running a standard BAI algorithm on those arms [Carpentier and Valko, 2015]. However this ideal number may rely on the difficulty of the learning task, which is hardly known in practice. The `Hyperband` algorithm [Li et al., 2017] takes a step further and successively queries several batches of arms from the reservoir, including a decreasing number of arms in each batch, while increasing the budget dedicated to each of them. `Sequential-Halving` [Karnin et al., 2013], a state-of-the-art fixed-budget BAI algorithm, is then run on each of these batches of arms. This approach seems more robust in that it trades off between *the number of arms that is needed to capture a good arm* and *how much measurement effort we should allocate to each of them.* However, a numerical study performed by Aziz et al. [2018b] seems to reveal that an infinite bandit algorithm based on `Sequential-Halving` should always query the maximal number of arms from the reservoir[3].

In Table 6.1, we summarize how to cast HPO as a BAI problem with infinitely-many arms.

BAI	HPO
query ν_0	pick a new configuration $\boldsymbol{\lambda}^{\cdot}$
sample an arm$^{\cdot}$	train the classifier $g_{\boldsymbol{\lambda}}$
reward	cross-validation loss

Table 6.1: Casting HPO as a BAI problem.

All existing algorithms are still subject to a pre-defined scheduling of how many arms should be queried from the reservoir. The algorithm (`D-TTTS`) that we propose in the next section does not need to decide in advance how many arms will be queried, and is therefore fully *dynamic*.

Remark 6.1. *`Hyperband` is proposed specifically for hyper-parameter tuning. Its original philosophy is to adaptively allocate* resources *to more promising configurations. Resources here can be time, dataset sub-sampling, feature sub-sampling, etc. In such a setting, the classifier is not always trained into completion given a parameter configuration, but is rather stopped early if it is shown to be bad so that we can allocate more resources to other configurations. In this case, different evaluations of a single configuration cannot be considered as i.i.d. anymore. Thus, HPO is stated as a non-stochastic infinitely-armed bandit problem. This idea of early stopping is also further investigated by combining Bayesian optimization with it [Falkner et al., 2018]. However, this is about the model evaluation as defined in Section 6.2.2 and is out of the scope of this thesis.*

6.5 Active `TTTS` for Hyper-Parameter Optimization

In this section, we introduce a new algorithm for BAI in an infinite bandit model, that is an adaptation of `TTTS` (see Chapter 3). Unlike `Sequential-Halving` that requires the

[3]This reference is a preliminary draft that has been withdrawn due to technical issues in the proofs. Yet we believe the experimental section to be sound.

knowledge of the total budget to operate, TTTS is particularly appealing as it does not need to have it. Remember that such algorithms are referred to as *anytime*. Besides, it is known to be optimal in a Bayesian (asymptotic) sense (see Chapter 3).

Recall that as a Bayesian algorithm, TTTS uses a prior distribution Π_0 over the vector of means of the K arms, $\boldsymbol{\mu} \triangleq (\mu_1, \cdots, \mu_K)$, which can be updated to a posterior distribution Π_n after n observations.

We consider Bernoulli bandit model in the rest of this chapter. Under Bernoulli bandit model, arm i produces a reward $r_{n,i} = 1$ with probability μ_i, and $r_{n,i} = 0$ with probability $1 - \mu_i$ when sampled at round n. Given independent uniform prior for the mean of each arm, the posterior distribution on $\boldsymbol{\mu}$ is a product of K Beta distributions: $\Pi_n = \bigotimes_{i=1}^{K} \texttt{Beta}(1 + S_{n,i}, N_{n,i} - S_{n,i} + 1)$, where $N_{n,i}$ is the number of selections of arm i until round n and $S_{n,i}$ is the sum of rewards obtained from that arm.

Why variants of TTTS? We further motivate experimentally in this section why we choose to build new algorithms upon TTTS.

We compare TTTS against some fixed-budget BAI algorithms as benchmark, including uniform allocation [Bubeck et al., 2009], UCB-E and Successive-Reject [Audibert and Bubeck, 2010], UGapE [Gabillon et al., 2012], Sequential-Halving, TS with a MPA strategy of decision (see Section 2.2.4), and one anytime algorithm AT-LUCB [Jun and Nowak, 2016].

We use 8 problem instances proposed by Audibert and Bubeck [2010], all settings consider Bernoulli bandits, and we compare their trending *simple regret* (see Definition 2.16) averaged on 1000 trials. The results are shown in Fig. 6.2.

Note that contrary to Chapter 3, we are interested in the fixed-budget setting in this chapter, hence the present experimental study with simple regret as performance measure.

- Setting 1: $\mu_1 = 0.5, \mu_{2:20} = 0.4$, budget = 2000
- Setting 2: $\mu_1 = 0.5, \mu_{2:6} = 0.42, \mu_{7:20} = 0.38$, budget = 2000
- Setting 3: $\mu = [0.5, 0.3631, 0.449347, 0.48125839]$, budget = 2000
- Setting 4: $\mu = [0.5, 0.42, 0.4, 0.4, 0.35, 0.35]$, budget = 600
- Setting 5: $\mu_1 = 0.5, \mu_i = \mu_1 - 0.025i, \forall i \in \{2 \dots 15\}$, budget = 4000
- Setting 6: $\mu_1 = 0.5, \mu_2 = 0.48, \mu_{3:20} = 0.37$, budget = 6000
- Setting 7: $\mu_1 = 0.5, \mu_{2:6} = 0.45, \mu_{7:20} = 0.43, \mu_{7:20} = 0.38$, budget = 6000
- Setting 8: $\mu_1 = 0.5, \mu_{2:6} = 0.45, \mu_{7:20} = 0.43, \mu_{7:20} = 0.38$, budget = 12000

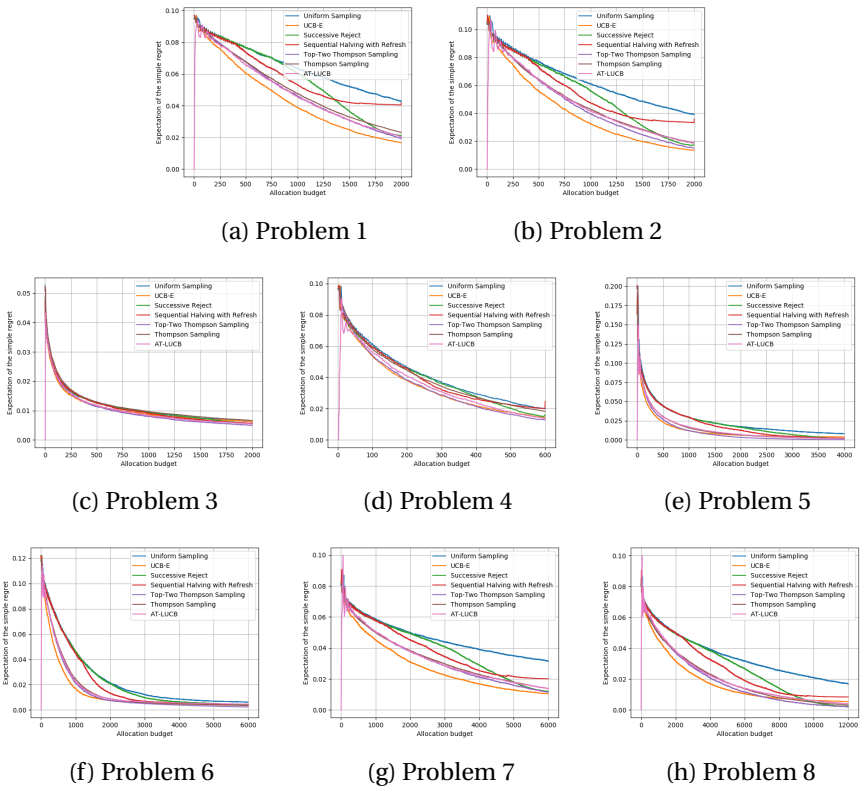
In these experiments, TTTS are always beating or at least performing as well as its competitors. It thus seems to be a good candidate to be further investigated.

Note that TTTS can also be used for bandit settings in which the rewards are bounded in $[0,1]$ by using a binarization trick first proposed by Agrawal and Goyal [2012]: When a reward $r_{n,i} \in [0,1]$ is observed, the algorithm is updated with a fake reward

$$r'_{n,i} \sim \texttt{Bern}(r_{n,i}) \in \{0,1\}.$$

TTTS can thus be used for BAI for a *finite* number of arms that with rewards in $[0,1]$. We now present a simple way of extending TTTS to deal with an infinite number of arms, namely Dynamic Top-Two Thompson Sampling (D-TTTS).

Dynamic TTTS. In an infinite bandit algorithm, at each round, we either query a new arm from the reservoir *and sample it*, or re-sample a previous arm. In a Bayesian setting, we can also imagine that at each round, an arm is queried from the reservoir and added



(a) Problem 1 (b) Problem 2

(c) Problem 3 (d) Problem 4 (e) Problem 5

(f) Problem 6 (g) Problem 7 (h) Problem 8

Figure 6.2: Simple regret as a function of allocation budget for various BAI algorithms.

with a *uniform prior* to the list of queried arms, *regardless of whether it is sampled or not.* Then, at round t, D-TTTS consists in running TTTS on these t arms, out of which several are endowed with a uniform prior and have never been sampled.

Leveraging the fact the the maximum of k uniform distribution has a $\texttt{Beta}(k,1)$ distribution and that TTTS only depends on the maxima of posterior samples, we give the following equivalent implementation for D-TTTS (Algorithm 6.1). Letting $\mathscr{L}_n$ be the list of arms that have been queried from the reservoir and sampled *at least once* before round t, at round t we run TTTS on the set $\mathscr{X}_n \triangleq \mathscr{L}_n \cup \{\mu_0\}$ where μ_0 is a pseudo-arm with posterior distribution $\texttt{Beta}(n-k_n,1)$, where $k_n \triangleq |\mathscr{L}_n|$.

It remains to decide how to recommend the arm as our best guess. It is obviously not a good idea to output the arm with the best empirical means since some lately sampled arms may have very high empirical mean with no confidence. In this chapter, we choose the most natural recommendation strategy for Bayesian algorithms that outputs the arm with the largest optimal action probability (see Section 3.2). Let Θ_i be the subset of the set Θ of possible mean vectors such that arm i is optimal, $\Theta_i \triangleq \left\{\boldsymbol{\theta} \in \Theta \,|\, \theta_i > \max_{j \neq i} \theta_j\right\}$, the posterior probability that arm i is optimal after round t is defined as $\Pi_n(\Theta_i)$. At any time n, we therefore recommend arm

$$J_n \triangleq \arg\max_{i \in \mathscr{X}} \Pi_n(\Theta_i)\,.$$

Hyper-TTTS. We present here also another simple way of extending TTTS to deal with an infinite number of arms, namely Hyper-TTTS or H-TTTS, a variant of Hyperband in which SHA is replaced by TTTS. This algorithm, whose sampling rule is formally stated as Algorithm 6.2, runs $s_{\max}$ batches of TTTS with different number of arms n and each batch with a same budget $T = \lceil B/s_{\max} \rceil$ with B the total budget. The number of arms within each

Algorithm 6.1 Sampling rule of Dynamic D-TTTS

Input: β; B (total budget); ν_0
Initialization: $\mu_1 \sim \nu_0$; $t \leftarrow 0$; $\mathcal{X} \leftarrow \{\mu_0, \mu_1\}$; $m \leftarrow 1$; $S_0, N_0 \leftarrow 0$; $S_1 \sim \texttt{Bern}(\mu_1)$, $N_1 \leftarrow 1$
while $n < B$ **do**
  $\forall i = 0, \dots, m$, $\theta_i \sim \texttt{Beta}(S_i + 1, N_i - S_i + 1)$; $U \sim \mathcal{U}([0,1])$
  $I^{(1)} \leftarrow \arg\max_{i=0,\dots,m} \theta_i$
  if $U > \beta$ **then**
    while $I^{(2)} = I^{(1)}$ **do**
      $\forall i = 0, \dots, m, \theta'_i \sim \texttt{Beta}(S_i + 1, N_i - S_i + 1)$
      $I^{(2)} \leftarrow \arg\max_{i=0,\dots,m} \theta'_i$
    end while
    $I^{(1)} \leftarrow I^{(2)}$
  end if
  if $I^{(1)} \neq 0$ **then**
    $Y \leftarrow$ Evaluate arm $I^{(1)}$; $X \sim \texttt{Bern}(Y)$
    $S_{I^{(1)}} \leftarrow S_{I^{(1)}} + X$; $N_{I^{(1)}} \leftarrow N_{I^{(1)}} + 1$; $S_0 \leftarrow S_0 + 1$
  else
    $\mu_{m+1} \sim \nu_0$; $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mu_{m+1}\}$;
    $Y \leftarrow$ Evaluate arm $m+1$; $X \sim \texttt{Bern}(Y)$
    $S_{m+1} \leftarrow X$; $N_{m+1} \leftarrow 1$; $m \leftarrow m+1$
  end if
  $t \leftarrow t+1$
end while

bracket is decreasing with an exponential rate of γ. One inconvenience of this algorithm is that $s_{\max}$ and γ still need to be tuned (in practice, we use the same tuning as the one of Hyperband). D-TTTS is thus proposed to circumvent this issue.

Algorithm 6.2 Sampling rule of H-TTTS

Input: β; γ; B; $s_{\max}$; ν_0
Initialization: $T = \lfloor B/s_{\max} \rfloor$
for $s \leftarrow s_{\max}$ to 0 **do**
  $K = \lceil \frac{s_{\max}+1}{s+1} \gamma^s \rceil$
  $\mathcal{X} \leftarrow \{i = 1, \dots, K : \mu_i \sim \nu_0\}$; $t = 0$
  while $t < T$ **do**
    Sample $\boldsymbol{\theta} \sim \Pi_n$
    $I^{(1)} \leftarrow \arg\max_{i \in \mathcal{X}} \theta_i$
    Sample $b \sim$ Bernoulli(β)
    if $b = 1$ **then**
      $Y \leftarrow$ Evaluate arm $I^{(1)}$
    else
      while $I^{(2)} = I^{(1)}$ **do**
        $\forall i \in \mathcal{X}, \theta'_i \sim \texttt{Beta}(S_i + 1, N_i - S_i + 1)$
        $I^{(2)} \leftarrow \arg\max_{i \in \mathcal{X}} \theta'_i$
      end while
      $I^{(1)} \leftarrow I^{(2)}$
      $Y \leftarrow$ Evaluate arm $I^{(1)}$
    end if
    $X \sim \texttt{Bernoulli}(Y)$
    $S_{I^{(1)}} \leftarrow S_{I^{(1)}} + X$; $N_{I^{(1)}} \leftarrow N_{I^{(1)}} + 1$
    $t = t + 1$
  end while
end for

6.6 Experiments

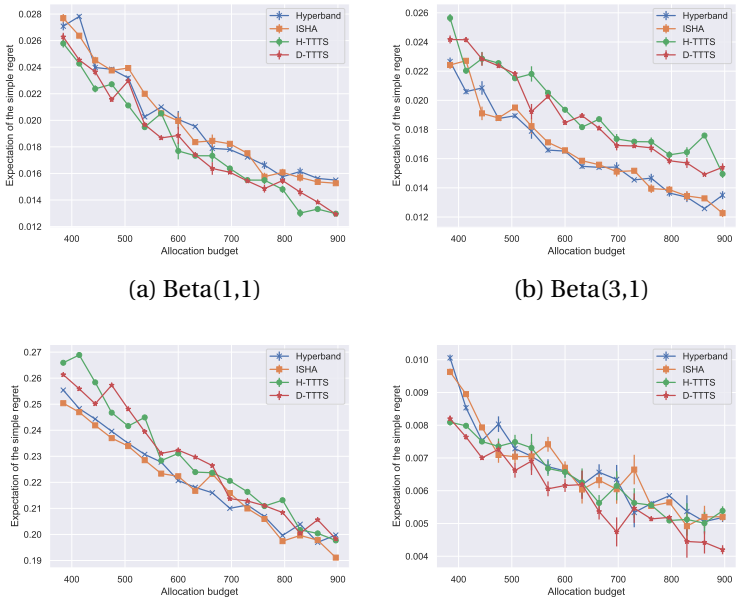
6.6.1 Some synthetic results

We first provide some synthetic experimental results comparing D-TTTS to Hyperband and ISHA. For these experiments, the arms are Bernoulli distributed and the reservoir distribution ν_0 is fixed to some Beta(a, b) distribution.

ISHA is the extension of Sequential-Halving to the SIAB setting. It consists in running Sequential-Halving on a fixed number of arms drawn from the reservoir. Observe that for a total budget B, there exists a maximum number of arms $K^\star$ that can be processed by Sequential-Halving, which satisfies $B = \lceil K^\star \log_2(K^\star) \rceil$. Following Aziz et al. [2018b], we run ISHA with $K^\star$ arms drawn from the reservoir.

We report in Fig. 6.3 the simple regret as a function of time for different algorithms and four Beta reservoir distributions. H-TTTS and D-TTTS are run with $\beta = 1/2$ which is known to be a robust choice [Russo, 2016]. Each point represents the expected simple regret $\mathbb{E}[1 - \mu_{I^*_n}]$ estimated over 1000 replications for an algorithm run with budget n. D-TTTS is very competitive on 3 reservoirs and H-TTTS is sometimes better, sometimes worse than Hyperband. We also tried the SiRI algorithm [Carpentier and Valko, 2015] (with b as the tail parameter when ν_0 =Beta(a, b)) but obtained worse performance and therefore do not report the results.

Note that in the implementation of Hyperband for this *stochastic* infinite bandit setting, the elimination phase of the underlying Sequential-Halving algorithm is carried out according to the averaged loss of previous samples (as samples from an arm are i.i.d.



(a) Beta(1,1)

(b) Beta(3,1)

(c) Beta(1,3)

(d) Beta(0.5,0.5)

Figure 6.3: Simple regret of D-TTTS (against Hyperband) as a function of the number of arms evaluations for different Beta reservoir.

in this setting and not a converging sequence). In the next section, we apply our algorithm to some real hyper-parameter optimization tasks.

6.6.2 Experiments on real datasets

We now benchmark our bandit-based strategy against different types of HPO algorithms, namely, TPE, random search, Hyperband and H-TTTS, for the tuning of classifiers (SVM and MLP) on 4 different classification tasks: *wine*, *breast cancer*, and *adult* datasets from UCI machine learning repository [Dua and Taniskidou, 2017]; and the MNIST dataset [LeCun et al., 1998].

For all the methods, a noisy evaluation of the black-box function f (see the terminology introduced in Section 6.3) for a hyper-parameter configuration $\boldsymbol{\lambda}$ consists in performing a shuffled 3-fold cross-validation on $\mathcal{D}_{\texttt{train}}$. More precisely, given a random partitioning $\cup_{j=1}^{3}\mathcal{D}^{j}_{\texttt{valid}}$ of $\mathcal{D}_{\texttt{train}}$, where the folds are of equal size, we train a classifier $\widehat{g}^{(j)}_{\boldsymbol{\lambda}}$ on $\mathcal{D}_{\texttt{train}}\backslash\mathcal{D}^{j}_{\texttt{valid}}$ for each fold j and compute the average validation error defined as

$$e \triangleq 1/|\mathcal{D}_{\texttt{train}}| \sum_{j=1}^{3} \sum_{i\in\mathcal{D}^{j}_{\text{valid}}} \mathbb{1}\{\widehat{g}^{(j)}_{\boldsymbol{\lambda}}(\mathbf{x}_i) \neq \mathbf{y}_i\},$$

which we report as a noisy estimate of the risk

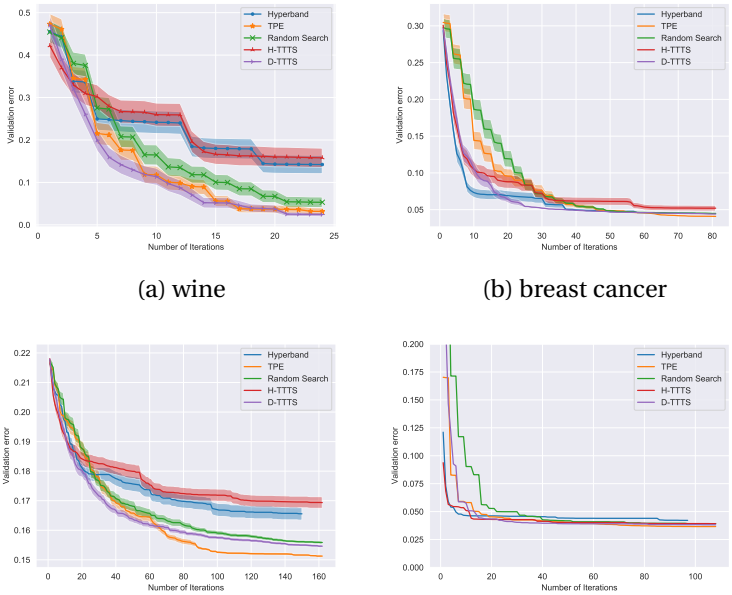
$$f(\boldsymbol{\lambda}) \triangleq \mathbf{P}(\widehat{g}^{(n)}_{\boldsymbol{\lambda}}(\mathbf{X}) \neq \mathbf{Y}),.$$

Observe that both the noisy evaluation and the value of f belong to $[0,1]$. Therefore we can introduce an *arm* with rewards in $[0,1]$ for each hyper-parameter $\boldsymbol{\lambda}$. Sampling arm $\boldsymbol{\lambda}$ produces reward $r \triangleq 1-e \in [0,1]$ with a different random partitioning and random seed for training for each selection. Arm λ is assumed to have mean of $1-f(\boldsymbol{\lambda})$. In an infinite

arm setting, querying a new arm from the reservoir corresponds to selecting a new hyper-parameter at random from the search space. With these two notions (*arm sampling* and *reservoir querying*), our algorithm for infinite BAI applies to HPO.

For the experiments, we adapt the recommendation rule of D-TTTS to the HPO applications considered and always recommend the hyper-parameter configuration that has produced the smallest cross-validation error so far (which is also the recommendation rule used by other approaches, e.g., Hyperband). For all methods, we report the cross-validation error for the recommended hyper-parameter configuration, as a function of time. We stress again that, unlike in standard bandits, where we could use the simple regret as a performance metric, we do not have access to the ground truth generalization error in real classification tasks. Therefore, we only report a proxy of the true error rate that we are interested in.

Results. We first benchmark[4] our methods on a few simple UCI datasets using SVM from scikit-learn as the classifier. We optimize over two hyper-parameters: the *penalty parameter* C and the *kernel coefficient* γ[5] for an RBF kernel, for which the pre-defined search bounds are both $[10^{-5}, 10^{5}]$.



(a) wine (b) breast cancer

(c) adult (d) MNIST

Figure 6.4: Mean cross-validation error of different HPO algorithms with (a) SVM run on the UCI wine dataset, (b) SVM run on the UCI breast cancer dataset, (c) SVM run on the UCI adult dataset and (d) MLP run on the MNIST dataset.

Fig. 6.4a shows the mean cross-validation error of SVM run on the UCI wine dataset over 24 pulls[6] averaged on 100 runs. The task is to predict the quality score of wine (between 0 and 10) given 11 attributes. Recall that one iteration corresponds to one arm pull. In this experiment, D-TTTS improves over other benchmark algorithms. Fig. 6.4b is the same experiment run on the UCI breast cancer dataset over 81 pulls. The task is to predict

[4] Code at http://researchers.lille.inria.fr/~valko/hp/publications/shang2019simple.code.zip

[5] γ is the parameter of the RBF kernel defined as $\exp(-\gamma||\mathbf{x}-\mathbf{x}'||^2)$

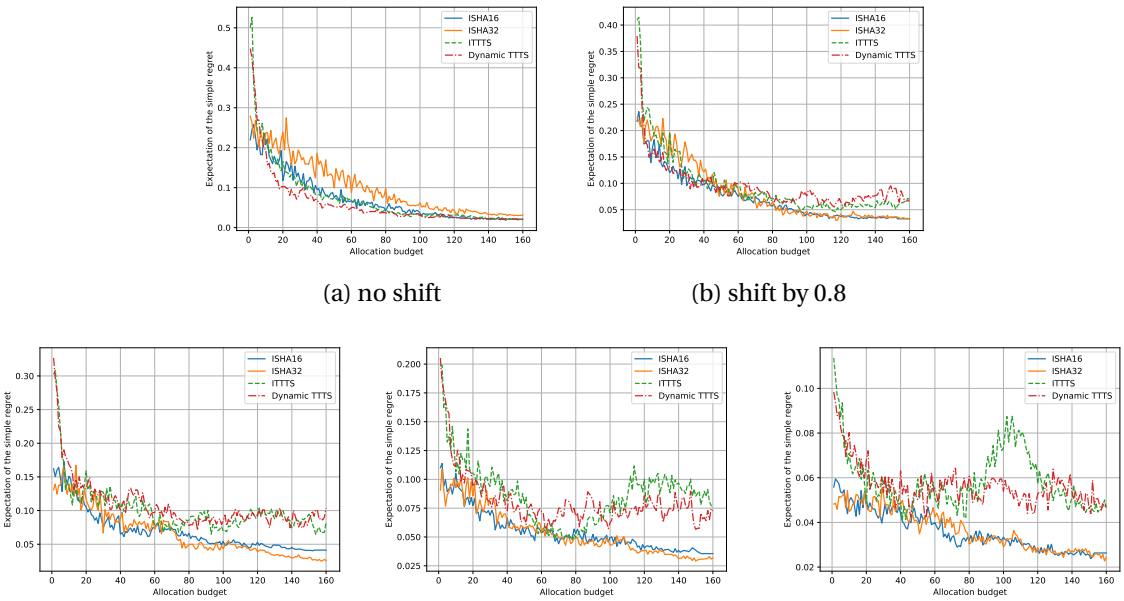
[6] The number of pulls here and later is chosen exactly as in the work of Li et al. [2017]

whether a patient has breast cancer based on 32 attributes. We repeat the experiment 100 times. This time, D-TTTS is slightly worse than Hyperband at the beginning, but improves later. Finally, we optimize SVM on a relatively more complicated UCI adult dataset over 162 pulls, for which the result is shown in Fig. 6.4c. The task is to tell whether the income of an individual is higher than 50k or not given 14 attributes. This experiment is also averaged over 100 runs. D-TTTS is better than other algorithms at the beginning, but is outperformed by TPE towards the end. We see that, although not always the best, D-TTTS shows a consistent, robust, and quite competitive performance in the 3 tasks.

We now carry out the classic MNIST digits classification task using multi-layer perceptron (MLP). We choose to optimize over three hyper-parameters: the *size of hidden layer* (an integer between 5 and 50), the ℓ_2 *penalty parameter* α (between 0 and 0.9) and the *initial learning rate* (bounded in $[10^{-5}, 10^{-1}]$). Fig. 6.4d shows the result of MLP run on MNIST over 108 pulls, this time averaged over 20 runs. D-TTTS is slightly worse than Hyperband and H-TTTS in the very beginning, but is performing well afterward.

6.7 Adaptivity to $\mu^\star$

One drawback of the present D-TTTS is that it may not work well if we do not know the oracle $\mu^\star$ ($\mu^\star$ is set to 1 in our previous experiments). Fig. 6.5 shows the expected simple regret of D-TTTS compared to ISHA and TTTS under a Beta(0.5, 0.5) reservoir shifted by 0.8, 0.6, 0.4, 0.2 and without shift respectively. A Beta distribution Beta(a, b) shifted by $\mu^\star$ is obtained by re-scaling to $[0, \mu^*]$ the corresponding distribution. More formally, a shifted Beta distribution on $[0, \mu^*]$, denoted by $\mathrm{SB}_{\mu^\star}(a, b)$ in the rest of the paper, is the distribution of $X\mu^*$ where $X \sim \mathrm{Beta}(a, b)$ (see Appendix 6.7 for more discussion on shifted Beta distributions). We can see that the performance of D-TTTS is getting worse along with the increasing shift.



(a) no shift (b) shift by 0.8

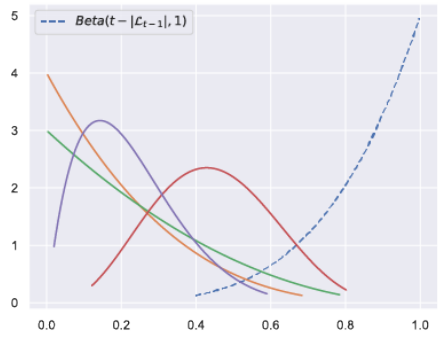
(c) shift by 0.6 (d) shift by 0.4 (e) shift by 0.2

Figure 6.5: Simple regret of D-TTTS (against Hyperband) for shifted Beta reservoir.

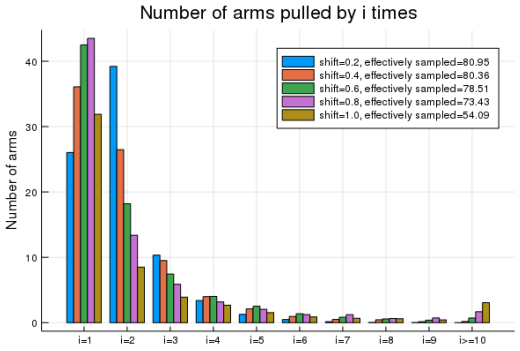
As suggested by the implementation trick introduced in Section 6.5, all the k arms that have been added but not effectively sampled can be seen as a virtual arm endowed with a

$\texttt{Beta}(k,1)$ posterior. Intuitively, if $\mu^\star < 1$, than this virtual arm would force the algorithm to sample too many new arms, thus would lack of attention on arms that are more likely to be near-optimal. This intuition is supported by the illustration in Fig. 6.6a: the posterior distributions of effectively sampled will eventually be supported mostly on the left of μ^*, while the pseudo-arm still put a lot of mass near 1.

In Fig. 6.6b, we report the number of arms that have been played $1,2,\ldots,9$ and more than 10 times for D-TTTS run under $\texttt{Beta}(0.5,0.5)$, $\mathrm{SB}_{0.8}(0.5,0.5)$, $\mathrm{SB}_{0.6}(0.5,0.5)$, $\mathrm{SB}_{0.4}(0.5,0.5)$, $\mathrm{SB}_{0.2}(0.5,0.5)$ reservoir respectively, which confirms the over-exploration effect caused by shifted reservoirs.



(a) posterior distributions of the effectively samples arms and the pseudo-arm



(b) number of effectively sampled arms, averaged over 100 runs

Figure 6.6: Illustration of over-exploration of D-TTTS under shifted reservoirs.

We now propose a natural extension of D-TTTS to overcome the present issue. In this section we assume that we have the knowledge of the maximum mean $\mu^\star$ of the reservoir. The core idea is to keep the same algorithm but with a different prior distribution over each queried arm, that is supported on $[0,\mu^\star]$ instead of $[0,1]$.

Bernoulli bandits. We still assume a Bernoulli bandit model for the rewards (although the algorithm is extended to any rewards bounded in $[0,1]$ with the binarization trick): An arbitrary arm produces at time t a reward 1 with probability θ and a reward 0 with probability $1-\theta$. The likelihood can be written as follow:

$$p(s|\theta) = \theta^s(1-\theta)^{1-s}; s \in \{0;1\}.$$

Sample from the shifted posterior. In order to implement the extension of D-TTTS, we need to know how to sample from the "shifted" posterior, that is the posterior assuming a uniform prior over $[0,\mu^*]$ instead of $[0,1]$. We now explain how to compute this posterior distribution on θ given a sequence of observations $Y_1, Y_2, \cdots, Y_N \in \{0;1\}$. Define

$$\begin{cases} a &= \sum_{i=1}^N Y_i + 1 \\ b &= N - \sum_{i=1}^N Y_i + 1, \end{cases}$$

then, according to the Bayes rule, we have

$$\begin{aligned} p(\theta|Y_1,\cdots,Y_N) &= \frac{p(Y_1,\cdots,Y_N|\theta)p(\theta)}{p(Y_1,\cdots,Y_N)} \\ &= \frac{p(Y_1,\cdots,Y_N|\theta)p(\theta)}{\int_0^1 p(Y_1,\cdots,Y_N|\theta')p(\theta')\mathbb{1}_{[0,\mu^\star]}(\theta')d\theta'} \\ &= \frac{\theta^{a-1}(1-\theta)^{b-1}\mathbb{1}_{[0,\mu^\star]}(\theta)/B(a,b)}{\int_0^{\mu^\star}(\theta')^{a-1}(1-\theta')^{b-1}/B(a,b)d\theta'} \\ &= \frac{\theta^{a-1}(1-\theta)^{b-1}\mathbb{1}_{[0,\mu^\star]}(\theta)}{B(a,b)F_{a,b}(\mu^\star)}, \end{aligned}$$

where $F_{a,b}$ is the *cumulative distribution function* (cdf) of $\texttt{Beta}(a,b)$. Thus the cdf of the posterior is

$$\mathbb{P}[\theta \le x|Y_1,\cdots,Y_N] = \frac{F_{a,b}(x)}{F_{a,b}(\mu^\star)} \triangleq G(x).$$

Now the sampling is quite straightforward as $G^{-1}(u)$ can be computed as

$$G^{-1}(u) = F^{-1}_{a,b}(u * F_{a,b}(\mu^\star)).$$

The computation of $F^{-1}_{a,b}$ and $F_{a,b}$ is easily accessible via existing libraries in different programming languages, and we can thus apply *inverse transform sampling* to obtain the observations, since if $U \sim \mathcal{U}([0,1])$, then $G^{-1}(U)$ follows the posterior distribution.

Shifted Beta distribution. Recall that we defined a shifted Beta distribution $\texttt{SB}_{\mu^\star}(a,b)$ as the distribution of the random variable $\theta' \triangleq \mu^\star\theta$, where a,b are the shape hyper-parameters of the Beta distribution and $\theta \sim \texttt{Beta}(a,b)$. The *probability density function* (pdf) of $\texttt{SB}_{\mu^\star}(a,b)$ can be written as

$$p(\theta') = \frac{1}{B(a,b)}\frac{(\theta')^{a-1}(\mu^\star-\theta')^{b-1}}{(\mu^\star)^{a+b-1}},$$

via the transformation $\theta' = \mu^\star\theta$. Here B is the Beta function[7].

The previous expression is particularly useful if we want to use the same efficient implementation trick that we employed in Algorithm 6.1, namely the order statistic trick.

Order statistic trick. Now we show that an "order statistic trick" still exists under a uniform prior over $[0,\mu^*]$, namely that the maximum of k random variables drawn from this prior distribution still has a nice distribution.

Given n random variables $X_1, X_2, \cdots, X_n$, the order statistics $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$ are also random variables, defined by sorting the values of $X_1, X_2, \cdots, X_n$ in an increasing order. In this section we treat the special case where they are i.i.d samples from the same distribution with a cdf. F_X. Following Gentle [2009], Chapter 1 Section 7, we know that the cumulative distribution function of the k-th order statistic can be written as follow:

$$F_{X_{(k)}}(x) = \sum_{j=k}^{n} (F_X(x))^j (F_X(x))^{n-j}.$$

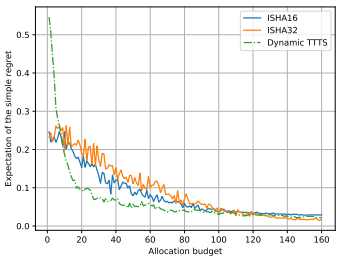
[7] $B(a,b) \triangleq \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$, and Γ is the Gamma function

Now, in our case, where the underlying distribution is the uniform distribution defined over $[0, \mu^\star]$, we obtain the pdf of the order statistic $X_{(k)}$ as follow:

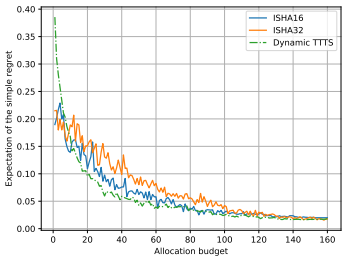
$$\begin{aligned} p_{X_{(k)}}(\theta') &= \frac{n!}{(k-1)!(n-k)!}(\mu^\star)^n(\theta')^{k-1}(\mu^\star - \theta')^{n-k} \\ &= \frac{1}{B(k, n+1-k)} \frac{(\theta')^{k-1}(\mu^\star - \theta')^{n-k}}{(\mu^\star)^{(k-1)+(n-k)+1}}. \end{aligned}$$

We recognize the density of a shifted Beta distribution with k and $n+1-k$ as shape hyper-parameters. In particular, in our case, the pseudo arm at time t is endowed with the distribution $\mathtt{SB}_{\mu^\star}(t - k_t, 1)$.

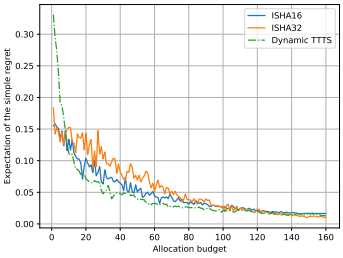
Some illustrations of the fix. Now we show some synthetic results after the previous tricks. Fig. 6.7 shows the expected simple regret of D-TTTS compared to ISHA, again, under $\mathtt{Beta}(0.5, 0.5)$, $\mathtt{SB}_{0.8}(0.5, 0.5)$, $\mathtt{SB}_{0.6}(0.5, 0.5)$, $\mathtt{SB}_{0.4}(0.5, 0.5)$ and $\mathtt{SB}_{0.2}(0.5, 0.5)$ reservoir respectively. We can see that the performance of D-TTTS for shifted cases has been significantly enhanced.



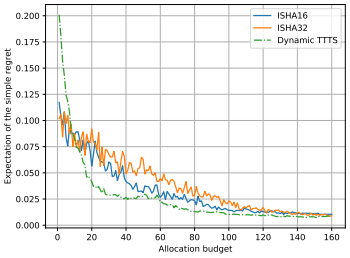
(a) no shift



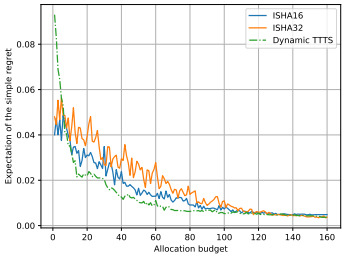
(b) shift by 0.8



(c) shift by 0.6



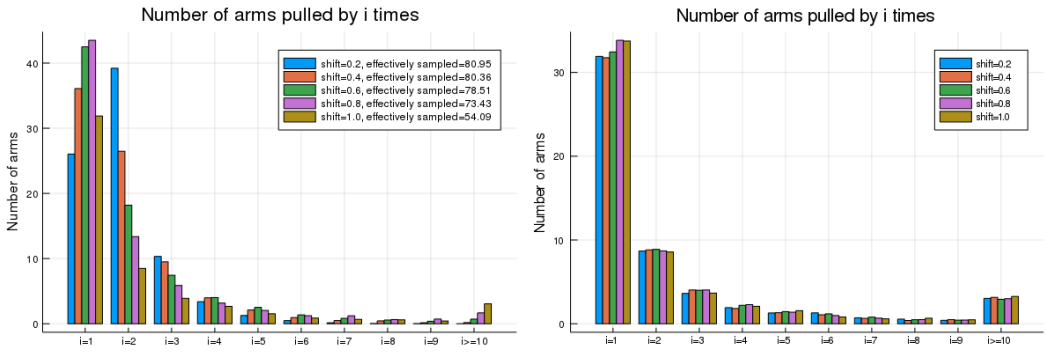
(d) shift by 0.4



(e) shift by 0.2

Figure 6.7: Simple regret of D-TTTS for shifted Beta reservoir after the fix.

We can also compare the number of effectively sampled arms under shifted Beta reservoirs before and after the fix, as shown in Fig. 6.8. Fig. 6.8a is the same figure as Fig. 6.6b, and Fig. 6.8b is the number of effectively sampled arms after the previous fix under a $\mathtt{Beta}(0.5, 0.5)$, $\mathtt{SB}_{0.8}(0.5, 0.5)$, $\mathtt{SB}_{0.6}(0.5, 0.5)$, $\mathtt{SB}_{0.4}(0.5, 0.5)$ and $\mathtt{SB}_{0.2}(0.5, 0.5)$ reservoir respectively. Indeed, we can see that now the exploration effort of D-TTTS under shifted Beta priors is more or less at the same level as that under a normal Beta reservoir.



(a) shift

(b) shift after fix

Figure 6.8: Distribution of effectively sampled arms of D-TTTS before and after the fix.

6.8 Discussion

We presented a way to use Thompson sampling for BAI for infinitely many-armed bandits and explained how to use it for HPO. We introduced the *first fully dynamic algorithm* for this setting and showed through an empirical study that it is a promising approach for HPO.

It would be interesting to establish theoretical guarantees to support the good performance of D-TTTS, with the hope to provide a finite-time upper bound on its probability of error. We also plan to investigate variants of this algorithm for the non-stochastic bandits for which `Hyperband` can be used, which would allow spending more time on the more promising algorithms.

Chapter 7

General Conclusion and Perspectives

" 将来现在将来，与现在有意义，才与将来会有意义。

鲁迅

Contents

7.1 General Discussion 106
7.2 Future Perspectives 106

7.1 General Discussion

In this thesis, we studied the multi-armed bandit problem in an optimization fashion. In particular, we investigated three different settings of best-arm identification (in a broad sense) in the first three chapters.

We first studied BAI in its simplest formulation (Chapter 3), that is bandits with scalar payoffs. We treated the problem with some Bayesian machinery and answered to one open question raised by Russo [2016] on the sample complexity. By showing the (β)-asymptotic optimality of TTTS and providing a computationally faster alternative T3C, we further advocated the use of Bayesian algorithms for BAI.

In the next chapter (Chapter 4), we studied the linear setting with the hope of extending previous Bayesian algorithms while keeping the same sample-complexity guarantee. We argued that previous notion of complexities for linear bandits BAI did not allow us to achieve the asymptotic optimality. Although the result was not satisfying regarding the Bayesian extensions, we managed to propose an alternative LinGame using a saddle-point approach that is asymptotically optimal whilst remaining computationally-friendly.

The third part (Chapter 5) consists of a rather different setting where we aimed to optimize a target function over a continuous-armed space with minimum regularity assumptions. We were interested in designing algorithms that are adaptive to the smoothness. Taking inspiration from POO, we proposed a new general cross-validation scheme GPO. Compared to POO that is only able to encapsulate hierarchical-bandit algorithms with a cumulative-regret guarantee, GPO is able to encapsulate algorithms with simple-regret guarantees.

The first chapters mostly came up with strongly theoretically grounded algorithms in the context of different sequential optimization settings, while in Chapter 6 we also explored a more practical topic, namely hyper-parameter optimization. Existing methods often require to fix an *ad hoc* number of configurations to test, while we managed to propose a dynamic algorithm D-TTTS that do not need such a workaround. It is worth noting that D-TTTS can also simply serve as a heuristic for *infinitely-armed bandits*, but without any theoretical guarantees. Analysis of D-TTTS appears to be difficult due to its dynamic nature, and is left for future work.

7.2 Future Perspectives

Direct follow-ups of the previous research. A prominent follow-up is to further investigate whether both theoretically and practically efficient Bayesian algorithms exist for linear BAI (or even more general structure). As discussed in Chapter 4, our first attempts to extend TTTS to the linear setting leads to a dead end from a theoretical point of view, but still shows some promising experimental performance. I think it is worth putting some efforts on the topic as Bayesian methods could probably avoid resolving complicated optimization problems that is required in most of the current existing state-of-the-art algorithms.

On the other hand, as advocated for example by Locatelli et al. [2016], the fixed-budget and fixed-confidence settings are drastically different. In the future, I would like to put more focus on the fixed-budget setting, and ideally propose finite-time analysis for TTTS.

Further investigation on hyper-parameter tuning. Even if the motivation of this thesis was to design efficient HPO algorithms, the present thesis does not explore the topic very deeply. A main reason is that I find it hard to propose new *task-agnostic* algorithms that

improve over existing methods significantly. Indeed, the most important contribution of `Hyperband` is to introduce a more clever evaluation mode that allocates more resources to more promising configurations. This trick has given way to a considerable improvement over previous methods, in particular on tasks using large DL models. Since `Hyperband`, however, with a lot of recent work trying to advance the state-of-the-art (see Section 6.2), no real breakthrough has been achieved in the field.

In my opinion, the future (or even ongoing) trend of the domain would be to focus on designing more efficient task-specific algorithms. Specific methods could eventually achieve better performance on specific tasks than more general approaches. To some extent, the recent progress on NAS is one running example to support this intuition as RL methods have been successfully applied on NAS (see e.g. Zoph and Le 2017).

A future direction that attracts me a lot is thus to explore the potential of HPO techniques in different industrial applications. For example, one important business need of companies dealing with microchips is the compiler optimization. The methodology will thus be a bit different: we need to design algorithms based on the task itself rather than proposing a general algorithm and testing it on different tasks.

On the other hand, sample-efficient HPO remains an active research field since there is not only neural networks that need automated hyper-parameter tuning. Indeed, in many data science challenges with real-world datasets, classical machine learning methods (in particular ensemble methods like `XGBoost`) often dominate. Bandit-inspired HPO methods can still be plausible candidates for those tasks and I am very interested in discovering further in this direction.

Link to reinforcement learning. Beyond MAB, I would also like to work more on RL in the future, whether it is about the theoretical foundation or finding real applications of RL. RL is a richer and more challenging domain than MAB, yet has strong links with MAB, in particular linear bandits. Indeed, RL extends upon contextual bandits by allowing for long-term consequences. More precisely, for contextual (linear) bandits, the actions only affect the current reward, whereas for RL, they can also affect the future rewards through the evolution of the context.

One possible topic that I would like to investigate – a topic that is also highly related to BAI – is the problem of *best-policy identification* (BPI) in MDPs. Similar to BAI, the goal of BPI is to devise learning algorithms that are able to return the best policy as early as possible. Marjani and Proutiere [2020] adapt `Track-and-Stop` to BPI in discounted MDPs, with the help of a generative model. It is interesting to see if our algorithms for linear bandits BAI can help design sound BPI algorithms in a more general context (without generative model).

In the long term, I will be interested in filling the gap between RL theory and practical usable RL. For example, a large number of recent theoretical RL work has adopted a linear function approximation assumption first studied by Jin et al. [2019] to various RL problem settings. This assumption is interesting from a theoretical point of view, but remains quite unrealistic in practice. It would be interesting to find a more realistic assumption while keeping (or improving) the current guarantees.

Societal impact. Last but not least, there is an emerging attention that has been paid on the societal impact recently. I value also a lot these societal factors or constraints that should be leveraged in the current machine learning research. I am particularly interested in safety which is often required in many real business projects, and also fairness that plays an crucial role in empowering the inclusion.

Some of my recently finished or on-going projects focus on some of those aspects (in particular, safety for linear bandits and a novel bandit setting designed upon fairness considerations) and it is something that I definitely want to take into account in my future work.

Bibliography

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011. 10, 65

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AIStats)*, 2012. 130

Yasin Abbasi-Yadkori, Peter L. Bartlett, Victor Gabillon, Alan Malek, and Michal Valko. Best of both worlds: Stochastic & adversarial best-arm identification. In *Proceedings of the 31st Annual Conference on Learning Theory (CoLT)*, 2018. 22

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, 1995. 72

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Conference on Learning Theory (CoLT)*, pages 1–26, 2012. 94

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 99–107, 2013. 14

S. Damla Ahipasaoglu, Peng Sun, and Michael J. Todd. Linear convergence of a modified Frank-Wolfe algorithm for computing minimum-volume enclosing ellipsoids. *Optimization Methods and Software*, 23(1):5–19, 2008. 52, 53

Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard. The non-stationary stochastic multi-armed bandit problem. *International Journal of Data Science and Analytics*, 3:267–283, 2017. 11

Corwin L. Atwood. Optimal and efficient designs of experiments. *The Annals of Mathematical Statistics*, 40(5):1570–1602, 1969. 52

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Annual Conference on Learning Theory (CoLT)*, 2010. 16, 18, 94

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002. 46

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning Journal*, 47(2-3):235–256, 2002a. viii, 6, 14, 15, 28, 88

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32:48–77, 2002b. 11

Peter Auer, Ronald Ortner, and Csaba Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *Proceedings of the 21st Annual Conference on Learning Theory (CoLT)*, volume 4539, pages 454–468, 2007. 72, 75

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Online stochastic optimization under correlated bandit feedback. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1557–1565, 2014. 72, 73, 79, 81, 82

Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Proceedings of the 29th International Conference on Algorithmic Learning Theory (ALT)*, 2018a. 88

Maryam Aziz, Kevin Jamieson, and Javed Aslam. Pure-exploration for infinite-armed bandits with general arm reservoirs. *arXiv preprint arXiv:1811.06149*, 2018b. 93, 97

MohammadJavad Azizi, Branislav Kveton, and Mohammad Ghavamzadeh. Fixed-budget best-Arm identification in contextual bandits: A static-adaptive algorithms. *arXiv preprint arXiv:2106.04763*, 2021. 22

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967. 82, 125

Akram Baransi, Odalric-ambrym Maillard, and Shie Mannor. Sub-sampling for multi-armed bandits. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2014 (ECML-PKDD)*, 2014. 14

Peter L. Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory (ALT)*, 2019. 73, 85, 88

Dorian Baudry, Emilie Kaufmann, and Odalric-Ambrym Maillard. Sub-sampling for efficient non-parametric bandit exploration. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020. 14

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 2546–2554, 2011. 91

Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *Annals of Statistics*, 25(5):2103–2116, 1997. 23, 88

Dimitri P. Bertsekas. Approximate Dynamic Programming. In *Dynamic Programming and Optimal Control*, volume II, pages 327–552. Athena Scientific, 2011. vii, 4

Lilian Besson. *Algorithmes de bandits multi-joueurs pour les réseaux de l'Internet des objets*. PhD thesis, CentraleSupélec Rennes, 2019. 10

Lilian Besson and Emilie Kaufmann. Multi-player bandits revisited. In *Proceedings of the 29th International Conference on Algorithmic Learning Theory (ALT)*, 2018. 11

Margaux Brégère, Pierre Gaillard, Yannig Goude, and Gilles Stoltz. Target tracking for contextual bandits: Application to demand side management. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. 10

Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. x, 7

Sébastien Bubeck and Nicolò Cesa-Bianchi. *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. now publishers, 2012. 11

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT)*, pages 23–37, 2009. vii, 2, 4, 19, 20, 25, 94

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. X-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627, 2010. x, 7, 23, 88

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011. vii, 4, 28, 78, 79

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013. 13

Adam D. Bull. Adaptive-treed bandits. *Bernoulli*, 21(4):2289–2307, 2015. 72

Apostolos N. Burnetas and Michaël N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996. 28

Olivier Cappé, Aurélien Garivier, Odalric Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013. 14, 28

Alexandra Carpentier and Andrea Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Proceedings of the 29th Annual Conference on Learning Theory (CoLT)*, 2016. 17

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *Proceedings of the 32nd International conference on Machine Learning (ICML)*, pages 1133–1141, 2015. 88, 93, 97

Nicolò Cesa-Bianchi and Gabor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012. 11

Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory*, 61(1): 549–564, 2015. 10

Hock Peng Chan. The multi-armed bandit problem: An efficient nonparametric solution. *Annals of Statistics*, 48(1):346–373, 2020. 14

Marie Agathe Charpagne, Florian Strub, and Tresa M. Pollock. Accurate reconstruction of EBSD datasets by a multimodal data approach using an evolutionary algorithm. *Materials Characterization*, 150:184–198, 2019. vi, 3

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 379–387, 2014. 11, 21

Wei Chen, Yihan Du, and Yuko Kuroki. Combinatorial pure exploration with partial or full-bandit linear feedback. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 22

Herman Chernoff. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3):755–770, 1959. 18, 62

Eric W. Cope. Regret and convergence bounds for a class of continuum-armed bandit problems. *IEEE Transactions on Automatic Control*, 54(6):1243–1253, 2009. 72

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc, 2006. 126

Nando de Freitas, Alex J. Smola, and Masrour Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1743–1750, 2012. 72

Steven de Rooij, Tim Van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15:1281–1316, 2014. 64

Rémy Degenne and Wouter M. Koolen. Pure exploration with multiple correct answers. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019. 21, 62

Rémy Degenne, Wouter Koolen, and Pierre Ménard. Non-asymptotic pure exploration by solving games. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019. 43, 46, 67, 168

Rémy Degenne, Pierre Ménard, Xuedong Shang, and Michal Valko. Gamification of pure exploration for linear bandits. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020a. 8, 47, 62, 65

Rémy Degenne, Han Shao, and Wouter M. Koolen. Structure Adaptive Algorithms for Stochastic Bandits. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020b. 11, 65, 167, 170

Aniket Anand Deshmukh, Srinagesh Sharma, James W. Cutler, Mark Moldwin, and Clayton Scott. Simple regret minimization for contextual bandits. In *2nd Workshop on Exploration in Reinforcement Learning at International Conference on Machine Learning (ICML-ERL)*, 2019. 22

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 88

Omar Darwiche Domingues, Yannis Flet-Berliac, Edouard Leurent, Pierre Ménard, Xuedong Shang, and Michal Valko. rlberry - A reinforcement learning library for research and education, 2021. 8

Dheeru Dua and Karra E. Taniskidou. UCI Machine Learning Repository, 2017. 98

Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de Novo Carcinogenesis. In *Proceedings of the 3rd Machine Learning for Health Care Conference (MLHC)*, 2018. vi, 3

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20:1–21, 2019. 90

Eyal Even-dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for reinforcement learning. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 162–169, 2003. vi, 4, 16, 18, 28

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. 93

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 2962–2970, 2015. 89

Tanner Fiez, Lalit Jain, Kevin Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019. 46, 51, 53, 54, 55, 60

Christodoulos A. Floudas and Panos M. Pardalos. *Optimization in Computational Chemistry and Molecular Biology : Local and Global Approaches*. Springer-Verlag, 2000. vi, 3

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. 52

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 3212–3220, 2012. 18, 42, 46, 55, 94

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Proceedings of the 29th Annual Conference on Learning Theory (CoLT)*, 2016. ix, 6, 18, 22, 24, 28, 32, 33, 35, 41, 42, 46, 48, 49, 57, 64, 65, 168

Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2018. 53

James E. Gentle. *Computational Statistics*. Springer-Verlag New York, 2009. 102

James A. Grant, Alexis Boukouvalas, Ryan-Rhys Griffiths, David S. Leslie, Sattar Vakili, and Enrique Munoz de Cote. Adaptive sensor placement for continuous spaces. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. 10

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pages 667–675, 2015. x, 7, 72, 73, 74, 75, 78, 83, 88, 188, 189

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Blazing the trails before beating the path: Sample-efficient Monte-Carlo planning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 4680–4688, 2016. 72

Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A Survey of the state-of-the-art. *arXiv preprint arXiv:1908.00709*, 2019. 89

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 124

Matthew W. Hoffman, Bobak Shahriari, and Nando de Freitas. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 365–374, 2014. 43, 88

Junya Honda and Akimichi Takemura. Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards. *Journal of Machine Learning Research*, 16:3721–3756, 2015. 14

Deng Huang, Theodore T. Allen, William. I. Notz, and Allen Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, 2006. 91

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization (LION)*, pages 507–523, 2011. 91

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges*. Springer-Verlag, 2019. 89

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AIStats)*, 2016. 72

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *Proceedings of the 27th Annual Conference on Learning Theory (CoLT)*, pages 423–439, 2014. 18

Yassir Jedra and Alexandre Proutière. Optimal best-arm identification in linear bandits. *arXiv preprint arXiv:2006.16073*, 2020. ix, 6, 68, 70

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. *arXiv preprint arXiv:1907.0538*, 2019. 107

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. 91

Kwang-Sung Jun and Robert Nowak. Anytime exploration for multi-armed bandits using confidence information. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 974–982, 2016. 18, 28, 56, 94

Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 511–518, 2010. 22

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 655–662, 2012. 18

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabás Póczos, and Eric Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2018. 90

Zohar Karnin. Verification based solution for structured MAB problems. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 145–153, 2016. 11

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1238–1246, 2013. 18, 46, 89, 93

Julian Katz-samuels, Lalit Jain, Zohar Karnin, and Kevin Jamieson. An empirical process approach to the union bound : Practical algorithms for combinatorial and linear bandits. *arXiv preprint arXiv:2006.11685*, 2020. 70

Emilie Kaufmann and Aurélien Garivier. Learning the distribution with largest mean: two bandit frameworks. *ESAIM: Proceedings and Surveys*, 60:114–131, 2017. 19, 22, 28

Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceedings of the 26th Annual Conference on Learning Theory (CoLT)*, pages 228–251, 2013. 18

Emilie Kaufmann and Wouter Koolen. Mixture martingales revisited with applications to sequential tests and confidence intervals. *arXiv preprint arXiv:1811.11419*, 2018. 35, 37

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory (ALT)*, 2012. 14

Emilie Kaufmann, Wouter M. Koolen, and Aurélien Garivier. Sequential test for the lowest mean: From Thompson to Murphy sampling. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 6332–6342, 2018. 21

Abbas Kazerouni and Lawrence M. Wein. Best arm identification in generalized linear bandits. *arXiv preprint arXiv:1905.08224*, 2019. 22, 46, 56

J. Kiefer and J. Wolfowitz. Optimum designs in regression problems. *The Annals of Mathematical Statistics*, 30(2):271–294, 1959. 51

Aaron Klein, Stefan Falkner, Numair Mansur, and Frank Hutter. RoBO: A flexible and robust Bayesian optimization framework in Python. In *7th Workshop on Bayesian Optimization at Neural Information Processing Systems (NIPS-BayesOpt)*, 2017. 91

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 697–704, 2004. 72

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandit problems in metric spaces. *Symposium on Theory of Computing*, 2008. 72

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *arXiv preprint arXiv:1312.1277*, 2013. 72

Nicolas Knudde, Joachim van der Herten, Tom Dhaene, and Ivo Couckuyt. GPflowOpt: A Bayesian optimization library using TensorFlow. *arXiv preprint arXiv:1711.03845*, 2017. 91

Levente Kocsis and Csaba Szepesvári. Bandit-based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, 2006. 72

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. *Journal of Machine Learning Research*, 40(2015):1–14, 2015. 11

Nathaniel Korda, Emilie Kaufmann, and Rémi Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 1448–1456, 2013. 14

Lars Kotthoff, Chris Thornton, Holger H. Hoos, Frank Hutter, and Kevin Leyton-Brown. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research*, 18(25):1–5, 2017. 89

Andreas Krause and Cheng Soon Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 2447–2455, 2011. 10

Tze-Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. 14

Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2018. 11, 15, 166, 167

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 98

Oleg V. Lepski. Asymptotically minimax adaptive estimation. I: Upper bounds. optimally adaptive estimates. *Theory of Probability & Its Applications*, 36(4):682–697, 1992. 76

Oleg V. Lepski and Vladimir G. Spokoiny. Optimal pointwise adaptive methods in nonparametric estimation. *Annals of Statistics*, 25(6):2512–2546, 1997. 76

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International World Wide Web Conference (WWW)*, pages 661–670. ACM, ACM Press, 2010. ix, 6, 10

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Ameet Talwalkar, and Afshin Rostamizadeh. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. x, 2, 7, 43, 72, 88, 91, 93, 99

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. 90

Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in X-armed bandits. In *Proceedings of the 31st Annual Conference on Learning Theory (CoLT)*, pages 1463–1492, 2018. 76

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2539–2554, 2016. vii, 4, 21, 106

Andrea Locatelli, Alexandra Carpentier, and Samory Kpotufe. Adaptivity to noise parameters in nonparametric active learning. In *Proceedings of the 30th Conference on Learning Theory (CoLT)*, pages 1383–1416, 2017. 76

Ilya Loshchilov and Frank Hutter. CMA-ES for hyperparameter optimization of deep neural networks. In *Workshop Track of the 4th International Conference on Learning Representations*, 2016. 91

Haihao Lu, Robert M. Freund, and Yurii Nesterov. Relatively-smooth convex optimization by first-order methods , and applications. *SIAM Journal of Optimization*, 28(1):333–354, 2018. 54

Aymen Al Marjani and Alexandre Proutiere. Adaptive sampling for best policy identification in Markov decision processes. *arXiv preprint arXiv:2009.13405*, 2020. 107

Joseph Mellor and Jonathan Shapiro. Thompson sampling in switching environments with Bayesian online change point detection. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 442–450, 2013. 11

Pierre Ménard. Gradient ascent for active exploration in bandit problems. *arXiv preprint arXiv:1905.08165*, 2019. 21, 42, 43, 46

Pierre Ménard, Omar Darwiche Domingues, Xuedong Shang, and Michal Valko. UCB momentum Q-learning: Correcting the bias without forgetting. *arXiv preprint arXiv:2103.01312*, 2021. 8

Felix Mohr, Marcel Wever, and Eyke Huellermeier. ML-Plan: Automated machine learning via hierarchical planning. *Machine Learning*, 107(8-10):1495–1515, 2018. 89

Carmen G. Moles, Pedro Mendes, and Julio R. Banga. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, pages 2467–2474, 2003. vi, 3

Rémi Munos. Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 783–791, 2011. 72

Rémi Munos. *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*, volume 7. now publishers, 2014. 72

Randal S. Olson and Jason H. Moore. TPOT: A tree-based pipeline optimization tool for automating machine learning. In *6th Workshop on Automated Machine Learning at International Conference on Machine Learning (ICML-AutoML)*, 2019. 89

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018. 91

Pierre Perrault, Etienne Boursier, Vianney Perchet, and Michal Valko. Statistical efficiency of Thompson sampling for combinatorial semi-bandits. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020. 11

Philippe Preux, Rémi Munos, and Michal Valko. Bandits attack function optimization. *IEEE Congress on Evolutionary Computation*, 2014. 72

Friedrich Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006. ix, 6, 51

Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5381–5391, 2017. 28, 32, 35, 36, 37, 38, 46, 129, 130, 135, 136, 149, 151, 153

Herilalaina Rakotoarison, Marc Schoenauer, and Michèle Sebag. Automated machine learning with Monte-Carlo tree search. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3296–3303, 2019. 89

Clémence Réda, Emilie Kaufmann, and Andrée Delahaye-Duriez. Machine learning applications in drug development. *Computational and Structural Biotechnology Journal*, 18:241–252, 2020. 10

Clémence Réda, Emilie Kaufmann, and Andrée Delahaye-Duriez. Top-m identification for linear bandits. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AIStats)*, 2021. 22

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58(5):527–535, 1952. vi, 4, 14

Daniel Russo. Simple Bayesian algorithms for best arm identification. In *Proceedings of the 29th Annual Conference on Learning Theory (CoLT)*, 2016. ix, 2, 6, 18, 28, 29, 32, 33, 34, 40, 43, 48, 97, 106, 131, 154, 161

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. *A Tutorial on Thompson Sampling*, volume 11. now publishers, 2018. viii, 6

Spyridon Samothrakis, Diego Perez, and Simon Lucas. Training gradient boosting machines using curve-fitting and information-theoretic features for causal direction detection. In *Workshop on Causality at Neural Information Processing Systems (NIPS-Causality)*, 2013. 72

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. 91

Xuedong Shang, Emilie Kaufmann, and Michal Valko. Adaptive black-box optimization got easier: HCT needs only local smoothness. In *14th European Workshop on Reinforcement Learning (EWRL)*, 2018. 73

Xuedong Shang, Emilie Kaufmann, and Michal Valko. General parallel optimization without a metric. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory (ALT)*, 2019a. 8, 73, 88

Xuedong Shang, Emilie Kaufmann, and Michal Valko. A simple dynamic bandit algorithm for hyper-parameter tuning. In *6th Workshop on Automated Machine Learning at International Conference on Machine Learning (ICML-AutoML)*, 2019b. 8, 89

Xuedong Shang, Rianne de Heide, Emilie Kaufmann, Pierre Ménard, and Michal Valko. Fixed-confidence guarantees for Bayesian best-arm identification. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AIStats)*, 2020a. 8, 29, 46

Xuedong Shang, Emilie Kaufmann, and Michal Valko. Simple (dynamic) bandit algorithms for hyper-parameter optimization. 2020b. 8, 89

Xuedong Shang, Han Shao, and Jian Qian. Stochastic bandits with vector losses: Minimizing ℓ^∞-norm of relative losses. *arXiv preprint arXiv:2010.08061*, 2020c. 8

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. v, vii, 3, 4

Aleksandrs Slivkins. Multi-armed bandits on implicit metric spaces. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 1602–1610, 2011. 72

Aleksandrs Slivkins. *Introduction to Multi-Armed Bandits*. now publishers, 2019. 11

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2951–2959, 2012. 91

Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 828–836, 2014. 46, 51, 52, 55, 65, 66, 67

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International conference on Machine Learning (ICML)*, pages 1015–1022, 2010. 88

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. vii, 4

Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2018. 10

Chao Tao, Saul A. Blanco, and Yuan Zhou. Best arm identification in linear bandits with linear dimension dependency. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 7773–7786, 2018. 46, 51, 54, 55, 67

Kazuki Teraoka, Kohei Hatano, and Eiji Takimoto. Efficient sampling method for monte carlo tree search problem. *IEICE Transactions on Information and Systems*, E97-D(3): 392–398, 2014. 21

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285, 1933. vi, viii, 3, 6, 10, 14, 28

Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 847–855, 2013. 89

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 24

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 19–27, 2013. 72

Claire Vernade, Olivier Cappé, and Vianney Perchet. Stochastic bandit models for delayed conversions. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017. 11

Roman Vershynin. *High-Dimensional Probability*. Cambridge University Press, 2018. 125

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. vii, 4

Gary G. Wang and Songqing Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370, 2007. vi, 3

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1729–1736, 2008. 88

Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019. 91

Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AIStats)*, pages 843–851, 2018. 46, 52, 55, 58

Junwen Yang and Vincent Y. F. Tan. Towards minimax optimal best arm identification in linear bandits. *arXiv preprint arXiv:2105.13017*, 2021. 70

Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 1081–1088, 2006. 51

Xiaotian Yu, Han Shao, Michael R Lyu, and Irwin King. Pure exploration of multi-armed bandits with heavy-tailed payoffs. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. 13

Mohammadi Zaki, Avinash Mohan, and Aditya Gopalan. Towards optimal and efficient best arm identification in linear bandits. In *Workshop on Machine Learning at Neural Information Processing Systems (NeurIPS-CausalML)*, 2019. 46, 56, 66

Mohammadi Zaki, Avi Mohan, and Aditya Gopalan. Explicit best arm identification in linear bandits using no-regret learners. *arXiv preprint arXiv:2006.07562*, 2020. 70

Wei Zeng, Meiling Fang, Junming Shao, and Mingsheng Shang. Uncovering the essential links in online commercial networks. *Scientific Reports*, 6, 2016. 10

Yixuan Zhai, Pouya Tehrani, Lin Li, Jiang Zhao, and Qing Zhao. Dynamic pricing under binary demand uncertainty: A multi-armed bandit with correlated arms. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 1597–1601, 2011. 10

Yang Zhang. Progress and challenges in protein structure prediction. *Current Opinion in Structural Biology*, 18(3):342–348, 2008. v, 2

William T. Ziemba and Raymond G. Vickson. *Stochastic Optimization Models in Finance*. Elsevier, 2010. vi, 3

Marc-André Zöller and Marco F. Huber. Survey on automated machine learning. *arXiv preprint arXiv:1904.12054*, 2019. 89

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. 107

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018. 90

Appendix A

Mathematical Tools

A.1 Some Reminders on Probability

We recall some important probability tools in this section. For the record, in the main text of the thesis, $\mathcal{U}([a,b])$ denotes a uniform distribution on the support $[a,b]$, and $\texttt{Beta}(a,b)$ denotes a Beta distribution with shape parameters a and b.

A.1.1 One-dimensional exponential family

In the literature of bandits, we are often interested in the exponential family probability distributions.

Definition A.1. *Give a random variable* X *parameterized by* θ*, we say that it belongs to the one-dimensional exponential family if it can be written as*

$$p_{\mathrm{X}}(x \mid \theta) = b(x) \exp\left[\eta(\theta) \cdot \mathrm{T}(x) + \mathrm{A}(\theta)\right], \tag{A.1}$$

where $\mathrm{T}(\mathrm{X})$ *is the natural sufficient statistic,* η, A *are known functions of* θ *and* b *is a known function of* x*.*

Note that the function b must be non-negative. Besides, the support of $p_{\mathrm{X}}(x \mid \theta)$ does not depend on θ[1]. In the whole thesis, we mainly used the following distributions from the one-dimensional exponential family.

- $\texttt{Ber}(\cdot)$ denotes a Bernoulli distribution.
- $\mathcal{B}(\cdot)$ denotes a Binomial distribution.
- $\mathcal{N}(\cdot,\cdot)$ denotes a normal distribution. Note that only normal distributions with known variance are in the one-dimensional exponential family.

A.1.2 Sub-Gaussian distributions

[1] This property can be used to exclude some parametric distributions from exponential family such as Pareto distributions.

Definition A.2 (sub-Gaussian)**.** *Let* X *be a random variable defined over a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$. X *is sub-Gaussian if there exists a constant* $a \geq 0$ *such that for any* $\lambda \in \mathbb{R}$*, we have*

$$\mathbb{E}\left[\exp(\lambda X)\right] \leq \exp\left\{\frac{a^2\lambda^2}{2}\right\}.$$

And X *is called a-sub-Gaussian.*

A.1.3 Martingales

A martingale is a stochastic process for which the conditional expectation of its value at time $n+1$ is equal to its present value, regardless of all prior values. The formal definition of a *discrete-time* martingale is given below.

Definition A.3. *a discrete-time martingale is a sequence of random variables* $X_1, X_2, \cdots$ *such that*

$$\forall n, \mathbb{E}[X_n] < \infty \text{ and } \mathbb{E}[X_{n+1}|X_1, X_2, \cdots, X_n] = X_n.$$

We can further define the notion of discrete-time submartingales (resp. supermartingales) as a sequence of *integrable* random variables such that for any time n,

$$\mathbb{E}[X_{n+1}|X_1, X_2, \cdots, X_n] \geq (\text{resp. } \leq) X_n.$$

Martingale is the base of many concentration inequalities (see e.g. the next section) that found the bandit theory.

A.2 Concentration Inequalities

Concentration inequalities are a omnipresent tool in MAB as they can serve as a way to bound the deviation of random variables with respect to some value (typically the expected value). In this section, we present two famous inequalities, that have been employed in this thesis.

A.2.1 Hoeffding's inequality

(Chernoff)-Hoeffding's inequality is probably the most known concentration inequality which is first studied by Hoeffding [1963]. We state the Hoeffding's inequality for bounded random variables below.

Theorem A.1. *Let* $X_1, X_2, \cdots, X_n$ *be a sequence of independent random variables bounded in the intervals* $[a_i, b_i]$ *for each* $i \in [n]$ *respectively, then the following inequalities hold*

$$\mathbb{P}\left[\overline{X} - \mathbb{E}\left[\overline{X}\right] \geq t\right] \leq \exp\left\{-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\},$$

$$\mathbb{P}\left[\left|\overline{X} - \mathbb{E}\left[\overline{X}\right]\right| \geq t\right] \leq 2\exp\left\{-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\},$$

where $\overline{(\mathrm{X})}$ *denotes the empirical mean of those random variables,*

$$\overline{\mathrm{X}} \triangleq \frac{1}{n}(\mathrm{X}_1 + \mathrm{X}_2 + \cdots + \mathrm{X}_n)\,.$$

A generalization of the previous inequalities to sub-Gaussian random variables also exists. We do not provide details in this thesis, interested readers can refer to Vershynin [2018].

A.2.2 Azuma's inequality

Another important inequality that has been used in this thesis is Azuma-(Hoeffding)'s inequality [Azuma, 1967]. The vanilla form of Azuma's inequality is stated as follow.

Theorem A.2. *Let* $(\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \cdots)$ *be a sequence of random variables, and we assume that it forms a martingale (or a super-martingale). Suppose that for any* $i \in \mathbb{N}$*,*

$$|\mathrm{X}_i - \mathrm{X}_{i-1}| \leq c_i \; a.s.,$$

with $(c_i)_{i\in\mathbb{N}}$ *a sequence of constants. Then for any positive integer* n *and any real* ε*, the following inequality holds,*

$$\mathbb{P}\left[\mathrm{X}_n - \mathrm{X}_0 \geq \varepsilon\right] \leq \exp\left\{\frac{-\varepsilon^2}{2\sum_{i=1}^n c_i^2}\right\}.$$

Symmetrically, if the sequence form a sub-martingale, then the following inequality holds,

$$\mathbb{P}\left[\mathrm{X}_n - \mathrm{X}_0 \leq -\varepsilon\right] \leq \exp\left\{\frac{-\varepsilon^2}{2\sum_{i=1}^n c_i^2}\right\}.$$

The two parts of Theorem A.2 can be combined together using a union bound to obtain the following two-side bound.

Corollary A.3. *Let* $(\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \cdots)$ *be a martingale such that for any* $i \in \mathbb{N}$*,*

$$|\mathrm{X}_i - \mathrm{X}_{i-1}| \leq c_i \; a.s..$$

Then we have

$$\mathbb{P}\left[\mathrm{X}_n - \mathrm{X}_0 \leq -\varepsilon\right] \leq \exp\left\{\frac{-\varepsilon^2}{2\sum_{i=1}^n c_i^2}\right\}.$$

A.3 Information Theory

In this section, we briefly recall some fundamental notions and results of information theory that are unceasingly used in the technical proofs of this thesis. Readers can refer to Cover and Thomas [2006] for more details.

A.3.1 Entropy

Given a random variable $X : \Omega \to \mathscr{X}$, the *entropy* X measures its uncertainty, and also defines the ultimate data compression. When the random variable is discrete, its entropy H(X) is defined as follow.

Definition A.4 (entropy)**.** *Let* X *be a discrete random variable defined over a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *to an arbitrary space* $\mathscr{X}$*, with probability mass function* p_X*, then its entropy* H(X) *is defined by*

$$\mathrm{H}(X) \triangleq -\sum_{x \in \mathscr{X}} p_X(x) \log p_X(x) .$$

The previous definition can be extended to continuous random variables, namely *differential entropy*.

Definition A.5 (differential entropy)**.** *Let* X *be a continuous random variable defined over a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$*, with probability density function* f*, then its differential entropy* $h(X)$ *is defined by*

$$h(X) \triangleq -\int f(x) \log f(x) \, dx .$$

A.3.2 Kullback-Leibler divergence

important concept is the *relative entropy* or *Kullback-Leibler divergence* (KL divergence), which measures the difference between two probability distributions.

Remark A.4. *KL divergence is not a distance since it does not satisfy the symmetry property in an usual distance definition.*

Before properly defining the KL divergence, let us first give the definition of an important prerequisite notion of *absolutely continuous* probability measures.

Definition A.6 (absolutely continuous probability measures)**.** *Let* $\mathbb{P}$ *and* $\mathbb{Q}$ *be two probability measures defined on a measurable space* $(\Omega, \mathscr{F})$*. If for any event* $F \in \mathscr{F}$ *such that* $\mathbb{Q}(F) = 0$*, we have also* $\mathbb{P}(F) = 0$*, then one says that* $\mathbb{P}$ *is absolutely continuous w.r.t* $\mathbb{Q}$*, and is denoted as* $\mathbb{P} \ll \mathbb{Q}$*.*

The KL divergence is then defined as follow.

Definition A.7 (KL divergence)**.** *For two probability measures $\mathbb{P}$ and $\mathbb{Q}$, if $\mathbb{P} \ll \mathbb{Q}$, then the KL divergence is defined as*

$$\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) \triangleq \int \log\left(\frac{d\mathbb{P}}{d\mathbb{Q}}\right) d\mathbb{P}.$$

A very important property of KL divergence is its non-negativity, which is established by Gibbs' theory.

Theorem A.5 (Gibbs' inequality)**.** *For two probability measures $\mathbb{P}$ and $\mathbb{Q}$, if $\mathbb{P} \ll \mathbb{Q}$, then* $\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) \geq 0$. *The equality holds if and only if $\mathbb{P} = \mathbb{Q}$ $\mathbb{P}$-almost everywhere.*

A.3.3 Two special cases: Gaussian and Bernoulli

For probability distributions in the one-dimensional exponential family, we can simply represent the KL-divergence by their means. For example, if μ_1 and μ_2 are respectively the means of $\mathbb{P}$ and $\mathbb{Q}$, then we can write

$$\mathrm{KL}(\mathbb{P};\mathbb{Q}) = d(\mu_1;\mu_2)\,.$$

For some particular probability distributions, simple closed-form expressions can be deduced. In Example A.1 and Example A.2, we show the expressions for Gaussian and Bernoulli distributions.

Example A.1 (KL-divergence between two Gaussian distributions)**.** *We compute the KL divergence between two normal distributions $\mathbb{P} \sim \mathcal{N}(\mu_1, \sigma_1)$ and $\mathbb{Q} \sim \mathcal{N}(\mu_2, \sigma_2)$.*

$$\mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}.$$

In particular, if $\sigma \triangleq \sigma_1 = \sigma_2$, then the two distributions are parameterized by their means, we can thus denote by

$$d(\mu_1;\mu_2) \triangleq \mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}.$$

Example A.2 (KL divergence between two Bernoulli distributions)**.** *We compute the KL divergence between two Bernoulli distributions $\mathbb{P} \sim \mathtt{Ber}(\mu_1)$ and $\mathbb{Q} \sim \mathtt{Ber}(\mu_2)$, we denote by*

$$kl(\mu_1;\mu_2) \triangleq \mathrm{KL}(\mathbb{P}\|\mathbb{Q}) = \mu_1 \ln\left(\frac{\mu_1}{\mu_2}\right) + (1-\mu_1)\ln\left(\frac{1-\mu_1}{1-\mu_2}\right).$$

Appendix B

Additional Proofs of Chapter 3

B.1 Notation

Table B.1: Table of notation for Chapter 3

Notation	Meaning
$\psi_{n,i} \triangleq \mathbb{P}[I_n = i \mid \mathscr{F}_{n-1}]$	probability of arm i being chosen at time n
$\Psi_{n,i} \triangleq \sum_{l=1}^{n} \psi_{l,i}$	sum of probability of arm i being chosen until time n
$\overline{\psi}_{n,i} \triangleq \frac{\Psi_{n,i}}{n}$	average of probability of arm i being chosen until time n
$T_{n,i}$	number of pulls of arm i before round n
$\mathbf{T}_n$	vector of the number of arm selections
$I_n^\star \triangleq \arg\max_{i \in \mathscr{A}} \mu_{n,i}$	empirical best arm at time n
$\Delta_{\min} \triangleq \min_{i \neq j} \lvert\mu_i - \mu_j\rvert$	minimum mean gap
$\Delta_{\max} \triangleq \max_{i \neq j} \lvert\mu_i - \mu_j\rvert$	maximum mean gap
$J_n^{(1)} \triangleq \arg\max_j a_{n,j}$	index of the largest optimal action probability
$J_n^{(2)} \triangleq \arg\max_{j \neq J_n^{(1)}} a_{n,j}$	index of the second largest optimal action probability

- Note that $J_n^{(1)}$ coincides with the Bayesian recommendation index J_n.
- For any $a, b > 0$, we define a function $C_{a,b}$ s.t. $\forall y$,

$$C_{a,b}(y) \triangleq (a+b-1)\, kl\left(\frac{a-1}{a+b-1}; y\right).$$

- Two real-valued sequences (a_n) and (b_n) are are said to be logarithmically equivalent if

$$\lim_{n\to\infty} \frac{1}{n} \log\left(\frac{a_n}{b_n}\right) = 0,$$

and is represented as $a_n \doteq b_n$.

B.2 Technical Lemmas

The whole fixed-confidence analysis for the two sampling rules are both substantially based on two lemmas: Lemma 5 of Qin et al. [2017] and Lemma 3.4. We prove Lemma 3.4 in this section.

Lemma 3.4. *There exists a random variable* W_2*, such that for all* $i \in \mathscr{A}$*,*

$$\forall n \in \mathbb{N}, |\mathrm{T}_{n,i} - \Psi_{n,i}| \leq \mathrm{W}_2 \sqrt{(n+1)\log(e^2+n)} \ a.s.,$$

and $\mathbb{E}\left[e^{\lambda \mathrm{W}_2}\right] < \infty$ *for any* $\lambda > 0$.

Proof. The proof shares some similarities with that of Lemma 6 of Qin et al. [2017]. For any arm $i \in \mathscr{A}$, define $\forall n \in \mathbb{N}$,

$$\mathrm{D}_n \triangleq \mathrm{T}_{n,i} - \Psi_{n,i},$$

$$d_n \triangleq \mathbb{1}\{\mathrm{I}_n = i\} - \psi_{n,i}.$$

It is clear that $\mathrm{D}_n = \sum_{l=1}^{n-1} d_l$ and $\mathbb{E}[d_n|\mathscr{F}_{n-1}] = 0$. Indeed,

$$\begin{aligned}\mathbb{E}[d_n|\mathscr{F}_{n-1}] &= \mathbb{E}\left[\mathbb{1}\{\mathrm{I}_n = i\} - \psi_{n,i}|\mathscr{F}_{n-1}\right] \\ &= \mathbb{P}[\mathrm{I}_n = i|\mathscr{F}_{n-1}] - \mathbb{E}[\mathbb{P}[\mathrm{I}_n = i|\mathscr{F}_{n-1}]|\mathscr{F}_{n-1}] \\ &= \mathbb{P}[\mathrm{I}_n = i|\mathscr{F}_{n-1}] - \mathbb{P}[\mathrm{I}_n = i|\mathscr{F}_{n-1}] = 0.\end{aligned}$$

The second last equality holds since $\mathbb{P}[\mathrm{I}_n = i|\mathscr{F}_{n-1}]$ is $\mathscr{F}_{n-1}$-measurable. Thus D_n is a martingale, whose increment are 1-sub-Gaussian as $d_n \in [-1,1]$ for all n.

Applying Corollary 8 of Abbasi-Yadkori et al. [2012][1], it holds that, with probability larger than $1-\delta$, for all n,

$$|\mathrm{D}_n| \leq \sqrt{2\,(1+n)\ln\left(\frac{\sqrt{1+n}}{\delta}\right)}$$

which yields the first statement of Lemma 3.4.

We now introduce the random variable

$$\mathrm{W}_2 \triangleq \max_{n\in\mathbb{N}} \max_{i\in\mathscr{A}} \frac{|\mathrm{T}_{n,i} - \Psi_{n,i}|}{\sqrt{(n+1)\ln(e^2+n)}}.$$

Applying the previous inequality with $\delta = e^{-x^2/2}$ yields

$$\begin{aligned}\mathbb{P}\left[\exists n \in \mathbb{N}^\star : |\mathrm{D}_n| > \sqrt{(1+n)\left(\ln(1+n) + x^2\right)}\right] &\leq e^{-x^2/2}, \\ \mathbb{P}\left[\exists n \in \mathbb{N}^\star : |\mathrm{D}_n| > \sqrt{(1+n)\ln\left(e^2+n\right)x^2}\right] &\leq e^{-x^2/2},\end{aligned}$$

where the last inequality uses that for all $a, b \geq 2$, we have $ab \geq a+b$.

Consequently $\forall x \geq 2$, for all $i \in \mathscr{A}$

$$\mathbb{P}\left[\max_{n\in\mathbb{N}} \frac{|\mathrm{T}_{n,i} - \Psi_{n,i}|}{\sqrt{(n+1)\log\left(e^2+n\right)}} \geq x\right] \leq e^{-x^2/2}.$$

[1] We could actually use several deviation inequalities that hold uniformly over time for martingales with sub-Gaussian increments.

Now taking a union bound over $i \in \mathscr{A}$, we have $\forall x \geq 2$,

$$\begin{aligned}
\mathbb{P}\left[\mathrm{W}_2 \geq x\right] &\leq \mathbb{P}\left[\max_{i\in\mathscr{A}} \max_{n\in\mathbb{N}} \frac{|\mathrm{T}_{n,i} - \Psi_{n,i}|}{(n+1)\log\left(\sqrt{e^2+n}\right)} \geq x\right] \\
&\leq \mathbb{P}\left[\bigcup_{i\in\mathscr{A}} \max_{n\in\mathbb{N}} \frac{|\mathrm{T}_{n,i} - \Psi_{n,i}|}{(n+1)\log\left(\sqrt{e^2+n}\right)} \geq x\right] \\
&\leq \sum_{i\in\mathscr{A}} \mathbb{P}\left[\max_{n\in\mathbb{N}} \frac{|\mathrm{T}_{n,i} - \Psi_{n,i}|}{(n+1)\log\left(\sqrt{e^2+n}\right)} \geq x\right] \\
&\leq \mathrm{K} e^{-x^2/2}.
\end{aligned}$$

The previous inequalities imply that $\forall i \in \mathscr{A}$ and $\forall n \in \mathbb{N}$, we have

$$|\mathrm{T}_{n,i} - \Psi_{n,i}| \leq \mathrm{W}_2 \sqrt{(n+1)\log(e^2+n)}$$

almost surely. Now it remains to show that $\forall \lambda > 0, \mathbb{E}\left[e^{\lambda \mathrm{W}_2}\right] < \infty$. Fix some $\lambda > 0$.

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda \mathrm{W}_2}\right] &= \int_{x=1}^{\infty} \mathbb{P}\left[e^{\lambda \mathrm{W}_2} \geq x\right] \mathrm{d}x = \int_{y=0}^{\infty} \mathbb{P}\left[e^{\lambda \mathrm{W}_2} \geq e^{2\lambda y}\right] 2\lambda e^{2\lambda y} \mathrm{d}y \\
&= 2\lambda \int_{y=0}^{2} \mathbb{P}\left[\mathrm{W}_2 \geq 2y\right] e^{2\lambda y} \mathrm{d}y + 2\lambda \int_{y=2}^{\infty} \mathbb{P}\left[\mathrm{W}_2 \geq 2y\right] e^{2\lambda y} \mathrm{d}y \\
&\leq \underbrace{2\lambda \int_{y=0}^{2} \mathbb{P}\left[\mathrm{W}_2 \geq 2y\right] e^{2\lambda y} \mathrm{d}y}_{= e^{4\lambda - 1}} + \underbrace{2\lambda \mathrm{C}_1 \int_{y=2}^{\infty} e^{-y^2/2} e^{2\lambda y} \mathrm{d}y}_{< \infty} < \infty,
\end{aligned}$$

where C_1 is some constant.

□

B.3 Fixed-Confidence Analysis for TTTS

This section is entirely dedicated to TTTS.

B.3.1 Sufficient exploration of all arms

We prove Lemma 3.5 for TTTS. To prove this lemma, we introduce the two following sets of indices for a given $\mathrm{L} > 0$: $\forall n \in \mathbb{N}$ we define

$$\mathrm{U}_n^{\mathrm{L}} \triangleq \{i : \mathrm{T}_{n,i} < \sqrt{\mathrm{L}}\},$$
$$\mathrm{V}_n^{\mathrm{L}} \triangleq \{i : \mathrm{T}_{n,i} < \mathrm{L}^{3/4}\}.$$

It is seemingly non trivial to manipulate directly TTTS's candidate arms, we thus start by connecting TTTS with TTPS (top two probability sampling). TTPS is another sampling rule presented by Russo [2016] for which the two candidate samples are defined as in Appendix B.1, we recall them in the following.

$$\mathrm{J}_n^{(1)} \triangleq \arg\max_j a_{n,j}, \mathrm{J}_n^{(2)} \triangleq \arg\max_{j \neq \mathrm{J}_n^{(1)}} a_{n,j}.$$

Lemma 3.5 is proved via the following sequence of lemmas.

Lemma B.1. *There exists* $\mathrm{L}_1 = \mathit{Poly}(\mathrm{W}_1)$ *s.t. if* $\mathrm{L} > \mathrm{L}_1$, *for all* n, $\mathrm{U}_n^{\mathrm{L}} \neq \emptyset$ *implies* $\mathrm{J}_n^{(1)} \in \mathrm{V}_n^{\mathrm{L}}$ *or* $\mathrm{J}_n^{(2)} \in \mathrm{V}_n^{\mathrm{L}}$.

Proof. If $\mathrm{J}_n^{(1)} \in \mathrm{V}_n^{\mathrm{L}}$, then the proof is finished. Now we assume that $\mathrm{J}_n^{(1)} \in \overline{\mathrm{V}_n^{\mathrm{L}}}$, and we prove that $\mathrm{J}_n^{(2)} \in \mathrm{V}_n^{\mathrm{L}}$.

Step 1. According to Lemma 3.3, there exists $\mathrm{L}_2 = \mathrm{Poly}(\mathrm{W}_1)$ s.t. $\forall \mathrm{L} > \mathrm{L}_2, \forall i \in \overline{\mathrm{U}_n^{\mathrm{L}}}$,

$$\begin{aligned}
|\mu_{n,i} - \mu_i| &\leq \sigma \mathrm{W}_1 \sqrt{\frac{\log(e + \mathrm{T}_{n,i})}{1 + \mathrm{T}_{n,i}}} \\
&\leq \sigma \mathrm{W}_1 \sqrt{\frac{\log(e + \sqrt{\mathrm{L}})}{1 + \sqrt{\mathrm{L}}}} \\
&\leq \sigma \mathrm{W}_1 \frac{\Delta_{\min}}{4\sigma \mathrm{W}_1} = \frac{\Delta_{\min}}{4}.
\end{aligned}$$

The second inequality holds since $x \mapsto \frac{\log(e+x)}{1+x}$ is a decreasing function. The third inequality holds for a large $\mathrm{L} > \mathrm{L}_2$ with $\mathrm{L}_2 = \ldots$.

Step 2. We now assume that $\mathrm{L} > \mathrm{L}_2$, and we define

$$\overline{\mathrm{J}_n^\star} \triangleq \arg\max_{j \in \overline{\mathrm{U}_n^{\mathrm{L}}}} \mu_{n,j} = \arg\max_{j \in \overline{\mathrm{U}_n^{\mathrm{L}}}} \mu_j.$$

The last equality holds since $\forall\, j \in \overline{\mathrm{U}_n^{\mathrm{L}}}$, $|\mu_{n,i} - \mu_i| \leq \Delta_{\min}/4$. We show that there exists $\mathrm{L}_3 = \mathrm{Poly}(\mathrm{W}_1)$ s.t. $\forall \mathrm{L} > \mathrm{L}_3$,

$$\overline{\mathrm{J}_n^\star} = \mathrm{J}_n^{(1)}.$$

We proceed by contradiction, and suppose that $\overline{\mathrm{J}_n^\star} \neq \mathrm{J}_n^{(1)}$, then $\mu_{n,\mathrm{J}_n^{(1)}} < \mu_{n,\overline{\mathrm{J}_n^\star}}$, since $\mathrm{J}_n^{(1)} \in \overline{\mathrm{V}_n^{\mathrm{L}}} \subset \overline{\mathrm{U}_n^{\mathrm{L}}}$. However, we have

$$\begin{aligned}
a_{n,\mathrm{J}_n^{(1)}} &= \Pi_n \left[\theta_{\mathrm{J}_n^{(1)}} > \max_{j \neq \mathrm{J}_n^{(1)}} \theta_j \right] \\
&\leq \Pi_n \left[\theta_{\mathrm{J}_n^{(1)}} > \theta_{\overline{\mathrm{J}_n^\star}} \right] \\
&\leq \frac{1}{2} \exp\left\{ - \frac{(\mu_{n,\mathrm{J}_n^{(1)}} - \mu_{n,\overline{\mathrm{J}_n^\star}})^2}{2\sigma^2 (1/\mathrm{T}_{n,\mathrm{J}_n^{(1)}} + 1/\mathrm{T}_{n,\overline{\mathrm{J}_n^\star}})} \right\}.
\end{aligned}$$

The last inequality uses the Gaussian tail inequality (3.7) of Lemma 3.2. On the other hand,

$$\begin{aligned}
|\mu_{n,\mathrm{J}_n^{(1)}} - \mu_{n,\overline{\mathrm{J}_n^\star}}| &= |\mu_{n,\mathrm{J}_n^{(1)}} - \mu_{\mathrm{J}_n^{(1)}} + \mu_{\mathrm{J}_n^{(1)}} - \mu_{\overline{\mathrm{J}_n^\star}} + \mu_{\overline{\mathrm{J}_n^\star}} - \mu_{n,\overline{\mathrm{J}_n^\star}}| \\
&\geq |\mu_{\mathrm{J}_n^{(1)}} - \mu_{\overline{\mathrm{J}_n^\star}}| - |\mu_{n,\mathrm{J}_n^{(1)}} - \mu_{\mathrm{J}_n^{(1)}} + \mu_{\overline{\mathrm{J}_n^\star}} - \mu_{n,\overline{\mathrm{J}_n^\star}}| \\
&\geq \Delta_{\min} - (\frac{\Delta_{\min}}{4} + \frac{\Delta_{\min}}{4}) \\
&= \frac{\Delta_{\min}}{2},
\end{aligned}$$

and

$$\frac{1}{T_{n,J_n^{(1)}}} + \frac{1}{T_{n,\overline{J_n^\star}}} \leq \frac{2}{\sqrt{L}}.$$

Thus, if we take L_3 s.t.

$$\exp\left\{-\frac{\sqrt{L_3}\Delta_{\min}^2}{16\sigma^2}\right\} \leq \frac{1}{2K},$$

then for any $L > L_3$, we have

$$a_{n,J_n^{(1)}} \leq \frac{1}{2K} < \frac{1}{K},$$

which contradicts the definition of $J_n^{(1)}$. We now assume that $L > L_3$, thus $J_n^{(1)} = \overline{J_n^\star}$.

Step 3. We finally show that for L large enough, $J_n^{(2)} \in V_n^L$. First note that $\forall\, j \in \overline{V_n^L}$, we have

$$a_{n,j} \leq \Pi_n\left[\theta_j \geq \theta_{\overline{J_n^\star}}\right] \leq \exp\left\{-\frac{L^{3/4}\Delta_{\min}^2}{16\sigma^2}\right\}. \tag{B.1}$$

This last inequality can be proved using the same argument as Step 2. Now we define another index $J_n^\star \triangleq \arg\max_{j\in U_n^L} \mu_{n,j}$ and the quantity $c_n \triangleq \max(\mu_{n,J_n^\star}, \mu_{n,\overline{J_n^\star}})$. We can lower bound $a_{n,J_n^\star}$ as follows:

$$\begin{aligned}
a_{n,J_n^\star} &\geq \Pi_n\left[\theta_{J_n^\star} \geq c_n\right] \prod_{j\neq J_n^\star} \Pi_n\left[\theta_j \leq c_n\right] \\
&= \Pi_n\left[\theta_{J_n^\star} \geq c_n\right] \prod_{j\neq J_n^\star; j\in U_n^L} \Pi_n\left[\theta_j \leq c_n\right] \prod_{j\in\overline{U_n^L}} \Pi_n\left[\theta_j \leq c_n\right] \\
&\geq \Pi_n\left[\theta_{J_n^\star} \geq c_n\right] \frac{1}{2^{K-1}}.
\end{aligned}$$

Now there are two cases:

- If $\mu_{n,J_n^\star} > \mu_{n,\overline{J_n^\star}}$, then we have

$$\Pi_n\left[\theta_{J_n^\star} \geq c_n\right] = \Pi_n\left[\theta_{J_n^\star} \geq \mu_{n,J_n^\star}\right] \geq \frac{1}{2}.$$

- If $\mu_{n,J_n^\star} < \mu_{n,\overline{J_n^\star}}$, then we can apply the Gaussian tail bound (3.8) of Lemma 3.2, and we obtain

$$\begin{aligned}
\Pi_n\left[\theta_{J_n^\star} \geq c_n\right] = \Pi_n\left[\theta_{J_n^\star} \geq \mu_{n,\overline{J_n^\star}}\right] &= \Pi_n\left[\theta_{J_n^\star} \geq \mu_{n,J_n^\star} + (\mu_{n,\overline{J_n^\star}} - \mu_{n,J_n^\star})\right] \\
&\geq \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(1 - \frac{\sqrt{T_{n,J_n^\star}}}{\sigma}(\mu_{n,J_n^\star} - \mu_{n,\overline{J_n^\star}})\right)^2\right\} \\
&= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(1 + \frac{\sqrt{T_{n,J_n^\star}}}{\sigma}(\mu_{n,\overline{J_n^\star}} - \mu_{n,J_n^\star})\right)^2\right\}.
\end{aligned}$$

On the other hand, by Lemma 3.3, we know that

$$\begin{aligned}
|\mu_{n,J_n^\star} - \mu_{n,\overline{J_n^\star}}| &= |\mu_{n,J_n^\star} - \mu_{J_n^\star} + \mu_{J_n^\star} - \mu_{\overline{J_n^\star}} + \mu_{\overline{J_n^\star}} - \mu_{n,\overline{J_n^\star}}| \\
&\leq |\mu_{J_n^\star} - \mu_{\overline{J_n^\star}}| + \sigma W_1 \sqrt{\frac{\log(e + T_{n,J_n^\star})}{1 + T_{n,J_n^\star}}} + \sigma W_1 \sqrt{\frac{\log(e + T_{n,\overline{J_n^\star}})}{1 + T_{n,\overline{J_n^\star}}}} \\
&\leq |\mu_{J_n^\star} - \mu_{\overline{J_n^\star}}| + 2\sigma W_1 \sqrt{\frac{\log(e + T_{n,J_n^\star})}{1 + T_{n,J_n^\star}}} \\
&\leq \Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + T_{n,J_n^\star})}{1 + T_{n,J_n^\star}}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pi_n \left[\theta_{J_n^\star} \geq c_n\right] &\geq \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(1 + \frac{\sqrt{T_{n,J_n^\star}}}{\sigma}\left(\Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + T_{n,J_n^\star})}{1 + T_{n,J_n^\star}}}\right)\right)^2 \right\} \\
&\geq \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(1 + \frac{\sqrt{\sqrt{L}}}{\sigma}\left(\Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + \sqrt{L})}{1 + \sqrt{L}}}\right)\right)^2 \right\} \\
&\geq \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(1 + \frac{L^{1/4}\Delta_{\max}}{\sigma} + 2W_1 \sqrt{\log(e + \sqrt{L})}\right)^2 \right\}.
\end{aligned}$$

Now we have

$$a_{n,J_n^\star} \geq \max\left(\left(\frac{1}{2}\right)^K, \left(\frac{1}{2}\right)^{K-1} \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\left(1 + \frac{L^{1/4}\Delta_{\max}}{\sigma} + 2W_1 \sqrt{\log(e + \sqrt{L})}\right)^2 \right\}\right),$$

and we have $\forall j \in \overline{V_n^L}$, $a_{n,j} \leq \exp\{-L^{3/4}\Delta_{\min}^2/(16\sigma^2)\}$, thus there exists $L_4 = \text{Poly}(W_1)$ s.t. $\forall L > L_4$, $\forall j \in \overline{V_n^L}$,

$$a_{n,j} \leq \frac{a_{n,J_n^\star}}{2},$$

and by consequence, $J_n^{(2)} \in V_n^L$.

Finally, taking $L_1 = \max(L_2, L_3, L_4)$, we have $\forall L > L_1$, either $J_n^{(1)} \in V_n^L$ or $J_n^{(2)} \in V_n^L$. □

Next we show that there exists at least one arm in V_n^L for whom the probability of being pulled is large enough. More precisely, we prove the following lemma.

Lemma B.2. *There exists* $L_1 = \text{Poly}(W_1)$ *s.t. for* $L > L_1$ *and for all* n *s.t.* $U_n^L \neq \emptyset$, *then there exists* $J_n \in V_n^L$ *s.t.*

$$\psi_{n,J_n} \geq \frac{\min(\beta, 1-\beta)}{K^2} \triangleq \psi_{\min}.$$

Proof. Using Lemma B.1, we know that $J_n^{(1)}$ or $J_n^{(2)} \in V_n^L$. On the other hand, we know that

$$\forall i \in \mathscr{A}, \psi_{n,i} = a_{n,i}\left(\beta + (1-\beta)\sum_{j \neq i} \frac{a_{n,j}}{1 - a_{n,j}}\right).$$

Therefore we have

$$\Psi_{n,J_n^{(1)}} \geq \beta a_{n,J_n^{(1)}} \geq \frac{\beta}{K},$$

since $\sum_{i\in\mathscr{A}} a_{n,i} = 1$, and

$$\begin{aligned}\Psi_{n,J_n^{(2)}} &\geq (1-\beta) a_{n,J_n^{(2)}} \frac{a_{n,J_n^{(1)}}}{1-a_{n,J_n^{(1)}}} \\ &= (1-\beta) a_{n,J_n^{(1)}} \frac{a_{n,J_n^{(2)}}}{1-a_{n,J_n^{(1)}}} \\ &\geq \frac{1-\beta}{K^2},\end{aligned}$$

since $a_{n,J_n^{(1)}} \geq 1/K$ and $\sum_{i\neq J_n^{(1)}} a_{n,i}/(1-a_{n,J_n^{(1)}}) = 1$, thus $a_{n,J_n^{(2)}}/(1-a_{n,J_n^{(1)}}) \geq 1/K$. □

The rest of this subsection is quite similar to that of Qin et al. [2017]. Indeed, with the above lemma, we can show that the set of poorly explored arms U_n^L is empty when n is large enough.

Lemma B.3. *Under TTTS, there exists* $L_0 = Poly(W_1, W_2)$ *s.t.* $\forall L > L_0$, $U_{\lfloor KL\rfloor}^L = \emptyset$.

Proof. We proceed by contradiction, and we assume that $U_{\lfloor KL\rfloor}^L$ is not empty. Then for any $1 \leq \ell \leq \lfloor KL\rfloor$, U_ℓ^L and V_ℓ^L are non empty as well.

There exists a deterministic L_5 s.t. $\forall L > L_5$,

$$\lfloor L\rfloor \geq KL^{3/4}.$$

Using the pigeonhole principle, there exists some $i \in \mathscr{A}$ s.t. $T_{\lfloor L\rfloor,i} \geq L^{3/4}$. Thus, we have $|V_{\lfloor L\rfloor}^L| \leq K-1$.

Next, we prove $|V_{\lfloor 2L\rfloor}^L| \leq K-2$. Otherwise, since U_ℓ^L is non-empty for any $\lfloor L\rfloor + 1 \leq \ell \leq \lfloor 2L\rfloor$, thus by Lemma B.2, there exists $J_\ell \in V_\ell^L$ s.t. $\psi_{\ell,J_\ell} \geq \psi_{\min}$. Therefore,

$$\sum_{i\in V_\ell^L} \psi_{\ell,i} \geq \psi_{\min},$$

and

$$\sum_{i\in V_{\lfloor L\rfloor}^L} \psi_{\ell,i} \geq \psi_{\min}$$

since $V_\ell^L \subset V_{\lfloor L\rfloor}^L$. Hence, we have

$$\sum_{i\in V_{\lfloor L\rfloor}^L} (\Psi_{\lfloor 2L\rfloor,i} - \Psi_{\lfloor L\rfloor,i}) = \sum_{\ell=\lfloor L\rfloor+1}^{\lfloor 2L\rfloor} \sum_{i\in V_{\lfloor L\rfloor}^L} \psi_{\ell,i} \geq \psi_{\min}\lfloor L\rfloor.$$

Then, using Lemma 3.4, there exists $L_6 = \mathrm{Poly}(W_2)$ s.t. $\forall L > L_6$, we have

$$\begin{aligned}\sum_{i\in V_{\lfloor L\rfloor}^L} (T_{\lfloor 2L\rfloor,i} - T_{\lfloor L\rfloor,i}) &\geq \sum_{i\in V_{\lfloor L\rfloor}^L} (\Psi_{\lfloor 2L\rfloor,i} - \Psi_{\lfloor L\rfloor,i} - 2W_2\sqrt{\lfloor 2L\rfloor \log(e^2 + \lfloor 2L\rfloor)}) \\ &\geq \sum_{i\in V_{\lfloor L\rfloor}^L} (\Psi_{\lfloor 2L\rfloor,i} - \Psi_{\lfloor L\rfloor,i}) - 2KW_2\sqrt{\lfloor 2L\rfloor \log(e^2 + \lfloor 2L\rfloor)} \\ &\geq \psi_{\min}\lfloor L\rfloor - 2KW_2C_2\lfloor L\rfloor^{3/4} \\ &\geq KL^{3/4},\end{aligned}$$

where C_2 is some absolute constant. Thus, we have one arm in $V^L_{\lfloor L \rfloor}$ that is pulled at least $L^{3/4}$ times between $\lfloor L \rfloor + 1$ and $\lfloor 2L \rfloor$, thus $|V^L_{\lfloor 2L \rfloor}| \leq K - 2$.

By induction, for any $1 \leq k \leq K$, we have $|V^L_{\lfloor kL \rfloor}| \leq K - k$, and finally if we take $L_0 = \max(L_1, L_5, L_6)$, then $\forall L > L_0$, $U^L_{\lfloor KL \rfloor} = \emptyset$. □

We can finally conclude the proof of Lemma 3.5 for TTTS.

Proof of Lemma 3.5 Let $N_1 = KL_0$ where $L_0 = \text{Poly}(W_1, W_2)$ is chosen according to Lemma B.3. For all $n > N_1$, we let $L = n/K$, then by Lemma B.3, we have $U^L_{\lfloor KL \rfloor} = U^{n/K}_n$ is empty, which concludes the proof.

■

B.3.2 Concentration of the empirical means

We prove Lemma 3.6 for TTTS. As a corollary of the previous section, we can show the concentration of $\mu_{n,i}$ to μ_i for TTTS[2].

By Lemma 3.3, we know that $\forall i \in \mathscr{A}$ and $n \in \mathbb{N}$,

$$|\mu_{n,i} - \mu_i| \leq \sigma W_1 \sqrt{\frac{\log(e + T_{n,i})}{T_{n,i} + 1}}.$$

According to the previous section, there exists $N_1 = \text{Poly}(W_1, W_2)$ s.t. $\forall n \geq N_1$ and $\forall i \in \mathscr{A}$, $T_{n,i} \geq \sqrt{n/K}$. Therefore,

$$|\mu_{n,i} - \mu_i| \leq \sqrt{\frac{\log(e + \sqrt{n/K})}{\sqrt{n/K} + 1}},$$

since $x \mapsto \log(e + x)/(x + 1)$ is a decreasing function. There exists $N'_2 = \text{Poly}(\varepsilon, W_1)$ s.t. $\forall n \geq N'_2$,

$$\sqrt{\frac{\log(e + \sqrt{n/K})}{\sqrt{n/K} + 1}} \leq \sqrt{\frac{2(n/K)^{1/4}}{\sqrt{n/K} + 1}} \leq \frac{\varepsilon}{\sigma W_1}.$$

Therefore, $\forall n \geq N_2 \triangleq \max\{N_1, N'_2\}$, we have

$$|\mu_{n,i} - \mu_i| \leq \sigma W_1 \frac{\varepsilon}{\sigma W_1}.$$

B.3.3 Measurement effort concentration of the optimal arm

In this section we show that the empirical arm draws proportion of the true best arm for TTTS concentrates to β when the total number of arm draws is sufficiently large. We prove Lemma 3.7 for TTTS.

The proof is established upon the following lemmas. First, we prove that the empirical best arm coincides with the true best arm when the total number of arm draws goes sufficiently large.

[2] this proof is the same as Proposition 3 of Qin et al. [2017]

Lemma B.4. *Under TTTS, there exists* $\mathrm{M}_1 = \mathit{Poly}(\mathrm{W}_1, \mathrm{W}_2)$ *s.t.* $\forall n > \mathrm{M}_1$, *we have* $\mathrm{I}_n^\star = \mathrm{I}^\star = \mathrm{J}_n^{(1)}$ *and* $\forall i \neq \mathrm{I}^\star$,

$$a_{n,i} \leq \exp\left\{-\frac{\Delta_{min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\}.$$

Proof. Using Lemma 3.6 with $\varepsilon = \Delta_{\min}/4$, there exists $\mathrm{N}_1' = \mathrm{Poly}(4/\Delta_{\min}, \mathrm{W}_1, \mathrm{W}_2)$ s.t. $\forall n > \mathrm{N}_1'$,

$$\forall i \in \mathscr{A}, |\mu_{n,i} - \mu_i| \leq \frac{\Delta_{\min}}{4},$$

which implies that starting from a known moment, $\mu_{n,\mathrm{I}^\star} > \mu_{n,i}$ for all $i \neq \mathrm{I}^\star$, hence $\mathrm{I}_n^\star = \mathrm{I}^\star$. Thus, $\forall i \neq \mathrm{I}^\star$,

$$\begin{aligned} a_{n,i} &= \Pi_n\left[\theta_i > \max_{j \neq i} \theta_j\right] \\ &\leq \Pi_n\left[\theta_i > \theta_{\mathrm{I}^\star}\right] \\ &\leq \frac{1}{2}\exp\left\{-\frac{(\mu_{n,i} - \mu_{n,\mathrm{I}^\star})^2}{2\sigma^2(1/\mathrm{T}_{n,i} + 1/\mathrm{T}_{n,\mathrm{I}^\star})}\right\}. \end{aligned}$$

The last inequality uses the Gaussian tail inequality of (3.7) Lemma 3.2. Furthermore,

$$\begin{aligned} (\mu_{n,i} - \mu_{n,\mathrm{I}^\star})^2 &= (|\mu_{n,i} - \mu_{n,\mathrm{I}^\star}|)^2 \\ &= (|\mu_{n,i} - \mu_i + \mu_i - \mu_{\mathrm{I}^\star} + \mu_{\mathrm{I}^\star} - \mu_{n,\mathrm{I}^\star}|)^2 \\ &\geq (|\mu_i - \mu_{\mathrm{I}^\star}| - |\mu_{n,i} - \mu_i + \mu_{\mathrm{I}^\star} - \mu_{n,\mathrm{I}^\star}|)^2 \\ &\geq \left(\Delta_{\min} - \left(\frac{\Delta_{\min}}{4} + \frac{\Delta_{\min}}{4}\right)\right)^2 = \frac{\Delta_{\min}^2}{4}, \end{aligned}$$

and according to Lemma 3.5, we know that there exists $\mathrm{M}_2 = \mathrm{Poly}(\mathrm{W}_1, \mathrm{W}_2)$ s.t. $\forall n > \mathrm{M}_2$,

$$\frac{1}{\mathrm{T}_{n,i}} + \frac{1}{\mathrm{T}_{n,\mathrm{I}^\star}} \leq \frac{2}{\sqrt{n/\mathrm{K}}}.$$

Thus, $\forall n > \max\{\mathrm{N}_1', \mathrm{M}_2\}$, we have

$$\forall i \neq \mathrm{I}^\star, a_{n,i} \leq \exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\}.$$

Then, we have

$$a_{n,\mathrm{I}^\star} = 1 - \sum_{i \neq \mathrm{I}^\star} a_{n,i} \geq 1 - (\mathrm{K}-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\}.$$

There exists M_2' s.t. $\forall n > \mathrm{M}_2'$, $a_{n,\mathrm{I}^\star} > 1/2$, and by consequence $\mathrm{I}^\star = \mathrm{J}_n^{(1)}$. Finally taking $\mathrm{M}_1 \triangleq \max\{\mathrm{N}_1', \mathrm{M}_2, \mathrm{M}_2'\}$ concludes the proof. □

Before we prove Lemma 3.7, we first show that $\Psi_{n,\mathrm{I}^\star}/n$ concentrates to β.

Lemma B.5. *Under TTTS, fix a constant* $\varepsilon > 0$, *there exists* $\mathrm{M}_3 = \mathit{Poly}(\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$ *s.t.* $\forall n > \mathrm{M}_3$, *we have*

$$\left|\frac{\Psi_{n,\mathrm{I}^\star}}{n} - \beta\right| \leq \varepsilon.$$

Proof. By Lemma B.4, we know that there exists $M_1' = \mathrm{Poly}(W_1, W_2)$ s.t. $\forall n > M_1'$, we have $I_n^\star = I^\star = J_n^{(1)}$ and $\forall i \neq I^\star$,

$$a_{n,i} \leq \exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{K}}\right\}.$$

Note also that $\forall n \in \mathbb{N}$, we have

$$\Psi_{n,I^\star} = a_{n,I^\star}\left(\beta + (1-\beta)\sum_{j\neq I^\star}\frac{a_{n,j}}{1-a_{n,j}}\right).$$

We proceed the proof with the following two steps.

Step 1. We first lower bound $\Psi_{n,I^\star}$ for a given ε. Take $M_4 > M_1'$ that we decide later, we have $\forall n > M_4$,

$$\begin{aligned}
\frac{\Psi_{n,I^\star}}{n} &= \frac{1}{n}\sum_{l=1}^{n}\psi_{l,I^\star} = \frac{1}{n}\sum_{l=I^\star}^{M_4}\psi_{l,I^\star} + \frac{1}{n}\sum_{l=M_4+1}^{n}\psi_{l,I^\star} \\
&\geq \frac{1}{n}\sum_{l=M_4+1}^{n}\psi_{l,I^\star} \geq \frac{1}{n}\sum_{l=M_4+1}^{n}a_{l,I^\star}\beta \\
&= \frac{\beta}{n}\sum_{l=M_4+1}^{n}\left(1-\sum_{j\neq I^\star}a_{l,j}\right) \\
&\geq \frac{\beta}{n}\sum_{l=M_4+1}^{n}\left(1-(K-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{K}}\right\}\right) \\
&= \beta - \frac{M_4}{n}\beta - \frac{\beta}{n}\sum_{l=M_4+1}^{n}(K-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{K}}\right\} \\
&\geq \beta - \frac{M_4}{n}\beta - \frac{(n-M_4)}{n}\beta(K-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{M_4}{K}}\right\} \\
&\geq \beta - \frac{M_4}{n}\beta - \beta(K-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{M_4}{K}}\right\}.
\end{aligned}$$

For a given constant $\varepsilon > 0$, there exists M_5 s.t. $\forall n > M_5$,

$$\beta(K-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{K}}\right\} < \frac{\varepsilon}{2}.$$

Furthermore, there exists $M_6 = \mathrm{Poly}(\varepsilon/2, M_5)$ s.t. $\forall n > M_6$,

$$\frac{M_5}{n}\beta < \frac{\varepsilon}{2}.$$

Therefore, if we take $M_4 \triangleq \max\{M_1', M_5, M_6\}$, we have $\forall n > M_4$,

$$\frac{\Psi_{n,I^\star}}{n} \geq \beta - \varepsilon.$$

Step 2. On the other hand, we can also upper bound $\Psi_{n,\mathrm{I}^\star}$. We have $\forall n > \mathrm{M}_3$,

$$\begin{aligned}
\frac{\Psi_{n,\mathrm{I}^\star}}{n} &= \frac{1}{n}\sum_{l=1}^{n}\psi_{l,\mathrm{I}^\star} \\
&= \frac{1}{n}\sum_{l=1}^{n} a_{l,\mathrm{I}^\star}\left(\beta + (1-\beta)\sum_{j\neq \mathrm{I}^\star}\frac{a_{l,j}}{1-a_{l,j}}\right) \\
&\leq \frac{1}{n}\sum_{l=1}^{n} a_{l,\mathrm{I}^\star}\beta + \frac{1}{n}\sum_{l=1}^{n} a_{l,\mathrm{I}^\star}(1-\beta)\sum_{j\neq \mathrm{I}^\star}\frac{a_{l,j}}{1-a_{l,j}} \\
&\leq \beta + \frac{1}{n}\sum_{l=1}^{n}(1-\beta)\sum_{j\neq \mathrm{I}^\star}\frac{a_{l,j}}{1-a_{l,j}} \\
&\leq \beta + \frac{1}{n}\sum_{l=1}^{n}(1-\beta)\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}.
\end{aligned}$$

Since, for a given $\varepsilon > 0$, there exists M_8 s.t. $\forall n > \mathrm{M}_8$,

$$\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\} < \frac{1}{2},$$

and there exists M_9 s.t. $\forall n > \mathrm{M}_9$,

$$(1-\beta)(\mathrm{K}-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{\mathrm{K}}}\right\} < \frac{\varepsilon}{4}.$$

Thus, $\forall n > \mathrm{M}_{10} \triangleq \max\{\mathrm{M}_8, \mathrm{M}_9\}$,

$$\begin{aligned}
\frac{\Psi_{n,\mathrm{I}^\star}}{n} &\leq \beta + \frac{1-\beta}{n}\left(\sum_{l=1}^{\mathrm{M}_{10}}\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}} + \sum_{l=\mathrm{M}_{10}+1}^{n}\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}\right) \\
&\leq \beta + \frac{1-\beta}{n}\sum_{l=1}^{\mathrm{M}_{10}}\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}} + 2(1-\beta)(\mathrm{K}-1)\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{\mathrm{M}_{10}}{\mathrm{K}}}\right\} \\
&\leq \beta + \frac{1-\beta}{n}\sum_{l=1}^{\mathrm{M}_{10}}\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}} + \frac{\varepsilon}{2}.
\end{aligned}$$

There exists $\mathrm{M}_{11} = \mathrm{Poly}(\varepsilon/2, \mathrm{M}_{10})$ s.t. $\forall n > \mathrm{M}_{11}$,

$$\frac{1-\beta}{n}\sum_{l=1}^{\mathrm{M}_{10}}\sum_{j\neq \mathrm{I}^\star}\frac{\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}}{1-\exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{l}{\mathrm{K}}}\right\}} < \frac{\varepsilon}{2}.$$

Therefore, $\forall n > \mathrm{M}_7 \triangleq \max\{\mathrm{M}_3, \mathrm{M}_{11}\}$, we have

$$\frac{\Psi_{n,\mathrm{I}^\star}}{n} \leq \beta + \varepsilon.$$

Conclusion. Finally, combining the two steps and define $M_3 \triangleq \max\{M_4, M_7\}$, we have $\forall n > M_3$,

$$\left| \frac{\Psi_{n,I^\star}}{n} - \beta \right| \leq \varepsilon.$$

□

With the help of the previous lemma and Lemma 3.4, we can finally prove Lemma 3.7.

Proof of Lemma 3.7 Fix an $\varepsilon > 0$. Using Lemma 3.4, we have $\forall n \in \mathbb{N}$,

$$\left| \frac{T_{n,I^\star}}{n} - \frac{\Psi_{n,I^\star}}{n} \right| \leq \frac{W_2 \sqrt{(n+1)\log(e^2+n)}}{n}.$$

Thus there exists M_{12} s.t. $\forall n > M_{12}$,

$$\left| \frac{T_{n,I^\star}}{n} - \frac{\Psi_{n,I^\star}}{n} \right| \leq \frac{\varepsilon}{2}.$$

And using Lemma B.5, there exists $M_3' = \mathrm{Poly}(\varepsilon/2, W_1, W_2)$ s.t. $\forall n > M_3'$,

$$\left| \frac{\Psi_{n,I^\star}}{n} - \beta \right| \leq \frac{\varepsilon}{2}.$$

Again, according to Lemma B.2, there exists M_3' s.t. $\forall n > M_3'$,

$$\frac{\Psi_{n,I^\star}}{n} \leq \beta + \frac{\varepsilon}{2}.$$

Thus, if we take $N_3 \triangleq \max\{M_3', M_{12}\}$, then $\forall n > N_3$, we have

$$\left| \frac{T_{n,I^\star}}{n} - \beta \right| \leq \varepsilon.$$

■

B.3.4 Measurement effort concentration of other arms

In this section, we show that, for TTTS, the empirical measurement effort concentration also holds for other arms than the true best arm. We prove Lemma 3.8 for TTTS.

We first show that if some arm is overly sampled at time n, then its probability of being picked is reduced exponentially.

Lemma B.6. *Under TTTS, for every $\xi \in (0,1)$, there exists $S_1 = \mathrm{Poly}(1/\xi, W_1, W_2)$ such that for all $n > S_1$, for all $i \neq I^\star$,*

$$\frac{\Psi_{n,i}}{n} \geq \omega_i^\beta + \xi \;\Rightarrow\; \psi_{n,i} \leq \exp\{-\varepsilon_0(\xi) n\},$$

where ε_0 is defined in (B.2) below.

Proof. First, by Lemma B.4, there exists $M_1'' = \text{Poly}(W_1, W_2)$ s.t. $\forall n > M_1''$,

$$I^\star = I_n^\star = J_n^{(1)}.$$

Then, following the similar argument as in Lemma B.17, one can show that for all $i \neq I^\star$ and for all $n > M_1''$,

$$\begin{aligned}
\psi_{n,i} &= a_{n,i}\left(\beta + (1-\beta)\sum_{j\neq i}\frac{a_{n,j}}{1-a_{n,j}}\right) \\
&\leq a_{n,i}\beta + a_{n,i}(1-\beta)\frac{\sum_{j\neq i} a_{n,j}}{1-a_{n,J_n^{(1)}}} \\
&= a_{n,i}\beta + a_{n,i}(1-\beta)\frac{\sum_{j\neq i} a_{n,j}}{1-a_{n,I^\star}} \\
&\leq a_{n,i}\beta + a_{n,i}(1-\beta)\frac{1}{1-a_{n,I^\star}} \\
&\leq \frac{a_{n,i}}{1-a_{n,I^\star}} \\
&\leq \frac{\Pi_n\left[\theta_i \geq \theta_{I^\star}\right]}{\Pi_n\left[\cup_{j\neq I^\star}\theta_j \geq \theta_{I^\star}\right]} \\
&\leq \frac{\Pi_n\left[\theta_i \geq \theta_{I^\star}\right]}{\max_{j\neq I^\star}\Pi_n\left[\theta_j \geq \theta_{I^\star}\right]}.
\end{aligned}$$

Using the upper and lower Gaussian tail bounds from Lemma 3.2, we have

$$\begin{aligned}
\psi_{n,i} &\leq \frac{\exp\left\{-\frac{(\mu_{n,I^\star}-\mu_{n,i})^2}{2\sigma^2\left(1/T_{n,I^\star}+1/T_{n,i}\right)}\right\}}{\exp\left\{-\min_{j\neq I^\star}\frac{1}{2}\left(\frac{(\mu_{n,I^\star}-\mu_{n,j})}{\sigma\sqrt{\left(1/T_{n,I^\star}+1/T_{n,j}\right)}}-1\right)^2\right\}} \\
&= \frac{\exp\left\{-n\frac{(\mu_{n,I^\star}-\mu_{n,i})^2}{2\sigma^2\left(n/T_{n,I^\star}+n/T_{n,i}\right)}\right\}}{\exp\left\{-n\left(\min_{j\neq I^\star}\frac{(\mu_{n,I^\star}-\mu_{n,j})}{\sqrt{2\sigma^2\left(n/T_{n,I^\star}+n/T_{n,j}\right)}}-\frac{1}{\sqrt{2n}}\right)^2\right\}},
\end{aligned}$$

where we assume that $n > S_2 = \text{Poly}(W_1, W_2)$ for which

$$\frac{(\mu_{n,I^\star}-\mu_{n,i})^2}{\sigma^2\left(1/T_{n,I^\star}+1/T_{n,i}\right)} \geq 1$$

according to Lemma 3.5. From there we take a supremum over the possible allocations to

lower bound the denominator and write

$$\begin{aligned}\Psi_{n,i} &\le \frac{\exp\left\{-n\frac{(\mu_{n,\mathrm{I}^\star}-\mu_{n,i})^2}{2\sigma^2\left(n/\mathrm{T}_{n,\mathrm{I}^\star}+n/\mathrm{T}_{n,i}\right)}\right\}}{\exp\left\{-n\left(\sup_{\boldsymbol{\omega}:\omega_{\mathrm{I}^\star}=\mathrm{T}_{n,\mathrm{I}^\star}/n}\min_{j\neq \mathrm{I}^\star}\frac{(\mu_{n,\mathrm{I}^\star}-\mu_{n,i})}{\sqrt{2\sigma^2\left(1/\omega_{\mathrm{I}^\star}+1/\omega_j\right)}}-\frac{1}{\sqrt{2n}}\right)^2\right\}} \\ &= \frac{\exp\left\{-n\frac{(\mu_{n,\mathrm{I}^\star}-\mu_{n,i})^2}{2\sigma^2\left(n/\mathrm{T}_{n,\mathrm{I}^\star}+n/\mathrm{T}_{n,i}\right)}\right\}}{\exp\left\{-n\left(\sqrt{\Gamma^\star_{\mathrm{T}_{n,\mathrm{I}^\star}/n}(\boldsymbol{\mu}_n)}-\frac{1}{\sqrt{2n}}\right)^2\right\}},\end{aligned}$$

where $\boldsymbol{\mu}_n \triangleq (\mu_{n,1},\cdots,\mu_{n,\mathrm{K}})$, and $(\beta,\boldsymbol{\mu})\mapsto\Gamma^\star_\beta(\boldsymbol{\mu})$ represents a function that maps β and $\boldsymbol{\mu}$ to the parameterized optimal error decay that any allocation rule can reach given parameter β and a set of arms with means $\boldsymbol{\mu}$. Note that this function is continuous with respect to β and $\boldsymbol{\mu}$ respectively.

Now, assuming $\Psi_{n,i}/n \ge \omega_i^\beta+\xi$ yields that there exists $\mathrm{S}_2' \triangleq \mathrm{Poly}(2/\xi,\mathrm{W}_2)$ s.t. for all $n>\mathrm{S}_2'$, $\mathrm{T}_{n,i}/n \ge \omega_i^\beta+\xi/2$, and by consequence,

$$\psi_{n,i} \le \exp\left\{-n\underbrace{\left(\frac{(\mu_{n,\mathrm{I}^\star}-\mu_{n,i})^2}{2\sigma^2\left(n/\mathrm{T}_{n,\mathrm{I}^\star}+1/(\omega_i^\beta+\xi/2)\right)}-\Gamma^\star_{\mathrm{T}_{n,\mathrm{I}^\star}/n}(\boldsymbol{\mu}_n)-\frac{1}{2n}+\sqrt{\frac{2\Gamma^\star_{\mathrm{T}_{n,\mathrm{I}^\star}/n}(\boldsymbol{\mu}_n)}{n}}\right)}_{\varepsilon_n(\xi)}\right\}.$$

Using Lemma 3.7, we know that for any ε, there exists $\mathrm{S}_3=\mathrm{Poly}(1/\varepsilon,\mathrm{W}_1,\mathrm{W}_2)$ s.t. $\forall n>\mathrm{S}_3$, $|\mathrm{T}_{n,\mathrm{I}^\star}/n-\beta|\le\varepsilon$, and $\forall j\in\mathscr{A}, |\mu_{n,j}-\mu_j|\le\varepsilon$. Furthermore, $(\beta,\boldsymbol{\mu})\mapsto\Gamma^\star_\beta(\boldsymbol{\mu})$ is continuous with respect to β and $\boldsymbol{\mu}$, thus for a given ε_0, there exists $\mathrm{S}_3'=\mathrm{Poly}(1/\varepsilon_0,\mathrm{W}_1,\mathrm{W}_2)$ s.t. $\forall n>\mathrm{S}_3'$, we have

$$\left|\varepsilon_n(\xi)-\left(\frac{(\mu_{\mathrm{I}^\star}-\mu_i)^2}{2\sigma^2\left(1/\beta+1/(\omega_i^\beta+\xi/2)\right)}-\Gamma^\star_\beta\right)\right|\le\varepsilon_0.$$

Finally, define $\mathrm{S}_1\triangleq\max\{\mathrm{S}_2,\mathrm{S}_2',\mathrm{S}_3'\}$, we have $\forall n>\mathrm{S}_1$,

$$\psi_{n,i}\le\exp\{-\varepsilon_0(\xi)n\},$$

where

$$\varepsilon_0(\xi)=\frac{(\mu_{\mathrm{I}^\star}-\mu_i)^2}{2\sigma^2\left(1/\beta+1/(\omega_i^\beta+\xi/2)\right)}-\Gamma^\star_\beta+\varepsilon_0. \tag{B.2}$$

□

Next, starting from some known moment, no arm is overly allocated. More precisely, we show the following lemma.

Lemma B.7. *Under TTTS, for every ξ, there exists $\mathrm{S}_4=\mathrm{Poly}(1/\xi,\mathrm{W}_1,\mathrm{W}_2)$ s.t. $\forall n>\mathrm{S}_4$,*

$$\forall i\in\mathscr{A},\ \frac{\Psi_{n,i}}{n}\le\omega_i^\beta+\xi.$$

Proof. From Lemma B.6, there exists $S_1' = \mathrm{Poly}(2/\xi, W_1, W_2)$ such that for all $n > S_1'$ and for all $i \neq I^\star$,

$$\frac{\Psi_{n,i}}{n} \geq \omega_i^\beta + \frac{\xi}{2} \;\Rightarrow\; \psi_{n,i} \leq \exp\{-\varepsilon_0(\xi/2)n\}.$$

Thus, for all $i \neq I^\star$,

$$\begin{aligned}\frac{\Psi_{n,i}}{n} &\leq \frac{S_1'}{n} + \frac{\sum_{\ell=S_1'+1}^{n} \psi_{\ell,i}\mathbb{1}\left(\frac{\Psi_{\ell,i}}{n} \geq \omega_i^\beta + \frac{\xi}{2}\right)}{n} + \frac{\sum_{\ell=S_1'+1}^{n} \psi_{\ell,i}\mathbb{1}\left(\frac{\Psi_{\ell,i}}{n} \leq \omega_i^\beta + \frac{\xi}{2}\right)}{n} \\ &\leq \frac{S_1'}{n} + \frac{\sum_{\ell=1}^{n} \exp\{-\varepsilon_0(\xi/2)n\}}{n} + \frac{\sum_{\ell=S_1'+1}^{\ell_n(\xi)} \psi_{\ell,i}\mathbb{1}\left(\frac{\Psi_{\ell,i}}{n} \leq \omega_i^\beta + \frac{\xi}{2}\right)}{n},\end{aligned}$$

where we let $\ell_n(\xi) = \max\left\{\ell \leq n : \Psi_{\ell,i}/n \leq \omega_i^\beta + \xi/2\right\}$. Then

$$\begin{aligned}\frac{\Psi_{n,i}}{n} &\leq \frac{S_1'}{n} + \frac{\sum_{\ell=1}^{n} \exp\{-\varepsilon_0(\xi/2)n\}}{n} + \Psi_{\ell_n(\xi),i} \\ &\leq \frac{S_1' + (1-\exp(-\varepsilon_0(\xi/2))^{-1}}{n} + \omega_i^\beta + \frac{\xi}{2}\end{aligned}$$

Then, there exists S_5 such that for all $n \geq S_5$,

$$\frac{S_1' + (1-\exp(-\varepsilon_0(\xi/2))^{-1}}{n} \leq \frac{\xi}{2}.$$

Therefore, for any $n > S_4 \triangleq \max\{S_1', S_5\}$, $\Psi_{n,i} \leq \omega_i^\beta + \xi$ holds for all $i \neq I^\star$. For $i = I^\star$, it is already proved for the optimal arm. □

We now prove Lemma 3.8 under TTTS.

Proof of Lemma 3.8 From Lemma B.7, there exists $S_4' = \mathrm{Poly}((K-1)/\xi, W_1, W_2)$ such that for all $n > S_4'$,

$$\forall i \in \mathscr{A}, \frac{\Psi_{n,i}}{n} \leq \omega_i^\beta + \frac{\xi}{K-1}.$$

Using the fact that $\Psi_{n,i}/n$ and ω_i^β all sum to 1, we have $\forall i \in \mathscr{A}$,

$$\begin{aligned}\frac{\Psi_{n,i}}{n} &= 1 - \sum_{j\neq i} \frac{\Psi_{n,j}}{n} \\ &\geq 1 - \sum_{j\neq i}\left(\omega_j^\beta + \frac{\xi}{K-1}\right) \\ &= \omega_i^\beta - \xi.\end{aligned}$$

Thus, for all $n > S_4'$, we have

$$\forall i \in \mathscr{A}, \left|\frac{\Psi_{n,i}}{n} - \omega_i^\beta\right| \leq \xi.$$

And finally we use the same reasoning as the proof of Lemma 3.7 to link $\mathrm{T}_{n,i}$ and $\Psi_{n,i}$. Fix an $\varepsilon > 0$. Using Lemma 3.4, we have $\forall n \in \mathbb{N}$,

$$\forall i \in \mathscr{A}, \left| \frac{\mathrm{T}_{n,i}}{n} - \frac{\Psi_{n,i}}{n} \right| \leq \frac{\mathrm{W}_2 \sqrt{(n+1)\log(e^2+n)}}{n}.$$

Thus there exists S_5 s.t. $\forall n > \mathrm{S}_5$,

$$\left| \frac{\mathrm{T}_{n,\mathrm{I}^\star}}{n} - \frac{\Psi_{n,\mathrm{I}^\star}}{n} \right| \leq \frac{\varepsilon}{2}.$$

And using the above result, there exists $\mathrm{S}_4'' = \mathrm{Poly}(2/\varepsilon, \mathrm{W}_1, \mathrm{W}_2)$ s.t. $\forall n > \mathrm{S}_4''$,

$$\left| \frac{\Psi_{n,i}}{n} - \omega_i^\beta \right| \leq \frac{\varepsilon}{2}.$$

Thus, if we take $\mathrm{N}_4 \triangleq \max\{\mathrm{S}_4'', \mathrm{S}_5\}$, then $\forall n > \mathrm{N}_4$, we have

$$\forall i \in \mathscr{A}, \left| \frac{\mathrm{T}_{n,i}}{n} - \omega_i^\beta \right| \leq \varepsilon.$$

■

B.4 Fixed-Confidence Analysis for T3C

This section is entirely dedicated to T3C. Note that the analysis to follow share the same proof line with that of TTTS, and some parts even completely coincide with those of TTTS. For the sake of simplicity and clearness, we shall only focus on the parts that differ and skip some redundant proofs.

B.4.1 Sufficient exploration of all arms

We prove Lemma 3.5 for T3C. To prove this lemma, we still need the two sets of indices for under-sampled arms like in Appendix B.3.1. We recall that for a given $\mathrm{L} > 0$: $\forall n \in \mathbb{N}$ we define

$$\mathrm{U}_n^{\mathrm{L}} \triangleq \{i : \mathrm{T}_{n,i} < \sqrt{\mathrm{L}}\},$$
$$\mathrm{V}_n^{\mathrm{L}} \triangleq \{i : \mathrm{T}_{n,i} < \mathrm{L}^{3/4}\}.$$

For T3C however, we investigate the following two indices,

$$\mathrm{J}_n^{(1)} \triangleq \arg\max_j a_{n,j}, \widetilde{\mathrm{J}_n^{(2)}} \triangleq \arg\min_{j \neq \mathrm{J}_n^{(1)}} \mathrm{W}_n(\mathrm{J}_n^{(1)}, j).$$

Lemma 3.5 is proved via the following sequence of lemmas.

Lemma B.8. *There exists* $\mathrm{L}_1 = \mathrm{Poly}(\mathrm{W}_1)$ *s.t. if* $\mathrm{L} > \mathrm{L}_1$, *for all* n, $\mathrm{U}_n^{\mathrm{L}} \neq \emptyset$ *implies* $\mathrm{J}_n^{(1)} \in \mathrm{V}_n^{\mathrm{L}}$ *or* $\widetilde{\mathrm{J}_n^{(2)}} \in \mathrm{V}_n^{\mathrm{L}}$.

Proof. If $\mathrm{J}_n^{(1)} \in \mathrm{V}_n^{\mathrm{L}}$, then the proof is finished. Now we assume that $\mathrm{J}_n^{(1)} \in \overline{\mathrm{V}_n^{\mathrm{L}}} \subset \overline{\mathrm{U}_n^{\mathrm{L}}}$, and we prove that $\mathrm{J}_n^{(2)} \in \mathrm{V}_n^{\mathrm{L}}$.

Step 1 Following the same reasoning as Step 1 and Step 2 of the proof of Lemma B.1, we know that there exists $L_2 = \text{Poly}(W_1)$ s.t. if $L > L_2$, then

$$\overline{J_n^\star} \triangleq \arg\max_{j \in \overline{U_n^L}} \mu_{n,j} = \arg\max_{j \in \overline{U_n^L}} \mu_j = J_n^{(1)}.$$

Step 2 Now assuming that $L > L_2$, and we show that for L large enough, $\widetilde{J_n^{(2)}} \in V_n^L$. In the same way that we proved (B.1) one can show that for all $\forall\, j \in \overline{V_n^L}$,

$$W_n(J_n^{(1)}, j) = \frac{(\mu_{n,I^\star} - \mu_{n,j})^2}{2\sigma^2\left(\frac{1}{T_{n,I^\star}} + \frac{1}{T_{n,j}}\right)} \geq \frac{L^{3/4}\Delta_{\min}^2}{16\sigma^2}.$$

Again, denote $J_n^\star \triangleq \arg\max_{j \in U_n^L} \mu_{n,j}$, we obtain

$$W_n(J_n^{(1)}, J_n^\star) = \begin{cases} 0 & \text{if } \mu_{n,J_n^\star} \geq \mu_{n,J_n^{(1)}}, \\ \dfrac{(\mu_{n,J_n^{(1)}} - \mu_{n,J_n^\star})^2}{2\sigma^2\left(\frac{1}{T_{n,J_n^{(1)}}} + \frac{1}{T_{n,J_n^\star}}\right)} & \text{else.} \end{cases}$$

In the second case, as already shown in Step 3 of Lemma B.1 we have that

$$\begin{aligned} |\mu_{n,J_n^\star} - \mu_{n,\overline{J_n^\star}}| &\leq \Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + T_{n,J_n^\star})}{1 + T_{n,J_n^\star}}} \\ &\leq \Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + \sqrt{L})}{1 + \sqrt{L}}}, \end{aligned}$$

since $J_n^\star \in U_n^L$. We also know that

$$2\sigma^2\left(\frac{1}{T_{n,J_n^{(1)}}} + \frac{1}{T_{n,J_n^\star}}\right) \geq \frac{2\sigma^2}{T_{n,J_n^\star}} \geq \frac{2\sigma^2}{\sqrt{L}}.$$

Therefore, we get

$$W_n(J_n^{(1)}, J_n^\star) \leq \frac{\sqrt{L}}{2\sigma^2}\left(\Delta_{\max} + 2\sigma W_1 \sqrt{\frac{\log(e + \sqrt{L})}{1 + \sqrt{L}}}\right)^2.$$

On the other hand, we know that for all $j \in \overline{V_n^L}$,

$$W_n(J_n^{(1)}, j) \geq \frac{L^{3/4}\Delta_{\min}^2}{16\sigma^2}.$$

Thus, there exists L_3 s.t. if $L > L_3$, then

$$\forall\, j \in \overline{V_n^L},\ W_n(J_n^{(1)}, j) \geq 2W_n(J_n^{(1)}, J_n^\star).$$

That means $\widetilde{J_n^{(2)}} \notin \overline{V_n^L}$ and by consequence, $\widetilde{J_n^{(2)}} \in V_n^L$.

Finally, taking $L_1 = \max(L_2, L_3)$, we have $\forall L > L_1$, either $J_n^{(1)} \in V_n^L$ or $\widetilde{J_n^{(2)}} \in V_n^L$. □

Next we show that there exists at least one arm in V_n^L for whom the probability of being pulled is large enough. More precisely, we prove the following lemma.

Lemma B.9. *There exists* $L_1 = Poly(W_1)$ *s.t. for* $L > L_1$ *and for all* n *s.t.* $U_n^L \neq \emptyset$, *then there exists* $J_n \in V_n^L$ *s.t.*

$$\psi_{n,J_n} \geq \frac{\min(\beta, 1-\beta)}{K^2} \triangleq \psi_{\min}.$$

Proof. Using Lemma B.8, we know that $J_n^{(1)}$ or $\widetilde{J_n^{(2)}} \in V_n^L$. We also know that under T3C, for any arm i, $\psi_{n,i}$ can be written as

$$\psi_{n,i} = \beta a_{n,i} + (1-\beta) \sum_{j \neq i} a_{n,j} \frac{\mathbb{1}\{W_n(j,i) = \min_{k \neq j} W_n(j,k)\}}{\left|\arg\min_{k \neq j} W_n(j,k)\right|}.$$

Note that $(\psi_{n,i})_i$ sums to 1,

$$\begin{aligned}\sum_i \psi_{n,i} &= \beta + (1-\beta) \sum_j a_{n,j} \sum_{i \neq j} \frac{\mathbb{1}\{W_n(j,i) = \min_{k \neq j} W_n(j,k)\}}{\left|\arg\min_{k \neq j} W_n(j,k)\right|} \\ &= \beta + (1-\beta) \sum_j a_{n,j} = 1\,.\end{aligned}$$

Therefore, we have

$$\psi_{n,J_n^{(1)}} \geq \beta a_{n,J_n^{(1)}} \geq \frac{\beta}{K}$$

on one hand, since $\sum_{i \in \mathcal{A}} a_{n,i} = 1$. On the other hand, we have

$$\begin{aligned}\psi_{n,\widetilde{J_n^{(2)}}} &\geq (1-\beta) \frac{a_{n,J_n^{(1)}}}{K} \\ &\geq \frac{1-\beta}{K^2},\end{aligned}$$

which concludes the proof. □

The rest of this subsection is exactly the same to that of TTTS. Indeed, with the above lemma, we can show that the set of poorly explored arms U_n^L is empty when n is large enough.

Lemma B.10. *Under T3C, there exists* $L_0 = Poly(W_1, W_2)$ *s.t.* $\forall L > L_0$, $U_{\lfloor KL \rfloor}^L = \emptyset$.

Proof. See proof of Lemma B.3 in Appendix B.3.1. □

We can finally conclude the proof of Lemma 3.5 for T3C in the same way as for TTTS in Appendix B.3.1. ■

B.4.2 Concentration of the empirical means

We prove Lemma 3.6 for T3C. As a corollary of the previous section, we can show the concentration of $\mu_{n,i}$ to μ_i, and the proof remains the same as that of TTTS in Appendix B.3.2.

B.4.3 Measurement effort concentration of the optimal arm

Next, we show that the empirical arm draws proportion of the true best arm for T3C concentrates to β when the total number of arm draws is sufficiently large. We prove Lemma 3.7 for T3C.

This proof also remains the same as that of TTTS in Appendix B.3.3.

B.4.4 Measurement effort concentration of other arms

In this section, we show that, for T3C, the empirical measurement effort concentration also holds for other arms than the true best arm. We prove Lemma 3.8 for T3C. Note that this part differs from that of TTTS.

We again establish first an over-allocation implies negligible probability result as follow.

Lemma B.11. *Under T3C, for every $\xi \le \varepsilon_0$ with ε_0 problem dependent, there exists $S_1 = Poly(1/\xi, W_1, W_2)$ such that for all $n > S_1$, for all $i \neq I^\star$,*

$$\frac{\Psi_{n,i}}{n} \ge \omega_i^\beta + 2\xi \;\Rightarrow\; \psi_{n,i} \le (K-1)\exp\left\{-\frac{\Delta_{min}^2}{16\sigma^2}\sqrt{\frac{n}{K}}\right\}.$$

Proof. Fix $i \neq I^\star$ s.t. $\Psi_{n,i}/n \ge \omega_i^\beta + 2\xi$, then using Lemma 3.4, there exists $S_2 = \text{Poly}(1/\xi, W_2)$ such that for any $n > S_2$, we have

$$\frac{T_{n,i}}{n} \ge \omega_i^\beta + \xi.$$

Then,

$$\begin{aligned}
\psi_{n,i} &\le \beta a_{n,i} + (1-\beta)\sum_{j\neq i} a_{n,j}\mathbb{1}\{W_n(j,i) = \min_{k\neq j} W_n(j,k)\} \\
&\le \beta a_{n,i} + (1-\beta)\left(\sum_{j\neq i, I^\star} a_{n,j} + a_{n,I^\star}\mathbb{1}\{W_n(I^\star,i) = \min_{k\neq I^\star} W_n(I^\star,k)\}\right) \\
&\le \sum_{j\neq I^\star} a_{n,j} + \mathbb{1}\{W_n(I^\star,i) = \min_{k\neq I^\star} W_n(I^\star,k)\}.
\end{aligned}$$

Next we show that the indicator function term in the previous inequality equals to 0.

Using Lemma 3.3 and Lemma 3.7 for T3C, there exists $S_3 = \text{Poly}(1/\xi, W_1, W_2)$ such that for any $n > S_3$,

$$\left|\frac{T_{n,I^\star}}{n} - \beta\right| \le \xi^2 \text{ and } \forall j \in \mathscr{A}, |\mu_{n,j} - \mu_j| \le \xi^2.$$

Now if $\forall j \neq I^\star, i$, we have $T_{n,j}/n > \omega_j^\beta$, then

$$\begin{aligned}
\frac{n-1}{n} &= \sum_{j\in\mathscr{A}} \frac{T_{n,j}}{n} \\
&= \frac{T_{n,I^\star}}{n} + \frac{T_{n,i}}{n} + \sum_{j\neq I^\star, i} \frac{T_{n,j}}{n} \\
&> \beta - \varepsilon^2 + \omega_i^\beta + \varepsilon + \sum_{j\neq I^\star, i} \omega_j^\beta \ge 1,
\end{aligned}$$

which is a contradiction.

Thus there exists at least one $j_0 \neq I^\star, i$, such that $T_{n,j_0}/n \leq \omega_j^\beta$. Assuming $n > \max(S_2, S_3)$, we have

$$\begin{aligned} W_n(I^\star, i) - W_n(I^\star, j_0) &= \frac{(\mu_{n,I^\star} - \mu_{n,i})^2}{2\sigma^2\left(\frac{1}{T_{n,I^\star}} + \frac{1}{T_{n,i}}\right)} - \frac{(\mu_{n,I^\star} - \mu_{n,j_0})^2}{2\sigma^2\left(\frac{1}{T_{n,I^\star}} + \frac{1}{T_{n,j_0}}\right)} \\ &\geq \underbrace{\frac{(\mu_{I^\star} - \mu_i - 2\xi^2)^2}{2\sigma^2\left(\frac{1}{\beta - \xi^2} + \frac{1}{\omega_i^\beta + \xi}\right)} - \frac{(\mu_{I^\star} - \mu_{j_0} + 2\xi^2)^2}{2\sigma^2\left(\frac{1}{\beta + \xi^2} + \frac{1}{\omega_{j_0}^\beta}\right)}}_{W_{i,j_0}^\xi}. \end{aligned}$$

According to Proposition 3.1, W_{i,j_0}^ξ converges to 0 when ξ goes to 0, more precisely we have

$$W_{i,j_0}^\xi = \frac{(\mu_{I^\star} - \mu_i)^2}{2\sigma^2}\left(\frac{\beta}{\beta + \omega_i^\beta}\right)^2 \xi + O(\xi^2),$$

thus there exists a ε_0 such that for all $\xi < \varepsilon_0$ it holds for all $i, j_0 \neq I^\star$, $W_{i,j_0}^\xi > 0$. It follows then

$$W_n(I^\star, i) - \min_{k \neq I^\star} W_n(I^\star, k) \geq W_n(I^\star, i) - W_n(I^\star, j_0) > 0,$$

and $\mathbb{1}\{W_n(I^\star, i) = \min_{k \neq I^\star} W_n(I^\star, k)\} = 0$.

Knowing that Lemma B.4 is also valid for T3C, thus there exists $M_1 = \text{Poly}(4/\Delta_{\min}, W_1, W_2)$ such that for all $n > M_1$,

$$\forall j \neq I^\star, a_{n,j} \leq \exp\left\{-\frac{\Delta_{\min}^2}{16\sigma^2}\sqrt{\frac{n}{K}}\right\},$$

which then concludes the proof by taking $S_1 \triangleq \max(M_1, S_2, S_3)$. □

The rest of this subsection almost coincides with that of TTTS. We first show that, starting from some known moment, no arm is overly allocated. More precisely, we show the following lemma.

Lemma B.12. *Under T3C, for every ξ, there exists $S_4 = \text{Poly}(1/\xi, W_1, W_2)$ s.t. $\forall n > S_4$,*

$$\forall i \in \mathscr{A}, \quad \frac{\Psi_{n,i}}{n} \leq \omega_i^\beta + 2\xi.$$

Proof. See proof of Lemma B.7 in Appendix B.3.4. Note that the previous step does not match exactly that of TTTS, so the proof would be slightly different. However, the difference is only a matter of constant, we thus still choose to skip this proof. □

It remains to prove Lemma 3.8 for T3C, which stays the same as that of TTTS.

Proof of Lemma 3.8 for T3C See proof of Lemma 3.8 for TTTS in Appendix B.3.4. ■

B.5 Proof of Lemma 3.1

Finally, it remains to prove Lemma 3.1 under the Gaussian case before we can conclude for Theorem 3.1 for TTTS or T3C.

Lemma 3.1. *Let* $\delta, \beta \in (0,1)$. *For any sampling rule which satisfies* $\mathbb{E}\left[T_\beta^\varepsilon\right] < \infty$ *for all* $\varepsilon > 0$, *we have*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \leq T_\beta^\star(\boldsymbol{\mu}),$$

if the sampling rule is coupled with stopping rule (3.3),

For the clarity, we recall the definition of generalized likelihood ratio. For any pair of arms i, j, We first define a weighted average of their empirical means,

$$\widehat{\mu}_{n,i,j} \triangleq \frac{T_{n,i}}{T_{n,i}+T_{n,j}}\widehat{\mu}_{n,i} + \frac{T_{n,j}}{T_{n,i}+T_{n,j}}\widehat{\mu}_{n,j}.$$

And if $\widehat{\mu}_{n,i} \geq \widehat{\mu}_{n,j}$, then the generalized likelihood ratio $Z_{n,i,j}$ for Gaussian noise distributions has the following analytic expression,

$$Z_{n,i,j} \triangleq T_{n,i}\, d(\widehat{\mu}_{n,i}; \widehat{\mu}_{n,i,j}) + T_{n,j}\, d(\widehat{\mu}_{n,j}; \widehat{\mu}_{n,i,j}).$$

We further define a statistic Z_n as

$$Z_n \triangleq \max_{i \in \mathscr{A}} \min_{j \in \mathscr{A}\setminus\{i\}} Z_{n,i,j}.$$

The following lemma stated by Qin et al. [2017] is needed in our proof.

Lemma B.13. *For any* $\zeta > 0$, *there exists* ε *s.t.* $\forall n \geq T_\beta^\varepsilon$, $Z_n \geq (\Gamma_\beta^\star - \zeta)n$.

To prove Lemma 3.1, we need the Gaussian tail inequality (3.7) of Lemma 3.2.

Proof. We know that

$$\begin{aligned}
1 - a_{n,I^\star} &= \sum_{i \neq I^\star} a_{n,i} \\
&\leq \sum_{i \neq I^\star} \Pi_n[\theta_i > \theta_{I^\star}] \\
&= \sum_{i \neq I^\star} \Pi_n[\theta_i - \theta_{I^\star} > 0] \\
&\leq (K-1) \max_{i \neq I^\star} \Pi_n[\theta_i - \theta_{I^\star} > 0].
\end{aligned}$$

We can further rewrite $\Pi_n\left[\theta_i - \theta_{I^\star} > 0\right]$ as

$$\Pi_n\left[\theta_i - \theta_{I^\star} > \mu_{n,i} - \mu_{n,I^\star} + \mu_{n,I^\star} - \mu_{n,i}\right].$$

We choose ε sufficiently small such that the empirical best arm $I_n^\star = I^\star$. Then, for all $n \geq T_\beta^n$ and for any $i \neq I^\star$, $\mu_{n,I^\star} \geq \mu_{n,i}$. Thus, fix any $\zeta \in (0, \Gamma_\beta^\star/2)$ and apply inequality (3.7) of Lemma 3.2 with $\mu_{n,I^\star}$ and $\mu_{n,i}$, we have for any $n \geq T_\beta^\varepsilon$,

$$\begin{aligned}
1 - a_{n,I^\star} &\leq (K-1) \max_{i \neq I^\star} \frac{1}{2} \exp\left\{ -\frac{\left(\mu_{n,I^\star} - \mu_{n,i}\right)^2}{2\sigma^2_{n,i,I^\star}} \right\} \\
&= \frac{(K-1)\exp\{-Z_n\}}{2} \\
&\leq \frac{(K-1)\exp\left\{-(\Gamma_\beta^\star - \zeta)n\right\}}{2}.
\end{aligned}$$

The last inequality is deduced from Lemma B.13. By consequence,

$$\forall n \geq T_\beta^\varepsilon, \ln\left(1 - a_{n,I^\star}\right) \leq \ln \frac{K-1}{2} - (\Gamma_\beta^\star - \zeta)n.$$

On the other hand, we have for any n,

$$1 - c_{n,\delta} = \frac{\delta}{2n(K-1)\sqrt{2\pi e}\exp\left\{\sqrt{2\ln\frac{2n(K-1)}{\delta}}\right\}}.$$

Thus, there exists a deterministic time N s.t. $\forall n \geq N$,

$$\begin{aligned}
\ln\left(1 - c_{n,\delta}\right) &= \ln \frac{\delta}{(K-1)\sqrt{8\pi e}} - \ln n - \sqrt{2\ln\frac{2n(K-1)}{\delta}} \\
&\geq \ln \frac{\delta}{2(K-1)\sqrt{2\pi e}} - \zeta n.
\end{aligned}$$

Let $C_3 \triangleq (K-1)^2\sqrt{2\pi e}$, we have for any $n \geq N_0 \triangleq T_\beta^\varepsilon + N$,

$$\ln\left(1 - a_{n,I^\star}\right) - \ln\left(1 - c_{n,\delta}\right) \leq \ln \frac{C_3}{\delta} - (\Gamma_\beta^\star - 2\zeta)n, \tag{B.3}$$

and it is clear that $\mathbb{E}[N_0] < \infty$.

Let us consider the following two cases:

Case 1 There exists $n \in [1, N_0]$ s.t. $a_{n,I^\star} \geq c_{n,\delta}$, then by definition,

$$\tau_\delta \leq n \leq N_1.$$

Case 2 For any $n \in [1, N_0]$, we have $a_{n,I^\star} < c_{n,\delta}$, then $\tau_\delta \geq N_0 + 1$, thus by Equation B.3,

$$\begin{aligned}
0 &\leq \ln\left(1 - a_{\tau_\delta - 1, I^\star}\right) - \ln\left(1 - c_{\tau_\delta - 1, \delta}\right) \\
&\leq \ln \frac{C_3}{\delta} - (\Gamma_\beta^\star - 2\zeta)(\tau_\delta - 1),
\end{aligned}$$

and we obtain

$$\tau_\delta \leq \frac{\ln(C_3/\delta)}{\Gamma^\star_\beta - 2\zeta} + 1.$$

Combining the two cases, and we have for any $\zeta \in (0, \Gamma^\star_\beta/2)$,

$$\begin{aligned}\tau_\delta &\leq \max\left\{N_0, \frac{\ln(C_3/\delta)}{\Gamma^\star_\beta - 2\zeta} + 1\right\} \\ &\leq N_0 + 1 + \frac{\ln(C_3)}{\Gamma^\star_\beta - 2\zeta} + \frac{\ln(1/\delta)}{\Gamma^\star_\beta - 2\zeta}.\end{aligned}$$

Since $\mathbb{E}[N_1] < \infty$, therefore

$$\limsup_\delta \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \leq \frac{1}{\Gamma^\star_\beta - 2\zeta}, \forall \zeta \in (0, \Gamma^\star_\beta/2),$$

which concludes the proof. □

B.6 Proof of Posterior Convergence for Gaussian Bandits

B.6.1 Proof of Theorem 3.4

Theorem 3.4. *Under TTTS, for Gaussian bandits with improper Gaussian priors, it holds almost surely that*

$$\lim_{n\to\infty} -\frac{1}{n}\log(1 - a_{n,I^\star}) = T^\star_\beta(\boldsymbol{\mu})^{-1}.$$

From Theorem 2 in Qin et al. [2017], any allocation rule satisfying $T_{n,i}/n \to \omega_i^\beta$ for each $i \in \mathscr{A}$, satisfies

$$\lim_{n\to\infty} -\frac{1}{n}\log(1 - a_{n,I^\star}) = \Gamma^\star_\beta.$$

Therefore, to prove Theorem 3.4, it is sufficient to prove that under TTTS,

$$\forall i \in \{1, \dots, K\}, \quad \lim_{n\to\infty} \frac{T_{n,i}}{n} \overset{a.s}{=} \omega_i^\beta. \tag{B.4}$$

Due to the concentration result in Lemma 3.4 that we restate below (and proved in Appendix B.3), which will be useful at several places in the proof, observe that

$$\lim_{n\to\infty} \frac{T_{n,i}}{n} \overset{a.s}{=} \omega_i^\beta \quad \Leftrightarrow \quad \lim_{n\to\infty} \frac{\Psi_{n,i}}{n} \overset{a.s}{=} \omega_i^\beta,$$

therefore it suffices to establish the convergence of $\overline{\psi}_{n,i} = \Psi_{n,i}/n$ to ω_i^β, which we do next. For that purpose, we need again the following maximality inequality lemma.

Lemma 3.4. *There exists a random variable* W_2*, such that for all* $i \in \mathscr{A}$*,*

$$\forall n \in \mathbb{N}, |T_{n,i} - \Psi_{n,i}| \leq W_2 \sqrt{(n+1)\log(e^2+n)} \text{ a.s.},$$

and $\mathbb{E}\left[e^{\lambda W_2}\right] < \infty$ *for any* $\lambda > 0$.

Step 1: TTTS draws all arms infinitely often and satisfies $T_{n,I^\star}/n \to \beta$**.** More precisely, we prove the following lemma.

Lemma B.14. *Under TTTS, it holds almost surely that*

1. *for all* $i \in \mathscr{A}$, $\lim_{n\to\infty} T_{n,i} = \infty$.
2. $a_{n,I^\star} \to 1$.
3. $T_{n,I^\star}/n \to \beta$.

Proof. Our first ingredient is a lemma showing the implications of finite measurement, and consistency when all arms are sampled infinitely often. Its proof follows standard posterior concentration arguments and is given in Appendix B.6.2.

Lemma B.15. *[Consistency and implications of finite measurement] Denote with* $\overline{\mathscr{I}}$ *the arms that are sampled only a finite amount of times:*

$$\overline{\mathscr{I}} = \{i \in \{1,\dots,k\} : \forall n, T_{n,i} < \infty\}.$$

If $\overline{\mathscr{I}}$ *is empty,* $a_{n,i}$ *converges almost surely to* 1 *when* $i = I^\star$ *and to* 0 *when* $i \neq I^\star$*. If* $\overline{\mathscr{I}}$ *is non-empty, then for every* $i \in \overline{\mathscr{I}}$*, we have* $\liminf_{n\to\infty} a_{n,i} > 0$ *a.s.*

First we show that $\sum_{n\in\mathbb{N}} T_{n,j} = \infty$ for each arm j. Suppose otherwise. Let $\overline{\mathscr{I}}$ again be the set of arms to which only finite measurement effort is allocated. Under TTTS, we have

$$\psi_{n,i} = a_{n,i}\left(\beta + (1-\beta)\sum_{j\neq i}\frac{a_{n,j}}{1-a_{n,j}}\right),$$

so $\psi_{n,i} \geq \beta a_{n,i}$. Therefore, by Lemma B.15, if $i \in \overline{\mathscr{I}}$, then $\liminf a_{n,i} > 0$ implies that $\sum_n \psi_{n,i} = \infty$. By Lemma 3.4, we then must have that $\lim_{n\to\infty} T_{n,i} = \infty$ as well: contradiction. Thus, $\lim_{n\to\infty} T_{n,i} = \infty$ for all i, and we conclude that $a_{n,I^\star} \to 1$, by Lemma B.15.

For TTTS with parameter β this implies that $\overline{\psi}_{n,I^\star} \to \beta$, and since we have a bound on $|T_{n,i}/n - \overline{\psi}_{n,i}|$ in Lemma 3.4, we have $T_{n,I^\star}/n \to \beta$ as well. □

Step 2: Controlling the over-allocation of sub-optimal arms. The convergence of $T_{n,I^\star}/n$ to β leads to following interesting consequence, expressed in Lemma B.16: if an arm is sampled more often than its optimal proportion, the posterior probability of this arm to be optimal is reduced compared to that of other sub-optimal arms.

Lemma B.16. *[Over-allocation implies negligible probability][3] Fix any $\xi > 0$ and $j \neq I^\star$. With probability 1, under any allocation rule, if $T_{n,I^\star}/n \to \beta$, there exist $\xi' > 0$ and a sequence ε_n with $\varepsilon_n \to 0$ such that for any $n \in \mathbb{N}$,*

$$\frac{T_{n,j}}{n} \geq \omega_j^\beta + \xi \Rightarrow \frac{a_{n,j}}{\max_{i \neq I^\star} a_{n,i}} \leq e^{-n(\xi' + \varepsilon_n)}.$$

Proof. We have $\Pi_n(\Theta_{\cup i \neq I^\star}) = \sum_{i \neq I^\star} a_{n,i} = 1 - a_{n,I^\star}$, therefore $\max_{i \neq I^\star} a_{n,i} \leq 1 - a_{n,I^\star}$. By Theorem 2 of Qin et al. [2017] we have, as $T_{n,I^\star}/n \to \beta$,

$$\limsup_{n \to \infty} -\frac{1}{n} \log \left(\max_{i \neq I^\star} a_{n,i} \right) \leq \Gamma_\beta^\star.$$

We also have the following from the standard Gaussian tail inequality, for $n \geq \tau$ after which $\mu_{n,I^\star} \geq \mu_{n,i}$, using that $\theta_i - \theta_{I^\star} \sim \mathcal{N}(\mu_{n,i} - \mu_{n,I^\star}, \sigma_{n,i}^2 + \sigma_{n,I^\star}^2)$ and $\sigma_{n,i}^2 + \sigma_{n,I^\star}^2 = \sigma^2(1/T_{n,i} + 1/T_{n,I^\star})$,

$$a_{n,i} \leq \Pi_n(\theta_i \geq \theta_{I^\star}) \leq \exp\left(\frac{-(\mu_{n,i} - \mu_{n,I^\star})^2}{2\sigma^2(1/T_{n,I^\star} + 1/T_{n,i})} \right) = \exp\left(-n \frac{(\mu_{n,i} - \mu_{n,1})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,i})} \right).$$

Thus, there exists a sequence $\varepsilon_n \to 0$, for which

$$\begin{aligned} \frac{a_{n,j}}{\max_{i \neq I^\star} a_{n,i}} &\leq \frac{\exp\left\{ -n \left(\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} - \varepsilon_n/2 \right) \right\}}{\exp\left\{ -n \left(\Gamma_\beta^\star + \varepsilon_n/2 \right) \right\})} \\ &= \exp\left\{ -n \left(\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} - \Gamma_\beta^\star - \varepsilon_n \right) \right\}. \end{aligned}$$

Now we take a look at the two terms in the middle:

$$\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} - \Gamma_\beta^\star.$$

Note that the first term is increasing in $T_{n,j}/n$. We have the definition from Qin et al. [2017], for any $j \neq I^\star$,

$$\Gamma_\beta^\star = \frac{(\mu_j - \mu_{I^\star})^2}{2\sigma^2 \left(1/\omega_{I^\star}^\beta + 1/\omega_j^\beta \right)},$$

and we have the premise

$$\frac{T_{n,j}}{n} \geq \omega_j^\beta + \xi.$$

Combining these with the convergence of the empirical means to the true means (consistency, see Lemma B.15), we can conclude that for all $\varepsilon > 0$, there exists a time n_0 such that for all later times $n \geq n_0$, we have

$$\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} \geq \frac{(\mu_j - \mu_{I^\star})^2}{2\sigma^2 \left(1/\beta + n/T_{n,j} \right)} - \varepsilon \geq \frac{(\mu_j - \mu_{I^\star})^2}{2\sigma^2 \left(1/\beta + 1/(\omega_j^\beta + \xi) \right)} - \varepsilon > \Gamma_\beta^\star,$$

where the first inequality follows from consistency, the second from monotonicity in $T_{n,j}/n$. That means that there exist a $\xi' > 0$ such that

$$\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} - \Gamma^\star_\beta > \xi',$$

and thus the claim follows that when $\frac{T_{n,j}}{n} \geq \omega_j^\beta + \xi$, we have

$$\frac{a_{n,j}}{\max_{i \neq I^\star} a_{n,i}} \leq \exp\left\{-n\left(\frac{(\mu_{n,j} - \mu_{n,I^\star})^2}{2\sigma^2(n/T_{n,I^\star} + n/T_{n,j})} - \Gamma^\star_\beta - \varepsilon_n\right)\right\} \leq e^{-n(\xi' + \varepsilon_n)}.$$

□

Step 3: $\overline{\psi}_{n,i}$ converges to ω_i^β for all arms. To establish the convergence of the allocation effort of all arms, we rely on the same sufficient condition used in the analysis of Russo [2016], that we recall below.

Lemma B.17. *[Sufficient condition for optimality][4] Consider any adaptive allocation rule. If we have*

$$\overline{\psi}_{n,I^\star} \to \beta, \quad \textit{and} \quad \sum_{n \in \mathbb{N}} \psi_{n,j} \mathbf{1}\left\{\overline{\psi}_{n,j} \geq \omega_j^\beta + \xi\right\} < \infty, \quad \forall j \neq I^\star, \xi > 0, \tag{B.5}$$

then $\overline{\psi}_n \to \psi^\beta$.

First, note that from Lemma B.14 we know that $T_{n,I^\star}/n \to \beta$, an by Lemma 3.4 this implies $\overline{\psi}_{n,I^\star} \to \beta$, hence we can use Lemma B.17 to prove convergence to the optimal proportions. Thus, we now show that (B.5) holds under TTTS. Recall that $J_n^{(1)} = \arg\max_j a_{n,j}$ and $J_n^{(2)} = \arg\max_{j \neq J_n^{(1)}} a_{n,j}$. Since $a_{n,I^\star} \to 1$ by Lemma B.14, there is some finite time τ after which for all $n > \tau$, $J_n^{(1)} = I^\star$. Under TTTS,

$$\begin{aligned}
\psi_{n,i} &= a_{n,i}\left(\beta + (1-\beta)\sum_{j \neq i} \frac{a_{n,j}}{1 - a_{n,j}}\right) \\
&\leq a_{n,i}\beta + a_{n,i}(1-\beta)\frac{\sum_{j \neq i} a_{n,j}}{1 - a_{n,J_n^{(1)}}} \\
&\leq a_{n,i}\beta + a_{n,i}(1-\beta)\frac{\sum_{j \neq i} a_{n,j}}{a_{n,J_n^{(2)}}} \\
&\leq a_{n,i}\beta + a_{n,i}(1-\beta)\frac{1}{a_{n,J_n^{(2)}}} \\
&\leq \frac{a_{n,i}}{a_{n,J_n^{(2)}}},
\end{aligned}$$

where we use the fact that for $j \neq J_n^{(1)}$, we have $a_{n,J_n^{(1)}} \geq a_{n,j}$ and $a_{n,J_n^{(2)}} \leq 1 - a_{n,J_n^{(1)}}$. For $n \geq \tau$ this means that $\psi_{n,i} \leq a_{n,i}/\max_{j \neq I^\star} a_{n,i}$ for any $i \neq I^\star$.

By Lemma B.16, there is a constant $\xi' > 0$ such and a sequence $\varepsilon_n \to 0$ such that

$$T_{n,i}/n \geq w_i^\beta + \xi \Rightarrow \frac{a_{n,i}}{\max_{j \neq I^\star} a_{n,j}} \leq e^{-n(\xi' - \varepsilon_n)}.$$

Now take a time τ large enough, such that for $n \geq \tau$ we have $|T_{n,j}/n - \overline{\psi}_{n,j}| \leq \xi$ (which can be found by Lemma 3.4). Then we have

$$\mathbb{1}\left\{\overline{\psi}_{n,j} \geq \psi_j^\beta + \xi\right\} \leq \mathbb{1}\left\{\frac{T_{n,j}}{n} \geq \omega_j^\beta + 2\xi\right\}$$

Therefore, for all $i \neq I^\star$, we have

$$\sum_{n\geq\tau} \psi_{n,i}\mathbb{1}\left\{\overline{\psi}_{n,j} \geq \psi_j^\beta + \xi\right\} \leq \sum_{n\geq\tau} \psi_{n,i}\mathbb{1}\left\{\frac{T_{n,j}}{n} \geq \omega_j^\beta + 2\xi\right\} \leq \sum_{n\geq\tau} e^{-n(\xi'-\varepsilon_n)} < \infty.$$

Thus (B.5) holds and the convergence to the optimal proportions follows by Lemma B.17.

B.6.2 Proof of auxiliary lemmas

Proof of Lemma B.15 Let $\overline{\mathscr{I}}$ be nonempty. Define

$$\mu_{\infty,n} \triangleq \lim_{n\to\infty} \mu_{n,i}, \text{ and } \sigma^2_{\infty,i} \triangleq \lim_{n\to\infty} \sigma^2_{n,i},$$

and recall that for $i \in \mathscr{A}$ for which $T_{n,i} = 0$, we have $\mu_{n_i} = \mu_{1,i} = 0$ and $\sigma^2_{n,i} = \sigma^2_{1,i} = \infty$, and if $T_{n,i} > 0$, we have

$$\mu_{n,i} = \frac{1}{T_{n,i}} \sum_{\ell=1}^{n-1} \mathbb{1}\{I_\ell = i\} Y_{\ell,I_\ell}, \text{ and } \sigma^2_{n,i} = \frac{\sigma^2}{T_{n,i}}.$$

For all arms that are sampled infinitely often, we therefore have $\mu_{\infty,i} = \mu_i$ and $\sigma^2_{\infty,i} = 0$. For all arms that are sampled only a finite number of times, i.e. $i \in \overline{\mathscr{I}}$, we have $\sigma^2_{\infty,i} > 0$, and there exists a time n_0 after which for all $n \geq n_0$ and $i \in \overline{\mathscr{I}}$, we have $T_{n,i} = T_{n_0,i}$. Define

$$\Pi_\infty \triangleq \mathcal{N}(\mu_{\infty,1}, \sigma^2_{\infty,1}) \otimes \mathcal{N}(\mu_{\infty,2}, \sigma^2_{\infty,2}) \otimes \ldots \otimes \mathcal{N}(\mu_{\infty,k}, \sigma^2_{\infty,k}) = \bigotimes_{i\notin\overline{\mathscr{I}}} \delta_{\mu_i} \otimes \bigotimes_{i\in\overline{\mathscr{I}}} \Pi_{n_0}.$$

Then for each $i \in \mathscr{A}$ we define

$$a_{\infty,i} \triangleq \Pi_\infty\left(\theta_i > \max_{j\neq i} \theta_j\right).$$

Then we have for all $i \in \overline{\mathscr{I}}$, $a_{\infty,i} \in (0,1)$, since $\sigma^2_{\infty,i} > 0$, and thus $a_{\infty,I^\star} < 1$.

When $\overline{\mathscr{I}}$ is empty, we have $a_{n,I^\star} = \Pi_n(\theta_{I^\star} > \max_{i\neq I^\star} \theta_i)$, but since $\Pi_\infty = \bigotimes_{i\in\mathscr{A}} \delta_{\mu_i}$, we have $a_{\infty,I^\star} = 1$ and $a_{\infty,i} = 0$ for all $i \neq I^\star$.

■

B.7 Proof of Posterior Convergence for Bernoulli Bandits

B.7.1 Preliminaries

We first introduce a crucial Beta tail bound inequality. Let $F^{\text{Beta}}_{a,b}$ denote the cdf of a Beta distribution with parameters a and b, and $F^{\text{B}}_{c,d}$ the cdf of a Binomial distribution with parameters c and d, then we have the following relationship, often called the 'Beta-Binomial trick',

$$F^{\text{Beta}}_{a,b}(y) = 1 - F^{\text{B}}_{a+b-1,y}(a-1),$$

so that we have

$$\mathbb{P}[\mathrm{X} \geq x] = \mathbb{P}\left[\mathrm{B}_{a+b-1,x} \leq a-1\right] = \mathbb{P}\left[\mathrm{B}_{a+b-1,1-x} \geq b\right].$$

We can bound Binomial tails with Sanov's inequality:

$$\frac{e^{-nd(k/n,x)}}{n+1} \leq \mathbb{P}\left[\mathrm{B}_{n,x} \geq k\right] \leq e^{-nd(k/n,x)},$$

where the last inequalities hold when $k \geq nx$.

Lemma B.18. *Let* $\mathrm{X} \sim \mathscr{B}eta(a,b)$ *and* $\mathrm{Y} \sim \mathscr{B}eta(c,d)$ *with* $0 < \frac{a-1}{a+b-1} < \frac{c-1}{c+d-1}$. *Then we have* $\mathbb{P}[\mathrm{X} > \mathrm{Y}] \leq \mathrm{D}e^{-\mathrm{C}}$ *where*

$$\mathrm{C} = \inf_{\frac{a-1}{a+b-1} \leq y \leq \frac{c-1}{c+d-1}} \mathrm{C}_{a,b}(y) + \mathrm{C}_{c,d}(y),$$

and

$$\mathrm{D} = 3 + \min\left(\mathrm{C}_{a,b}\left(\frac{c-1}{c+d-1}\right), \mathrm{C}_{c,d}\left(\frac{a-1}{a+b-1}\right)\right).$$

Note that this lemma is the Bernoulli version of Lemma 3.2.

Theorem B.1. *Consider the Beta-Bernoulli setting. For* $\beta \in (0,1)$, *under any allocation rule satisfying* $\mathrm{T}_{n,\mathrm{I}^\star}/n \to \omega^\beta_{\mathrm{I}^\star}$,

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,\mathrm{I}^\star}) \leq \Gamma^\star_\beta,$$

and under any allocation rule satisfying $\mathrm{T}_{n,i}/n \to \omega^\beta_i$ *for each* $i \in \mathscr{A}$,

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,\mathrm{I}^\star}) = \Gamma^\star_\beta.$$

Proof. Denote again with $\overline{\mathscr{I}}$ again the set of arms sampled only finitely many times. For $\overline{\mathscr{I}}$ empty, we thus have $\mu_{\infty,i} \triangleq \lim_{n\to\infty} \mu_{n,i} = \mu_i$. The posterior variance is

$$\begin{aligned}\sigma^2_{n,i} &= \frac{\alpha_{n,i}\beta_{n,i}}{(\alpha_{n,i}+\beta_{n,i})^2(\alpha_{n,i}+\beta_{n,i}+1)} \\ &= \frac{(1+\sum_{\ell=1}^{n-1}\mathbb{1}\{\mathrm{I}_\ell = i\}\mathrm{Y}_{\ell,\mathrm{I}_\ell})(1+\mathrm{T}_{n,i}-\sum_{\ell=1}^{n-1}\mathbb{1}\{\mathrm{I}_\ell = i\}\mathrm{Y}_{\ell,\mathrm{I}_\ell})}{(2+\mathrm{T}_{n,i})^2(2+\mathrm{T}_{n,i}+1)}.\end{aligned}$$

We see that when $\overline{\mathscr{I}}$ is empty, we have $\sigma^2_{\infty,i} \triangleq \lim_{n\to\infty}\sigma^2_{n,i} = 0$, i.e., the posterior is concentrated.

Step 1: A lower bound when some arms are sampled only finitely often. First, note that when $T_{n,i} = 0$ for some $i \in \mathscr{A}$, the empirical mean for that arm equals the prior mean $\mu_{n,i} = \alpha_{0,i}/(\alpha_{0,i}+\beta_{0,i})$, and the variance is strictly positive:

$$\sigma^2_{n,i} = (\alpha_{0,i}\beta_{0,i})/\left((\alpha_{0,i}+\beta_{0,i})^2(\alpha_{0,i}+\beta_{0,i}+1)\right) > 0\,.$$

When $\overline{\mathscr{I}}$ is not empty, then for every $i \in \overline{\mathscr{I}}$ we have $\sigma^2_{\infty,i} > 0$, and $a_{\infty,i} \in (0,1)$, implying $a_{\infty,I^\star} < 1$, and thus

$$\lim_{n\to\infty} -\frac{1}{n}\log\left(1-a_{n,I^\star}\right) = -\frac{1}{n}\log\left(1-a_{\infty,I^\star}\right) = 0.$$

Step 2: A lower bound when every arm is sampled infinitely often. Suppose now that $\overline{\mathscr{I}}$ is empty, then we have

$$\max_{i\neq I^\star}\Pi_n(\theta_i \geq \theta_{I^\star}) \leq 1 - a_{n,I^\star} \leq \sum_{i\neq I^\star}\Pi_n(\theta_i \geq \theta_{I^\star}) \leq (k-1)\max_{i\neq I^\star}\Pi_n(\theta_i \geq \theta_{I^\star}).$$

Thus, we have $1 - a_{n,I^\star} \leq (k-1)\max_{i\neq I^\star}\Pi_n(\theta_i \geq \theta_{I^\star})$ and also $1 - a_{n,I^\star} \doteq \max_{i\neq I^\star}\Pi_n(\theta_i \geq \theta_{I^\star})$. We have

$$\begin{aligned}
\Gamma^\star &= \max_{w\in W}\min_{i\neq I^\star} C_i(\omega_{I^\star},\omega_i),\\
\Gamma^\star_\beta &= \max_{w\in W;\omega_{I^\star}=\beta}\min_{i\neq I^\star} C_i(\beta,\omega_i), \text{ with}\\
C_i(\omega_{I^\star},\omega_i) &= \min_{x\in\mathbb{R}} \omega_{I^\star} d(\theta_{I^\star};x) + \omega_i d(\theta_i;x) = \omega_{I^\star} d(\theta_{I^\star};\overline{\theta}) + \omega_i d(\theta_i;\overline{\theta}),
\end{aligned}$$

where $\overline{\theta} \in [\theta_i, \theta_{I^\star}]$ is the solution to

$$A'(\overline{\theta}) = \frac{\omega_{I^\star}A'(\theta_{I^\star}) + \omega_i A'(\theta_i)}{\omega_{I^\star}+\omega_i}.$$

Since every arm is sampled infinitely often, when n is large, we have $\mu_{n,I^\star} > \mu_{n,i}$. Define $S_{n,i} \triangleq \sum_{\ell=1}^{n-1}\mathbb{1}\{I_\ell = i\}Y_{\ell,I_\ell}$. Recall that the posterior is a Beta distribution with parameters $a_{n,i} = S_{n,i}+1$ and $\beta_{n,i} = T_{n,i} - S_{n,i} + 1$. Let $\tau \in \mathbb{N}$ be such that for every $n \geq \tau$, we have $S_{n,i}/(T_{n,i}+1) < S_{n,I^\star}/(T_{n,I^\star}+1)$. For the sake of simplicity, we define for any $i \in \mathscr{A}$ the interval

$$I_{i,I^\star} \triangleq \left[\frac{S_{n,i}}{T_{n,i}+1}, \frac{S_{n,I^\star}}{T_{n,I^\star}+1}\right].$$

Then using Lemma B.18 with $a = S_{n,i}+1$, $b = T_{n,i}-S_{n,i}+1$, $c = S_{n,I^\star}+1$, $d = T_{n,I^\star}-S_{n,I^\star}+1$, we have

$$\Pi_n(\theta_i - \theta_{I^\star} \geq 0) \leq D\exp\left\{-\inf_{y\in I_{i,I^\star}} C_{S_{n,i}+1,T_{n,i}-S_{n,i}+1}(y) + C_{S_{n,I^\star}+1,T_{n,I^\star}-S_{n,I^\star}+1}(y)\right\}.$$

This implies

$$\frac{1}{n}\log\left(\frac{\Pi_n(\theta_i \geq \theta_{I^\star})}{\exp\left\{-\inf_{y\in I_{i,I^\star}} C_{S_{n,i}+1,T_{n,i}-S_{n,i}+1}(y) + C_{S_{n,I^\star}+1,T_{n,I^\star}-S_{n,I^\star}+1}(y)\right\}}\right) \leq \frac{1}{n}\log(D),$$

which goes to zero as n goes to infinity. Indeed replacing a, b, c, d by their values in the definition of D we get

$$\begin{aligned}
D &\leq 3 + (T_{n,i}-1)\,kl\left(\frac{S_{n,i}}{T_{n,i}+1}; \frac{S_{n,I^\star}}{T_{n,I^\star}+1}\right)\\
&\leq 3 + (n+1)\,kl\left(0; \frac{n}{n+1}\right) = (n+1)\log(n+1)\,.
\end{aligned}$$

Hence,

$$\Pi_n(\theta_i \geq \theta_{\mathrm{I}^\star}) \doteq \exp\left\{-\inf_{y\in \mathrm{I}_{i,\mathrm{I}^\star}} \mathrm{C}_{\mathrm{S}_{n,i}+1,\mathrm{T}_{n,i}-\mathrm{S}_{n,i}+1}(y) + \mathrm{C}_{\mathrm{S}_{n,\mathrm{I}^\star}+1,\mathrm{T}_{n,\mathrm{I}^\star}-\mathrm{S}_{n,\mathrm{I}^\star}+1}(y)\right\}.$$

We thus have for any i,

$$\begin{aligned}
1-a_{n,i} &\doteq \max_{j\neq \mathrm{I}^\star} \Pi_n\left[\theta_j \geq \theta_{\mathrm{I}^\star}\right] \\
&\doteq \max_{j\neq \mathrm{I}^\star} \exp\left\{-\inf_{y\in \mathrm{I}_{j,\mathrm{I}^\star}} \mathrm{C}_{\mathrm{S}_{n,j}+1,\mathrm{T}_{n,j}-\mathrm{S}_{n,j}+1}(y) + \mathrm{C}_{\mathrm{S}_{n,\mathrm{I}^\star}+1,\mathrm{T}_{n,\mathrm{I}^\star}-\mathrm{S}_{n,\mathrm{I}^\star}+1}(y)\right\} \\
&\doteq \exp\left\{-n\min_{j\neq \mathrm{I}^\star} \inf_{y\in \mathrm{I}_{j,\mathrm{I}^\star}} \frac{\mathrm{T}_{n,j}+1}{n} kl\left(\frac{\mathrm{S}_{n,j}}{\mathrm{T}_{n,j}+1}; y\right) + \frac{\mathrm{T}_{n,\mathrm{I}^\star}+1}{n} kl\left(\frac{\mathrm{S}_{n,\mathrm{I}^\star}}{\mathrm{T}_{n,\mathrm{I}^\star}+1}; y\right)\right\} \\
&\geq \exp\left\{-n\max_{\boldsymbol{\omega}}\min_{j\neq \mathrm{I}^\star} \inf_{y\in \mathrm{I}_{j,\mathrm{I}^\star}} \omega_i kl\left(\frac{\mathrm{S}_{n,j}}{\mathrm{T}_{n,j}+1}; y\right) + \omega_{\mathrm{I}^\star} kl\left(\frac{\mathrm{S}_{n,\mathrm{I}^\star}}{\mathrm{T}_{n,j}+1}; y\right)\right\}.
\end{aligned}$$

Fix some $\varepsilon > 0$, then there exists some $n_0(\varepsilon)$ such that for all $n \geq n_0(\varepsilon)$, we have for any j,

$$\mathrm{I}_{j,\mathrm{I}^\star} = \left[\frac{\mathrm{S}_{n,j}}{\mathrm{T}_{n,j}+1}, \frac{\mathrm{S}_{n,\mathrm{I}^\star}}{\mathrm{T}_{n,\mathrm{I}^\star}+1},\right] \subset \left[\mu_j + \varepsilon, \mu_{\mathrm{I}^\star} - \varepsilon\right] \triangleq \mathrm{I}^\star_{j,\varepsilon},$$

and because KL-divergence is uniformly continuous on the compact interval $\mathrm{I}^\star_{j,\varepsilon}$, there exists an n_1 such that for every $n \geq n_1$ we have

$$kl\left(\frac{\mathrm{S}_{n,j}}{\mathrm{T}_{n,j}+1}; y\right) \geq (1-\varepsilon) kl\left(\mu_j; y\right),$$

for any y and for all $j \in \mathscr{A}$. Therefore, we have

$$\begin{aligned}
1-a_{n,i} &\doteq \exp\left\{-n\max_{\boldsymbol{\omega}}\min_{j\neq \mathrm{I}^\star} \inf_{y\in \mathrm{I}_{j,\mathrm{I}^\star}} \omega_j kl\left(\frac{\mathrm{S}_{n,j}}{\mathrm{T}_{n,j}+1}; y\right) + \omega_{\mathrm{I}^\star} kl\left(\frac{\mathrm{S}_{n,\mathrm{I}^\star}}{\mathrm{T}_{n,\mathrm{I}^\star}+1}; y\right)\right\} \\
&\geq \exp\left\{-n\max_{\boldsymbol{\omega}}\min_{i\neq \mathrm{I}^\star} \inf_{y\in \mathrm{I}^\star_{j,\varepsilon}} \omega_i kl(\mu_j; y) + \omega_{\mathrm{I}^\star} kl(\mu_{\mathrm{I}^\star}; y)\right\}.
\end{aligned}$$

Therefore, we have

$$\limsup_{n\to\infty} -\frac{1}{n}\log(1-a_{n,i}) \leq \Gamma^\star.$$

If $\mathrm{T}_{n,i}/n \to \omega_i^\star$ for each $i \in \mathscr{A}$, we have

$$\begin{aligned}
&\lim_{n\to\infty} \inf_{y\in \mathrm{I}_{i,\mathrm{I}^\star}} \frac{\mathrm{T}_{n,i}+1}{n} kl\left(\frac{\mathrm{S}_{n,i}}{\mathrm{T}_{n,i}+1}; y\right) + \frac{\mathrm{T}_{n,\mathrm{I}^\star}+1}{n} kl\left(\frac{\mathrm{S}_{n,\mathrm{I}^\star}}{\mathrm{T}_{n,i}+1}; y\right) \\
&= \inf_{y\in[\mu_i,\mu_{\mathrm{I}^\star}]} \omega_i^\star kl(\mu_i; y) + \omega_{\mathrm{I}^\star}^\star kl(\mu_{\mathrm{I}^\star}; y) \\
&= \Gamma^\star,
\end{aligned}$$

and thus

$$\begin{aligned}
1-a_{n,i} &\doteq \exp\left\{-n\max_{\boldsymbol{\omega}}\min_{j\neq \mathrm{I}^\star} \inf_{y\in \mathrm{I}^\star_\varepsilon} \omega_i kl(\mu_j; y) + \omega_{\mathrm{I}^\star} kl(\mu_{\mathrm{I}^\star}; y)\right\} \\
&\doteq \exp\left\{-n\Gamma^\star\right\},
\end{aligned}$$

implying

$$\lim_{n\to\infty} -\frac{1}{n}\log\left(1-a_{n,i}\right) = \Gamma^{\star}.$$

Everything goes similarly when $\omega_{I^\star} = \beta \in (0,1)$, so under any sampling rule satisfying $T_{n,I^\star}/n \to \beta$ we have

$$\limsup_{n\to\infty} -\frac{1}{n}\log(1-a_{n,i}) \leq \Gamma^{\star}_{\beta}$$

and under any sampling rule satisfying $T_{n,i}/n \to \omega_i^{\beta}$ for each $i \in \mathscr{A}$, we have

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,i}) = \Gamma^{\star}_{\beta}.$$

□

B.7.2 Proof of Theorem 3.5

Theorem 3.5. *Under TTTS, for Bernoulli bandits and uniform priors, it holds almost surely that*

$$\lim_{n\to\infty} -\frac{1}{n}\log(1-a_{n,I^\star}) = T^{\star}_{\beta}(\boldsymbol{\mu})^{-1}.$$

From Theorem B.1 we know that under any allocation rule satisfying $T_{n,i}/n \to \omega_i^{\beta}$ for every $i \in \mathscr{A}$, we have

$$\lim_{n\to\infty} -\frac{1}{n}\log\left(1-a_{n,I^\star}\right) = \Gamma^{\star}_{\beta}.$$

Thus, we only need to prove that under TTTS, for all $i \in \mathscr{A}$, we have

$$\lim_{n\to\infty} \frac{T_{n,i}}{n} \overset{a.s}{=} \omega_i^{\beta}.$$

Just as for the proof of the Gaussian case, we can use Lemma 3.4 (proof in Appendix B.6.2), which implies

$$\lim_{n\to\infty} \frac{T_{n,i}}{n} \overset{a.s}{=} \omega_i^{\beta} \quad \Leftrightarrow \quad \lim_{n\to\infty} \frac{\Psi_{n,i}}{n} \overset{a.s}{=} \omega_i^{\beta}.$$

Therefore, it suffices to show convergence for $\overline{\psi}_{n,i} = \Psi_{n,i}/n$ to ω_i^{β}, which we will do next, following the same steps as in the proof for the Gaussian case.

Step 1: TTTS draws all arms infinitely often and satisfies $T_{n,I^\star}/n \to \beta$. We prove the following lemma.

Lemma B.19. *Under TTTS, it holds almost surely that*

1. *for all $i \in \mathscr{A}$, $\lim_{n\to\infty} T_{n,i} = \infty$.*
2. *$a_{n,I^\star} \to 1$.*
3. *$\frac{T_{n,I^\star}}{n} \to \beta$.*

Proof. First, we give a lemma showing the implications of finite measurement, and consistency when all arms are sampled infinitely often, which provides a proof for 2. The proof of this lemma follows from the proof of Theorem B.1, and is given in Appendix B.7.3.

Lemma B.20. *[Consistency and implications of finite measurement] Denote with $\overline{\mathscr{I}}$ the arms that are sampled only a finite amount of times:*

$$\overline{\mathscr{I}} = \{i \in \{1,\dots,k\} : \forall n, \mathrm{T}_{n,i} < \infty\}.$$

If $\overline{\mathscr{I}}$ is empty, $a_{n,i}$ converges almost surely to 1 when $i = \mathrm{I}^\star$ and to 0 when $i \neq \mathrm{I}^\star$. If $\overline{\mathscr{I}}$ is non-empty, then for every $i \in \overline{\mathscr{I}}$, we have $\liminf_{n\to\infty} a_{n,i} > 0$ a.s.

Now we can show 1. of Lemma B.19: we show that under TTTS, for each $j \in \mathrm{A}$, we have $\sum_{n\in\mathbb{N}} \mathrm{T}_{n,j} = \infty$. The proof is exactly equal to the proof for Gaussian arms.

Under TTTS, we have

$$\psi_{n,i} = a_{n,i}\left(\beta + (1-\beta)\sum_{j\neq i}\frac{a_{n,j}}{1-a_{n,j}}\right),$$

so $\psi_{n,i} \geq \beta a_{n,i}$, therefore, by Lemma B.15, if $i \in \overline{\mathscr{I}}$, then $\liminf a_{n,i} > 0$ implies that $\sum_n \psi_{n,i} = \infty$. By Lemma 3.4, we then must have that $\lim_{n\to\infty} \mathrm{T}_{n,i} = \infty$ as well: contradiction. Thus, $\lim_{n\to\infty} \mathrm{T}_{n,i} = \infty$ for all i, and we conclude that $a_{n,\mathrm{I}^\star} \to 1$, by Lemma B.15.

Lastly we prove point 3. of Lemma B.19. For TTTS with parameter β, the above implies that $\overline{\psi}_{n,\mathrm{I}^\star} \to \beta$, and since we have a bound on $|\mathrm{T}_{n,i}/n - \overline{\psi}_{n,i}|$ in Lemma 3.4, we have $\mathrm{T}_{n,\mathrm{I}^\star}/n \to \beta$ as well.

□

Step 2: Controlling the over-allocation of sub-optimal arms. Following the proof for the Gaussian case again, we can establish a consequence of the convergence of $\mathrm{T}_{n,\mathrm{I}^\star}/n$ to β : if an arm is sampled more often than its optimal proportion, the posterior probability of this arm to be optimal is reduced compared to that of other sub-optimal arms. We can prove this by using ingredients from the proof of the lower bound in Theorem B.1.

Lemma B.21. *[Over-allocation implies negligible probability][5] Fix any $\xi > 0$ and $j \neq \mathrm{I}^\star$. With probability 1, under any allocation rule, if $\mathrm{T}_{n,\mathrm{I}^\star}/n \to \beta$, there exist $\xi' > 0$ and a sequence ε_n with $\varepsilon_n \to 0$ such that for any $n \in \mathbb{N}$,*

$$\frac{\mathrm{T}_{n,j}}{n} \geq \omega_j^\beta + \xi \implies \frac{a_{n,j}}{\max_{i\neq\mathrm{I}^\star} a_{n,i}} \leq e^{-n(\xi'+\varepsilon_n)}.$$

Proof. By Theorem B.1, we have, as $\mathrm{T}_{n,\mathrm{I}^\star}/n \to \beta$,

$$\limsup_{n\to\infty} -\frac{1}{n}\log\left(\max_{i\neq\mathrm{I}^\star} a_{n,i}\right) \leq \Gamma^\star_\beta,$$

since $\max_{i\neq\mathrm{I}^\star} a_{n,i} \leq 1 - a_{n,\mathrm{I}^\star}$. We also have from Lemma B.18 a deviation inequality, so that we can establish the following logarithmic equivalence:

$$a_{n,j} \leq \Pi_n(\theta_j \geq \theta_{\mathrm{I}^\star}) \doteq \exp\left\{-n\mathrm{C}_j\left(w_{n,\mathrm{I}^\star}, \omega_{n,j}\right)\right\} \doteq \exp\left\{-n\mathrm{C}_j\left(\beta, \omega_{n,j}\right)\right\},$$

where we denote $\omega_{n,j} \triangleq \frac{T_{n,j}}{n}$. We can combine these results, which implies that there exists a non-negative sequence $\varepsilon_n \to 0$ such that

$$\frac{a_{n,j}}{\max_{i \neq I^\star} a_{n,i}} \leq \frac{\exp\left\{-nC_j\left(\beta, \omega_{n,j}\right) - \varepsilon_n/2\right\}}{\exp\left\{-n(\Gamma^\star_\beta + \varepsilon/2)\right\}} = \exp\left\{-n\left(C_j\left(\beta, \omega_{n,j}\right) - \Gamma^\star_\beta\right) - \varepsilon_n\right\}.$$

We know that $C_j\left(\beta, \omega_j^\beta\right)$ is strictly increasing in ω_j^β, and $C_j\left(\beta, \omega_j^\beta\right) = \Gamma^\star_\beta$, thus, there exists some $\xi' > 0$ such that

$$\omega_{n,j} \geq \omega_j^\beta + \xi \implies C_j\left(\beta, \omega_{n,j}\right) - \Gamma^\star_\beta > \xi'.$$

□

Step 3: $\overline{\psi}_{n,i}$ converges to ω_i^β for all arms. To establish the convergence of the allocation effort of all arms, we rely on the same sufficient condition used in the analysis of Russo [2016], restated above in Lemma B.17, and we will restate it here again for convenience.

Lemma B.22. *[Sufficient condition for optimality] Consider any adaptive allocation rule. If*

$$\overline{\psi}_{n,I^\star} \to \beta, \quad \textit{and} \quad \sum_{n \in \mathbb{N}} \psi_{n,j} \mathbf{1}\left\{\overline{\psi}_{n,j} \geq \omega_j^\beta + \xi\right\} < \infty, \ \forall j \neq I^\star, \xi > 0, \tag{B.6}$$

then $\overline{\psi}_n \to \psi^\beta$.

First, note that from Lemma B.19 we know that $\frac{T_{n,I^\star}}{n} \to \beta$, and by Lemma 3.4 this implies $\overline{\psi}_{n,I^\star} \to \beta$, hence we can use the lemma above to prove convergence to the optimal proportions. This proof is already given in Step 3 of the proof for the Gaussian case, and since it does not depend on the specifics of the Gaussian case, except for invoking Lemma B.15 (consistency), which for the Bernoulli case we replace by Lemma B.20, it gives a proof for the Bernoulli case as well. We conclude that (B.5) holds, and the convergence to the optimal proportions follows by Lemma B.17.

B.7.3 Proof of auxiliary lemmas

Proof of Lemma B.18

$$\begin{aligned}
\mathbb{P}[X > Y] = \mathbb{E}[\mathbb{P}[X > Y|Y]] &\leq \mathbb{E}\left[\mathbb{1}\{Y < \frac{a-1}{a+b-1}\} + \mathbb{1}\{Y \geq \frac{a-1}{a+b-1}\}\mathbb{P}[X > Y|Y]\right] \\
&\leq \exp\left\{-(c+d-1)kl\left(\frac{c-1}{c+d-1}; \frac{a-1}{a+b-1}\right)\right\} \\
&\quad + \mathbb{E}\left[\exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; Y\right)\right\}\mathbb{1}\{Y \geq \frac{a-1}{a+b-1}\}\right],
\end{aligned}$$

Using the Beta-Binomial trick in the second inequality. Then we have (call the second half A)

$$\begin{aligned}
A \leq \mathbb{E}\left[\mathbb{1}\{\frac{a-1}{a+b-1} \leq Y \leq \frac{c-1}{c+d-1}\}\right] &\exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; Y\right)\right\} \\
+ \exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; \frac{c-1}{c+d-1}\right)\right\}&
\end{aligned}$$

(call the first half B). Denote with f the density of Y, then

$$\mathrm{B} = \int_{\frac{a-1}{a+b-1}}^{\frac{c-1}{c+d-1}} \exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; y\right)\right\} f(y)\,\mathrm{d}y.$$

Via integration by parts we obtain

$$\begin{aligned}
\mathrm{B} &= \left[\exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; y\right)\right\}\mathbb{P}\left[\mathrm{Y}\le y\right]\right]_{\frac{a-1}{a+b-1}}^{\frac{c-1}{c+d-1}} \\
&+ \int_{\frac{a-1}{a+b-1}}^{\frac{c-1}{c+d-1}} (a+b-1)\frac{\mathrm{d}}{\mathrm{d}y}kl\left(\frac{a-1}{a+b-1}; y\right)\exp\left\{-\mathrm{C}_{a,b}(y)\right\}\mathrm{P}(\mathrm{Y}\le y)\,\mathrm{d}y \\
&\le \int_{\frac{a-1}{a+b-1}}^{\frac{c-1}{c+d-1}} (a+b-1)\frac{\mathrm{d}}{\mathrm{d}y}kl\left(\frac{a-1}{a+b-1}; y\right)\exp\left\{-(\mathrm{C}_{a,b}(y)+\mathrm{C}_{c,d}(y))\right\}\mathrm{d}y \\
&+ \exp\left\{-(a+b-1)kl\left(\frac{a-1}{a+b-1}; \frac{c-1}{c+d-1}\right)\right\},
\end{aligned}$$

where the first inequality uses the Binomial trick again. Let

$$\begin{aligned}
\mathrm{C} &= \inf_{\frac{a-1}{a+b-1}\le y\le\frac{c-1}{c+d-1}} (a+b-1)kl\left(\frac{a-1}{a+b-1}; y\right) + (c+d-1)kl\left(\frac{c-1}{c+d-1}; y\right) \\
&= \inf_{\frac{a-1}{a+b-1}\le y\le\frac{c-1}{c+d-1}} \mathrm{C}_{a,b}(y) + \mathrm{C}_{c,d}(y),
\end{aligned}$$

then note that in particular we have

$$\begin{aligned}
\mathrm{C} &\le \min\left((a+b-1)kl\left(\frac{a-1}{a+b-1}; \frac{c-1}{c+d-1}\right), (c+d-1)kl\left(\frac{c-1}{c+d-1}; \frac{a-1}{a+b-1}\right)\right) \\
&= \min\left(\mathrm{C}_{a,b}\left(\frac{c-1}{c+d-1}\right), \mathrm{C}_{c,d}\left(\frac{a-1}{a+b-1}\right)\right).
\end{aligned}$$

Then

$$\begin{aligned}
\mathrm{B} &\le e^{-\mathrm{C}} \int_{\frac{a-1}{a+b-1}}^{\frac{c-1}{c+d-1}} (a+b-1)\frac{\mathrm{d}}{\mathrm{d}y}kl(\frac{a-1}{a+b-1}; y)\,\mathrm{d}y + e^{-\mathrm{C}} \\
&= \left[(a+b-1)kl\left(\frac{a-1}{a+b-1}; \frac{c-1}{c+d-1}\right)+1\right]e^{-\mathrm{C}}.
\end{aligned}$$

Thus we have

$$\mathbb{P}\left[\mathrm{X}>\mathrm{Y}\right] \le \left(3+(a+b-1)kl\left(\frac{a-1}{a+b-1}; \frac{c-1}{c+d-1}\right)\right)e^{-\mathrm{C}}.$$

By symmetry, we have

$$\mathbb{P}\left[\mathrm{X}>\mathrm{Y}\right] \le \left(3+\min\left(\mathrm{C}_{a,b}\left(\frac{c-1}{c+d-1}\right), \mathrm{C}_{c,d}\left(\frac{a-1}{a+b-1}\right)\right)\right)e^{-\mathrm{C}},$$

where

$$\mathrm{C} = \inf_{\frac{a-1}{a+b-1}\le y\le\frac{c-1}{c+d-1}} (a+b-1)kl\left(\frac{a-1}{a+b-1}; y\right) + (c+d-1)kl\left(\frac{c-1}{c+d-1}; y\right).$$

■

Proof of Lemma B.20 Let $\overline{\mathscr{I}}$ be empty, then we have $\mu_{\infty,i} \triangleq \lim_{n\to\infty} \mu_{n,i} = \mu_i$. The posterior variance is

$$\sigma^2_{n,i} = \frac{\alpha_{n,i}\beta_{n,i}}{(\alpha_{n,i}+\beta_{n,i})^2(\alpha_{n,i}+\beta_{n,i}+1)} = \frac{(1+\sum_{\ell=1}^{n-1}\mathbb{1}\{\mathrm{I}_\ell = i\}\mathrm{Y}_{\ell,\mathrm{I}_\ell})(1+\mathrm{T}_{n,i}-\sum_{\ell=1}^{n-1}\mathbb{1}\{\mathrm{I}_\ell = i\}\mathrm{Y}_{\ell,\mathrm{I}_\ell})}{(2+\mathrm{T}_{n,i})^2(2+\mathrm{T}_{n,i}+1)},$$

We see that when $\overline{\mathscr{I}}$ is empty, we have $\sigma^2_{\infty,i} \triangleq \lim_{n\to\infty}\sigma^2_{n,i} = 0$, i.e., the posterior is concentrated.

When $\mathrm{T}_{n,i} = 0$ for some $i \in \mathscr{A}$, the empirical mean for that arm equals to the prior mean $\mu_{n,i} = \alpha_{1,i}/(\alpha_{1,i}+\beta_{1,i})$, and the variance is strictly positive:

$$\sigma^2_{n,i} = (\alpha_{n,i}\beta_{n,i})/\left((\alpha_{1,i}+\beta_{1,i})^2(\alpha_{1,i}+\beta_{1,i}+1)\right) > 0\,.$$

When $\overline{\mathscr{I}}$ is not empty, then for every $i \in \overline{\mathscr{I}}$ we have $\sigma^2_{\infty,i} > 0$, and $\alpha_{\infty,i} \in (0,1)$, implying $\alpha_{\infty,\mathrm{I}^\star} < 1$, hence the posterior is not concentrated.

■

Appendix C

Additional Proofs of Chapter 4

C.1 Notation

Table C.1: Table of notation for Chapter 4.

Notation	Meaning
Θ	set of parameters
M	upper bound on the norm of θ
$\mathscr{X}$	finite set or arms
K	number of arms
$\mathscr{Y}$	transductive set
B	number of elements in the transductive set
$\mathscr{I}$	finite set of answers
A	number of answers
L	upper bound on the norms of the arms
$\boldsymbol{\theta}$	parameter in Θ
$\widehat{\mathbf{x}}_n$	arm pulled at time n
$\mathrm{T}_n^{\mathbf{x}} = \sum_{t=1}^n \mathbb{1}_{\{\widehat{\mathbf{x}}_t = \mathbf{x}\}}$	number of draws of arm $\mathbf{x}$ up to time n
$\mathrm{T}_{n,i}$	number of draws of arm indexed i up to time n
$\mathbf{T}_n = (\mathrm{T}_n^{\mathbf{x}})_{\mathbf{x}\in\mathscr{X}}$	vector of number of draws
$\mathrm{T}_n^{\mathbf{x},i} = \sum_{t=1}^n \mathbb{1}_{\{\widehat{\mathbf{x}}_t = \mathbf{x}, i_t = i\}}$	number of draws of arm $\mathbf{x}$ for a given answer i
λ	regularization parameter
$\widehat{\boldsymbol{\theta}}_n^{\lambda}$	regularized least square estimate

C.2 Technical Lemmas

C.2.1 Lagrangian lemma

Lemma C.1. *For $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, $\boldsymbol{\omega}$ in the interior of the probability simplex $\mathring{\Sigma}_{\mathrm{K}}$, $\mathbf{y} \in \mathbb{R}^d$, $x \in \mathbb{R}$, we have*

$$\inf_{\boldsymbol{\theta}' :\, \mathbf{y}^\mathsf{T}\boldsymbol{\theta}' \geq x} \frac{\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\omega}}{2} = \begin{cases} \dfrac{(x-\mathbf{y}^\mathsf{T}\boldsymbol{\theta}')^2}{2\left\|\mathbf{y}\right\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}} & \text{if } x \geq \mathbf{y}^\mathsf{T}\boldsymbol{\theta}' \\ 0 & \text{otherwise} \end{cases}.$$

Proof. We consider the Lagrangian of the problem, and we obtain

$$\begin{aligned} \inf_{\boldsymbol{\theta}' :\, \mathbf{y}^\mathsf{T}\boldsymbol{\theta}' \geq x} \frac{\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\omega}}{2} &= \sup_{\lambda \geq 0} \inf_{\boldsymbol{\theta}' \in \mathbb{R}^d} \frac{\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_\omega}}{2} + \lambda(x - \mathbf{y}^\mathsf{T}\boldsymbol{\theta}') \\ &= \sup_{\lambda \geq 0} \lambda(x-\mathbf{y}^\mathsf{T}\boldsymbol{\theta}) - \lambda^2 \frac{\left\|\mathbf{y}\right\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}}{2} \\ &= \begin{cases} \dfrac{(x-\mathbf{y}^\mathsf{T}\boldsymbol{\theta}')^2}{2\left\|\mathbf{y}\right\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}} & \text{if } x \geq \mathbf{y}^\mathsf{T}\boldsymbol{\theta}' \\ 0 & \text{otherwise} \end{cases}, \end{aligned}$$

where the infimum in the first equality is reached at $\boldsymbol{\theta}' = \boldsymbol{\theta} + \lambda \boldsymbol{\Lambda}_\omega^{-1}\mathbf{y}$ and the supremum in the last equality is reached at

$$\lambda = \begin{cases} \dfrac{(x-\mathbf{y}^\mathsf{T}\boldsymbol{\theta})}{\left\|\mathbf{y}\right\|^2_{\boldsymbol{\Lambda}_\omega^{-1}}} & \text{if } x \geq \mathbf{y}^\mathsf{T}\boldsymbol{\theta} \\ 0 & \text{otherwise} \end{cases}.$$

□

C.2.2 Concentration results

We restate here the Theorem 20.4 (in combination with the Equation 20.10) by Lattimore and Szepesvari [2018].

Theorem C.1. *For all* $\lambda > 0$ *and* $\delta \in (0,1)$,

$$\mathbb{P}_{\boldsymbol{\theta}}\left[\exists n \in \mathbb{N},\, \frac{1}{2}\left\|\widehat{\boldsymbol{\theta}}^\lambda_n - \boldsymbol{\theta}\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n} + \lambda \mathrm{I}_d} \geq d_{n,\delta}\right] \leq \delta,$$

where

$$\begin{aligned} d_{n,\delta} &\triangleq \left(\sqrt{\log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1 + \frac{n\mathrm{L}^2}{\lambda d}\right)} + \sqrt{\frac{\lambda}{2}}\mathrm{M}\right)^2 \\ &= \log\left(\frac{1}{\delta}\right) + \frac{d}{2}\log\left(1 + \frac{n\mathrm{L}^2}{\lambda d}\right) + \mathrm{M}\sqrt{\lambda}\sqrt{2\log\left(\frac{1}{\delta}\right) + d\log\left(1 + \frac{n\mathrm{L}^2}{\lambda d}\right)} + \frac{\lambda \mathrm{M}^2}{2}. \end{aligned}$$

C.2.3 Other technical lemmas

We regroup in this section some other useful technical lemmas.

Lemma C.2. *For all* $\alpha, y \geq 0$, *if for some* $x \geq 0$ *if holds* $y \geq x - \alpha\sqrt{x}$ *then*

$$x \leq y + \alpha\sqrt{y} + \alpha^2.$$

Proof. Just note that for $z = \sqrt{x}$ we have

$$z^2 - \alpha z - y \leq 0\,,$$

thus

$$x \leq \frac{1}{4}\left(\alpha + \sqrt{\alpha^2 + 4y}\right)^2 \leq y + \frac{\alpha^2}{2} + \frac{\alpha}{2}\sqrt{\alpha^2 + 4y} \leq y + \alpha\sqrt{y} + \alpha^2\,.$$

□

We then state a result derived from the concavity of $\mathbf{\Lambda} \mapsto \log\det(\mathbf{\Lambda})$.

Lemma C.3. *Let $(w_t)_{t\geq 1}$ be a sequence in Σ_K and $\lambda > 0$ then*

$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_{\mathbf{x}}^s \|\mathbf{x}\|^2_{W_s+\lambda I_d} \leq d\log\left(1 + \frac{tL^2}{d\lambda}\right).$$

where $W_t = \sum_{s=1}^{t} w_s$.

Proof. Define the function $f(W) = \log\det(\mathbf{\Lambda}_W + \lambda I_d)$ for any $W \in (\mathbb{R}^+)^K$. It is a concave function since the function $\mathbf{\Lambda} \mapsto \log\det(\mathbf{\Lambda})$ is a concave function over the set of positive definite matrices (see Exercise 21.2 of Lattimore and Szepesvari 2018). And its partial derivative with respect to the coordinate a at W is

$$\nabla_{\mathbf{x}} f(W) = \|\mathbf{x}\|^2_{(W+\lambda I_d)^{-1}}\,.$$

Hence using the concavity of f we have

$$\sum_{\mathbf{x}\in\mathscr{X}} w_{\mathbf{x}}^s \|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{W_s}+\lambda I_d)^{-1}} = \langle W_s - W_{s-1}, \nabla_{\mathbf{x}} f(W_s)\rangle \leq f(W_s) - f(W_{s-1})\,.$$

Which implies that

$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_{\mathbf{x}}^s \|\mathbf{x}\|^2_{\mathbf{\Lambda}_{W_s}+\lambda I_d} \leq f(W_t) - f(W_0) = \log\left(\frac{\det(\mathbf{\Lambda}_{W_t} + \lambda I_d)}{\det(\lambda I_d)}\right) \leq d\log\left(1 + \frac{tL^2}{d\lambda}\right),$$

where for the last inequality we use the inequality of arithmetic and geometric means in combination with $\mathrm{Tr}(W_t) \leq tL^2$. □

A simple consequence of the previous lemma follows.

Lemma C.4. *For all t,*

$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} \widetilde{w}_s^{\mathbf{x}} \|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{T_{s-1}}+\lambda I_d)^{-1}} \leq 2h(t) = 2d_{t,1/t^\alpha}$$
$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x}} \|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{T_{s-1}}+\lambda I_d)^{-1}} \leq 2h(t)\,.$$

Proof. According to the tracking procedure of Degenne et al. [2020b], we know that $T_{s-1}^{\mathbf{x}} \geq \widetilde{W}_{s-1}^{\mathbf{x}} - \log(KA)$. Thus, in combination with the choice of λ we can replace counts by weights

$$\mathbf{\Lambda}_{T_{s-1}} + \lambda I_d \geq \mathbf{\Lambda}_{\widetilde{W}_s^{\mathbf{x}}} - \mathbf{\Lambda}_{\widetilde{w}_s^{\mathbf{x}}} - \log(KA)\mathbf{\Lambda}_{\mathbf{1}_K} + \lambda I_d \geq \mathbf{\Lambda}_{\widetilde{W}_s^{\mathbf{x}}} - (\log(K)+1)\mathbf{\Lambda}_{\mathbf{1}_K} + \lambda I_d \geq \mathbf{\Lambda}_{W_s} + \frac{\lambda}{2} I_d\,,$$

where $\mathbf{1}_K = (1,\dots,1) \in \mathbb{R}^K$. Hence we obtain

$$\|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{T_{s-1}}+\lambda I_d)^{-1}} \le \|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{\widetilde{W}^{\mathbf{x}}_s}+(\lambda/2)I_d)^{-1}},$$

and applying Lemma C.3 leads to

$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathcal{X}} \widetilde{w}^{\mathbf{x}}_s \|\mathbf{x}\|^2_{(\mathbf{\Lambda}_{T_{s-1}}+\lambda I_d)^{-1}} \le d\log\left(1+\frac{tL^2}{d\lambda}\right) \le 2h(t).$$

The exact same proof holds for $w^{\mathbf{x}}_s$ instead of $\widetilde{w}^{\mathbf{x}}_s$ since thanks to the tracking we have also in this case $T^{\mathbf{x}}_{s-1} \ge W^{\mathbf{x}}_{s-1} - \log(K) \ge W^{\mathbf{x}}_{s-1} - \log(KA)$.

□

C.3 Sample Complexity of LinGame

C.3.1 Events

We fix a constant $\alpha > 2$ and define the event

$$\mathcal{E}_t = \left\{\forall s \le t : \frac{1}{2}\left\|\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta}\right\|^2_{\mathbf{\Lambda}_{T_s}+\lambda I_d} \le h(t) \triangleq d_{t,1/t^\alpha}\right\}.$$

This event holds with high probability

Lemma C.5. *For all $t \ge 1$*

$$\mathbb{P}_{\boldsymbol{\theta}}\left[\left(\mathcal{E}^c_t\right)\right] \le \frac{1}{t^{\alpha-1}}.$$

We now prove that we can construct, on this event, upper confidence bounds on the loss given to the learners.

Lemma C.6. *On the event $\mathcal{E}_t$, for all $(\mathbf{x}, i) \in \mathcal{X}\times\mathcal{I}$ and $\boldsymbol{\theta}' \in \neg i$, for all $s \le t$,*

$$\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\mathbf{x}\mathbf{x}^\intercal} \le \min\left(\max_{\pm}\left(\langle\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta}', \mathbf{x}\rangle \pm \sqrt{2h(t)}\,\|\mathbf{x}\|_{(\mathbf{\Lambda}_{T_s}+\lambda I_d)^{-1}}\right)^2, 4M^2\right)$$

Proof. First, note that since $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{M}$, their norms are bounded by M, thus it holds

$$\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\mathbf{x}\mathbf{x}^\intercal} = \langle\boldsymbol{\theta}-\boldsymbol{\theta}', \mathbf{x}\rangle^2 \le \left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2 \left\|\boldsymbol{\theta}'\right\|^2 \le 4M^2L^2.$$

Furthermore on $\mathcal{E}_t$ we have

$$\begin{aligned}\left\|\boldsymbol{\theta}-\boldsymbol{\theta}'\right\|^2_{\mathbf{x}\mathbf{x}^\intercal} = \langle\boldsymbol{\theta}-\boldsymbol{\theta}', \mathbf{x}\rangle^2 &\le \sup_{\left\{\boldsymbol{\theta}' :\, \left\|\widehat{\boldsymbol{\theta}}_s-\boldsymbol{\theta}'\right\|^2_{(\mathbf{\Lambda}_{T_s}+\lambda I_d)^{-1}} \le 2h(t)\right\}} \langle\boldsymbol{\theta}'-\boldsymbol{\theta}', \mathbf{x}\rangle^2 \\ &= \max_{\pm}\left(\langle\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta}', \mathbf{x}\rangle \pm \sqrt{2h(t)}\,\|\mathbf{x}\|_{(\mathbf{\Lambda}_{T_s}+\lambda I_d)^{-1}}\right)^2.\end{aligned}$$

Combining the two inequalities above allows us to conclude. □

We thus define the upper confidence $U^{\mathbf{x},i}_s$ on the coordinate $(\mathbf{x}, i)$ of the loss at time $s \le t$,

$$U^{\mathbf{x},i}_s = \min\left(\max_{\pm}\left(\langle\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}', \mathbf{x}\rangle \pm \sqrt{2h(t)}\,\|\mathbf{x}\|_{(\mathbf{\Lambda}_{T_{s-1}}+\lambda I_d)^{-1}}\right)^2, 4M^2\right). \tag{C.1}$$

The first step of our analysis is to restrict it to the event $\mathcal{E}_t$, as is done by Degenne et al. [2019]; Garivier and Kaufmann [2016].

Lemma C.7. *Let $\mathscr{E}_t$ be an event and $\mathrm{T}_0(\delta) \in \mathbb{N}$ be such that for $t \geq \mathrm{T}_0(\delta)$, $\mathscr{E}_t \subseteq \{\tau_\delta \leq t\}$. Then*

$$\begin{aligned}\mathbb{E}[\tau_\delta] &\leq \mathrm{T}_0(\delta) + \sum_{t=\mathrm{T}_0(\delta)}^{+\infty} \mathbb{P}(\mathscr{E}_{\mathrm{T}}^c) \\ &\leq \mathrm{T}_0(\delta) + \sum_{t=1}^{+\infty} \frac{1}{t^{\alpha-1}}\,.\end{aligned}$$

We need to prove that if $\mathscr{E}_t$ holds, there exists such a time $\mathrm{T}_0(\delta)$.

C.3.2 Analysis under concentration

In this section we assume that the event $\mathscr{E}_t$ holds. And we set $\mathrm{I}^\star = \mathrm{I}^\star(\boldsymbol{\theta})$ and define the following quantities

$$w_t^i = \mathbb{1}_{\{i_t = i\}} w_t \quad \mathrm{W}_t^i = \sum_{s=1}^t w_s^i \quad \mathrm{T}_{t,i} = \sum_{s=1}^t \mathbb{1}\{\widehat{\mathbf{x}}_t = \mathbf{x}, i_t = i\}\,.$$

C.3.3 When $i_s = \mathrm{I}^\star$.

If the algorithm does not stop at stage t we have

$$d_{t,\delta} \geq \frac{1}{2} \max_{i \in \mathscr{I}} \inf_{\boldsymbol{\theta}'_i \in \neg i} \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_i\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} \geq \frac{1}{2} \inf_{\boldsymbol{\theta}' \in \neg \mathrm{I}^\star(\boldsymbol{\theta})} \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\,,$$

Let $\boldsymbol{\theta}'_{i,w}(\boldsymbol{\theta}) \in \arg\min_{\boldsymbol{\theta}' \in \neg i} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_{\boldsymbol{\Lambda}_w}$, such that we have $d_{t,\delta} \geq \frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}$. We first transform that norm into a sum over the rounds of divergences from $\widehat{\boldsymbol{\theta}}_{s-1}$ (instead of $\widehat{\boldsymbol{\theta}}_t$).

Lemma C.8. *If $\mathscr{E}_t$ holds, then an algorithm using C-Tracking ensures that*

$$\begin{aligned}&\frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} + 20\mathrm{A}\left(\sqrt{h(t)g(t)\frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}} + 2h(t)g(t)\right) \\ &\geq \frac{1}{2}\sum_{s=1}^t \|\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\tilde{w}_s}}\,.\end{aligned}$$

Proof. Using the triangular inequality,

$$\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_t\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} + \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} \geq \|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathrm{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\,.$$

We obtain

$$\begin{aligned}\frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} &\geq \frac{1}{2}\left(\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} - \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\right)^2 \\ &\geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} - \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\,.\end{aligned}$$

By definition of the event $\mathscr{E}_t$ we know that $\frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} \leq h(t)$ where $h(t)$ is of order $\mathrm{O}(\log(t))$. Thus we get

$$\frac{1}{2}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} \geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} - \sqrt{4h(t)\frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}}\,,$$

which leads to, using Lemma C.2,

$$d_{t,\delta} + \sqrt{4h(t)d_{t,\delta}} + 4h(t) \geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}. \tag{C.2}$$

We now continue the proof by finding a lower bound for the right hand sum. Using a tracking property from Degenne et al. [2020b] to state that for all a, $-\log(\mathrm{A}) \leq \mathrm{T}^{\mathbf{x}}_t - \mathrm{W}^{\mathbf{x}}_t \leq 1$, we get

$$\begin{aligned}
\frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}} &\geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}} - \frac{\log(\mathrm{A})}{2}\sum_{\mathbf{x}\in\mathscr{X}}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\mathbf{x}\mathbf{x}^\intercal} \\
&\geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}} - \frac{\log(\mathrm{A})}{2}\sqrt{\sum_{\mathbf{x}\in\mathscr{X}}\mathrm{T}^{\mathbf{x}}_t\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\mathbf{x}\mathbf{x}^\intercal}\sum_{a:\mathrm{T}^{\mathbf{x}}_t\geq 1}\frac{1}{\mathrm{T}^{\mathbf{x}}_t}} \\
&= \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}} - \frac{\log(\mathrm{A})}{2}\sqrt{\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}\sum_{a:\mathrm{T}^{\mathbf{x}}_t\geq 1}\frac{1}{\mathrm{T}^{\mathbf{x}}_t}} \\
&\geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}} - \frac{\log(\mathrm{A})}{2}\sqrt{\mathrm{A}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_t}}}.
\end{aligned}$$

Combining the last inequality with (C.2) yields

$$\begin{aligned}
&d_{t,\delta} + \sqrt{4h(t)d_{t,\delta}} + 4h(t) + \frac{\log(\mathrm{A})}{2}\sqrt{2\mathrm{A}}\sqrt{d_{t,\delta} + \sqrt{4h(t)d_{t,\delta}} + 4h(t)} \\
&\geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}}.
\end{aligned}$$

Some simplifications, using the fact that $h(t) \geq 1$, give us

$$d_{t,\delta} + 6\mathrm{A}\left(\sqrt{h(t)d_{t,\delta}} + 2h(t)\right) \geq \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}}. \tag{C.3}$$

We now go from $\boldsymbol{\theta}$ to each $\widehat{\boldsymbol{\theta}}_s$ for $s \leq t$ in the right hand term of the inequality above

$$\begin{aligned}
\frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\mathrm{W}_t}} &= \frac{1}{2}\sum_{s=1}^{t}\|\boldsymbol{\theta} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{w_s}} \\
&\geq \frac{1}{2}\sum_{s=1}^{t}\left(\|\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{w_s}} - 2\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|_{\boldsymbol{\Lambda}_{w_s}}\|\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|_{\boldsymbol{\Lambda}_{w_s}}\right) \\
&\geq \frac{1}{2}\sum_{s=1}^{t}\|\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{w_s}} \\
&\quad - \sqrt{\sum_{s=1}^{t}\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\boldsymbol{\Lambda}_{w_s}}\sum_{s=1}^{t}\|\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{w_s}}}.
\end{aligned} \tag{C.4}$$

We need to upper bound the quantity $\sum_{s=1}^{t} w^{\mathbf{x}}_s\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\intercal}$. By definition of the event $\mathscr{E}_t$ we have

$$\begin{aligned}
\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\intercal} &= \langle\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}, \mathbf{x}\rangle^2 \\
&\leq \left\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}} + \lambda\mathrm{I}_d}\|a\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}} + \lambda\mathrm{I}_d)^{-1}} \\
&\leq 2h(t)\,\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}} + \lambda\mathrm{I}_d)^{-1}}.
\end{aligned}$$

Thus thanks to Lemma C.4 we get

$$\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x}}\|\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\intercal} \le 2h(t)\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} \widetilde{w}_s^{\mathbf{x}}\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda \mathrm{I}_d)^{-1}} \le 2h(t)g(t)\,.$$

Going back to (C.4) in combination with (C.3) and Lemma C.2 leads to

$$d_{t,\delta}+20\mathrm{A}\left(\sqrt{h(t)g(t)d_{t,\delta}}+2h(t)g(t)\right) \ge \frac{1}{2}\sum_{s=1}^{t}\|\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\widetilde{w}_s}}\,. \tag{C.5}$$

□

Lemma C.8 introduces a sum of gains/losses on which our algorithms for pulling and for Nature interact. By definition of the best response $\widetilde{\boldsymbol{\theta}}_s^i = \inf_{\boldsymbol{\theta}'\in\neg i}\|\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}'\|^2_{\widetilde{w}_s^i}$, we have

$$\begin{aligned}
\frac{1}{2}\sum_{s=1}^{t}\|\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}'_{\mathrm{I}^\star,\mathbf{T}_t}(\widehat{\boldsymbol{\theta}}_t)\|^2_{\boldsymbol{\Lambda}_{\widetilde{w}_s}} &\ge \frac{1}{2}\inf_{\boldsymbol{\theta}'\in\neg\mathrm{I}^\star}\sum_s\|\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}'\|^2_{\boldsymbol{\Lambda}_{w_s^{\mathrm{I}^\star}}}\\
&\ge \frac{1}{2}\sum_{s=1}^{t}\inf_{\boldsymbol{\theta}'\in\neg\mathrm{I}^\star}\|\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}'\|^2_{\boldsymbol{\Lambda}_{w_s^{\mathrm{I}^\star}}}\\
&= \frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\|\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal}
\end{aligned} \tag{C.6}$$

Note that our algorithm computes $\widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}$ only when $w_s^{\mathrm{I}^\star}\neq 0$, i.e. only when $i_s = i^*$.

We now introduce the upper confidence bounds

$$\mathrm{U}_s^{\mathbf{x},i} = \min\left(\max_{\pm}\left(\langle\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^i),a\rangle \pm \sqrt{2h(t)}\,\|\mathbf{x}\|_{(\boldsymbol{\Lambda}_{\mathrm{T}_s}+\lambda\mathrm{I}_d)^{-1}}\right)^2, 4\mathrm{L}^2\mathrm{M}^2\right).$$

Lemma C.9. *The upper confidence bounds are such that, under $\mathscr{E}_t$,*

$$\begin{aligned}
\frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\|\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal} \ge \frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star} - h(t)g(t)\\
-2\sqrt{h(t)g(t)}\sqrt{\frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\|\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal}}
\end{aligned}$$

Proof.

$$\begin{aligned}
\mathrm{U}_s^{\mathbf{x},i}-\|\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^i\|^2_{\mathbf{x}\mathbf{x}^\intercal} &\le \max_{\pm}\left(\langle\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}',\mathbf{x}\rangle \pm \sqrt{2h(t)}\,\|\mathbf{x}\|_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda\mathrm{I}_d)^{-1}}\right)^2 - \|\widehat{\boldsymbol{\theta}}_{s-1}-\widetilde{\boldsymbol{\theta}}_s^i\|^2_{\mathbf{x}\mathbf{x}^\intercal}\\
&\le 2h(t)\,\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda\mathrm{I}_d)^{-1}} + 2\sqrt{2h(t)}\,\|\mathbf{x}\|_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda\mathrm{I}_d)^{-1}}\,|\langle\widehat{\boldsymbol{\theta}}_{s-1}-\boldsymbol{\theta}',\mathbf{x}\rangle|\,.
\end{aligned}$$

Hence summing over times and using the Cauchy-Schwarz inequality we obtain

$$\begin{aligned}
&\frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\left(\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star} - \|\widehat{\boldsymbol{\theta}}_{s-1} - (\boldsymbol{\theta}')_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal}\right)\\
&\leq \sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star} h(t)\,\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda \mathrm{I}_d)^{-1}} + w_s^{\mathbf{x},\mathrm{I}^\star}\sqrt{2h(t)}\,\|\mathbf{x}\|_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda \mathrm{I}_d)^{-1}}\,|\langle\widehat{\boldsymbol{\theta}}_{s-1} - \boldsymbol{\theta}', \mathbf{x}\rangle|\\
&\leq h(t)\sum_{\mathbf{x}\in\mathscr{X}}\sum_{s=1}^{t} w_s^{\mathbf{x},\mathrm{I}^\star}\,\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda \mathrm{I}_d)^{-1}}\\
&+ \sqrt{2h(t)}\sqrt{\sum_{\mathbf{x}\in\mathscr{X}}\sum_{s=1}^{t} w_s^{\mathbf{x},\mathrm{I}^\star}\,\|\mathbf{x}\|^2_{(\boldsymbol{\Lambda}_{\mathbf{T}_{s-1}}+\lambda \mathrm{I}_d)^{-1}}}\sqrt{\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\,\|\widehat{\boldsymbol{\theta}}_{s-1} - \widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal}}\\
&\leq h(t)g(t) + 2\sqrt{h(t)g(t)}\sqrt{\frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\,\|\widehat{\boldsymbol{\theta}}_{s-1} - \widetilde{\boldsymbol{\theta}}_s^{\mathrm{I}^\star}\|^2_{\mathbf{x}\mathbf{x}^\intercal}},
\end{aligned}$$

where in the last inequality we used Lemma C.4. □

Thus combining the previous inequality with (C.6) and Lemma C.8 yields, with some simplifications,

$$d_{t,\delta} + 40\mathrm{A}\left(\sqrt{h(t)g(t)d_{t,\delta}} + 2h(t)g(t)\right) \geq \frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star}. \tag{C.7}$$

Now we control the regret of the learner $\mathscr{L}_w^{\mathrm{I}^\star}$, thanks to the bound for `AdaHedge`, see Lemma 4.2, we have

$$\sup_{w\in\Sigma_{\mathrm{K}}} \frac{1}{2}\sum_{s=1}^{t}\mathbb{1}_{\{i_t=\mathrm{I}^\star\}}\sum_{\mathbf{x}\in\mathscr{X}} w^{\mathbf{x}}\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star} - \frac{1}{2}\sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x},\mathrm{I}^\star}\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star} \leq \mathrm{C}'\sqrt{\mathrm{T}_{t,i}} \leq \mathrm{C}'\sqrt{t}.$$

Then using this inequality in combination with the fact that the losses are optimistic we obtain

$$\begin{aligned}
d_{t,\delta} + 40\mathrm{A}\left(\sqrt{h(t)g(t)d_{t,\delta}} + 2h(t)g(t)\right) + \mathrm{C}'\sqrt{t} &\geq \sup_{w\in\Sigma_{\mathrm{K}}} \frac{1}{2}\sum_{s=1}^{t}\mathbb{1}_{\{i_s=\mathrm{I}^\star\}}\sum_{\mathbf{x}\in\mathscr{X}} w^{\mathbf{x}}\mathrm{U}_s^{\mathbf{x},\mathrm{I}^\star}\\
&\geq \sup_{w\in\Sigma_{\mathrm{K}}} \frac{1}{2}\sum_{s=1}^{t}\mathbb{1}_{\{i_s=\mathrm{I}^\star\}}\sum_{\mathbf{x}\in\mathscr{X}} w^{\mathbf{x}}\left\|\boldsymbol{\theta} - (\boldsymbol{\theta}'_s)^{\mathrm{I}^\star}\right\|^2_{\mathbf{x}\mathbf{x}^\intercal}\\
&= \mathrm{T}_{t,\mathrm{I}^\star}\sup_{w\in\Sigma_{\mathrm{K}}} \frac{1}{\mathrm{T}_t^{i^*}}\sum_{s=1}^{t}\mathbb{1}_{\{i_s=\mathrm{I}^\star\}}\frac{1}{2}\left\|\boldsymbol{\theta} - (\boldsymbol{\theta}')_s^{\mathrm{I}^\star}\right\|^2_{\boldsymbol{\Lambda}_w}
\end{aligned}$$

Now remark that $\frac{1}{\mathrm{T}_t^{i^\star}}\sum_{s=1}^{t}\mathbb{1}_{\{i_s=\mathrm{I}^\star\}}\frac{1}{2}\left\|\boldsymbol{\theta} - (\boldsymbol{\theta}')_s^{\mathrm{I}^\star}\right\|^2_{\boldsymbol{\Lambda}_w}$ is the expectation under some distribution of $\frac{1}{2}\left\|\boldsymbol{\theta} - (\boldsymbol{\theta}')_s^{\mathrm{I}^\star}\right\|^2_{\boldsymbol{\Lambda}_w}$.

$$\begin{aligned}
d_{t,\delta} + 40\mathrm{A}\left(\sqrt{h(t)g(t)d_{t,\delta}} + 2h(t)g(t)\right) + \mathrm{C}'\sqrt{t} &\geq \mathrm{T}_{t,\mathrm{I}^\star}\inf_{q\in\mathscr{P}(\neg\mathrm{I}^\star)}\sup_{w\in\Sigma_{\mathrm{K}}}\frac{1}{2}\mathbb{E}_{\boldsymbol{\theta}'\sim q}\left\|\boldsymbol{\theta} - \boldsymbol{\theta}'\right\|^2_{\boldsymbol{\Lambda}_w}\\
&= \mathrm{T}_{t,\mathrm{I}^\star}\mathrm{T}^\star(\boldsymbol{\theta})^{-1},
\end{aligned}$$

where in the last line we use the theorem of Sion, see...Note that the last inequality holds also if $\mathrm{T}_{t,\mathrm{I}^\star} = 0$. It remains to show that $\mathrm{T}_{t,\mathrm{I}^\star} = t - \mathrm{O}(\sqrt{t})$.

C.3.4 When $i_s \neq \mathrm{I}^\star$.

Theorem C.2. *For $i \in \mathscr{I}$ and $t \in \mathbb{N}$, let $\mathrm{T}_{t,i} = \sum_{s=1}^{t} \mathbb{1}\{i_s = i\}$. Then under event $\mathscr{E}_t$,*

$$\mathrm{T}_t^* \geq t - \sqrt{t} - \frac{16}{\Delta_{\min}^2}\left((\mathrm{I}-1)\mathrm{C}'\sqrt{t} + (1+\sqrt{\mathrm{I}-1})^2 h(t)g(t)\right).$$

Since $\boldsymbol{\theta} \in \neg i$ for all $i \neq i^*$, under $\mathscr{E}_t$

$$\begin{aligned} h(t)g(t) \geq \sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x}} \frac{1}{2}\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}} &\geq \sum_{s\leq t, i_s \neq i^*}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{\mathbf{x}} \frac{1}{2}\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}} \\ &\geq \sum_{j\neq i^*} \inf_{\boldsymbol{\theta}'\in\neg j} \sum_{s=1}^{t}\sum_{\mathbf{x}\in\mathscr{X}} w_s^{j,\mathbf{x}} \frac{1}{2}\|\boldsymbol{\theta}' - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}}. \end{aligned}$$

As previously done for i^*, we obtain for all $j \neq i^*$,

$$\inf_{\boldsymbol{\theta}'\in\neg j} \sum_{s=1}^{t}\sum_{\mathbf{x}} w_s^{j,\mathbf{x}} \frac{1}{2}\|\boldsymbol{\theta}' - \widehat{\boldsymbol{\theta}}_{s-1}\|^2_{\mathbf{x}\mathbf{x}^\mathsf{T}} \geq \left(\sqrt{\frac{1}{2}\max_{\mathbf{x}\in\mathscr{X}}\sum_{s=1}^{t}\mathbb{1}_{\{i_s=j\}}\mathrm{U}_s^{\mathbf{x}} - \mathrm{C}'\sqrt{t}} - \sqrt{h(t)g(t)}\right)^2.$$

We show that the sum on the right is proportional to its number of terms $n = t - \mathrm{T}_t^*$, then use that the sum on the left is $\mathcal{O}(\sqrt{t})$. We obtain that $n = \mathcal{O}(\sqrt{t})$.

Lemma C.10. *If the event $\mathscr{E}_t$ holds, then for all $s \leq t$, if $i^*(\widehat{\boldsymbol{\theta}}_s) \neq i^*(\boldsymbol{\theta})$ then there exists an arm $\mathbf{x} \in \mathscr{X}$ such that*

$$\mathrm{U}_s^{\mathbf{x}} \geq 2h(s)\mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}_{\mathrm{T}_s}^{-1}\mathbf{x} \geq \Delta_{\min}^2/4.$$

Proof. If $i^*(\widehat{\boldsymbol{\theta}}_s) \neq i^*(\boldsymbol{\theta})$, then there exists an arm $\mathbf{x} \in \mathscr{X}$ such that

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_s^\mathsf{T}(\mathbf{x} - \mathbf{x}^\star(\boldsymbol{\theta})) \geq 0 &\Rightarrow (\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}(\mathbf{x} - \mathbf{x}^\star(\boldsymbol{\theta})) \geq \Delta_{\min} \\ &\Rightarrow |(\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}\mathbf{x}| + |(\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}\mathbf{x}^\star(\boldsymbol{\theta})| \geq \Delta_{\min} \end{aligned}$$

Hence either

$$|(\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}\mathbf{x}| \geq \frac{\Delta_{\min}}{2},$$

or

$$|(\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}\mathbf{x}^\star(\boldsymbol{\theta})| \geq \frac{\Delta_{\min}}{2}.$$

By consequence, there exists an arm $\mathbf{x}'$ such that

$$|(\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta})^\mathsf{T}\mathbf{x}'| \geq \frac{\Delta_{\min}}{2}.$$

With the Cauchy-Schwarz inequality, there exists $\mathbf{x} \in \mathscr{X}$ such that

$$\frac{\Delta_{\min}}{2} \leq \sqrt{\|\widehat{\boldsymbol{\theta}}_s - \boldsymbol{\theta}\|^2_{\boldsymbol{\Lambda}_{\mathrm{T}_s}} \mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}_{\mathrm{T}_s}^{-1}\mathbf{x}}.$$

Under the event $\mathscr{E}_t$ we obtain

$$\frac{\Delta_{\min}}{2} \leq \sqrt{2h(s)\mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}_{\mathrm{T}_s}^{-1}a} \leq \sqrt{\mathrm{U}_s^{\mathbf{x}}}.$$

□

Lemma C.11. *If we have*

$$2h(t)\mathbf{x}^\top \mathbf{\Lambda}_{\mathbf{T}_t}^{-1}\mathbf{x} \geq x\,,$$

then the following holds

$$\sum_{s\leq t} \mathrm{U}_s^{\mathbf{x}} \geq x \sum_{s\leq t} h(s)/h(t)\,.$$

For $i \neq i^*$, let $t' = \max\{s \leq t, i_s = i\}$. Let a' be an arm such that $2h(t')\mathbf{x}^\top \mathbf{\Lambda}_{\mathrm{N}_{t'}}^{-1} a \geq \Delta_{\min}^2/4$. Then

$$\begin{aligned}
\max_{\mathbf{x}\in\mathscr{X}} \sum_{s\leq t, i_s=i} \mathrm{U}_s^{\mathbf{x}} \geq \sum_{s\leq t, i_s=i} \mathrm{U}_s^{\mathbf{x}'} &\geq \frac{\Delta_{\min}^2}{4} \sum_{s\leq t, i_s=i} h(s)/h(t) \\
&\geq \frac{\Delta_{\min}^2}{4} \sum_{\sqrt{t}\leq s\leq t, i_s=i} h(s)/h(t) \\
&\geq \frac{\Delta_{\min}^2}{4} \sum_{\sqrt{t}\leq s\leq t, i_s=i} \frac{1}{2} \\
&= \frac{\Delta_{\min}^2}{8}(\mathrm{T}_{t,i} - \mathrm{N}^i_{\sqrt{t}})
\end{aligned}$$

We get that $\sum_{i\neq i^*} \max_{\mathbf{x}\in\mathscr{X}} \sum_{s\leq t, i_s=i} \mathrm{U}_s^{\mathbf{x}} \geq \frac{\Delta_{\min}^2}{8}(t - \mathrm{T}_t^* - \sqrt{t})$.

$$\begin{aligned}
h(t)g(t) &\geq \sum_{j\neq i^*} \left(\sqrt{\frac{1}{2}\max_{\mathbf{x}\in\mathscr{A}} \sum_{s=1}^{t} \mathbb{1}_{\{i_s=j\}} \mathrm{U}_s^{\mathbf{x}} - \mathrm{C}'\sqrt{t}} - \sqrt{h(t)g(t)} \right)^2 \\
&\geq \sum_{j\neq i^*} \left(\sqrt{\frac{\Delta_{\min}^2}{16}(\mathrm{T}_{t,j} - \mathrm{N}^j_{\sqrt{t}}) - \mathrm{C}'\sqrt{t}} - \sqrt{h(t)g(t)} \right)^2 \\
&= \frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}^*_{\sqrt{t}}) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t} \\
&\quad - 2\sqrt{h(t)g(t)} \sum_{j\neq i^*} \sqrt{\frac{\Delta_{\min}^2}{16}(\mathrm{T}_{t,j} - \mathrm{N}^j_{\sqrt{t}}) - \mathrm{C}'\sqrt{t}} + (\mathrm{I}-1)h(t)g(t) \\
&\geq \frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}^*_{\sqrt{t}}) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t} \\
&\quad - 2\sqrt{(\mathrm{I}-1)h(t)g(t)} \sqrt{\sum_{j\neq i^*} (\frac{\Delta_{\min}^2}{16}(\mathrm{T}_{t,j} - \mathrm{N}^j_{\sqrt{t}}) - \mathrm{C}'\sqrt{t})} + (\mathrm{I}-1)h(t)g(t) \\
&= \frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}^*_{\sqrt{t}}) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t} \\
&\quad - 2\sqrt{(\mathrm{I}-1)h(t)g(t)} \sqrt{\frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}^*_{\sqrt{t}}) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t})} + (\mathrm{I}-1)h(t)g(t) \\
&= \left(\sqrt{\frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}^*_{\sqrt{t}}) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t})} - \sqrt{(\mathrm{I}-1)h(t)g(t)} \right)^2 .
\end{aligned}$$

$$\frac{\Delta_{\min}^2}{16}(t - \mathrm{T}_t^* - \sqrt{t} + \mathrm{N}_{\sqrt{t}}^*) - (\mathrm{I}-1)\mathrm{C}'\sqrt{t} \leq (1+\sqrt{\mathrm{I}-1})^2 h(t)g(t)$$
$$\Rightarrow t - \mathrm{T}_t^* \leq \sqrt{t} + \frac{16}{\Delta_{\min}^2}\left((\mathrm{I}-1)\mathrm{C}'\sqrt{t} + (1+\sqrt{\mathrm{I}-1})^2 h(t)g(t)\right).$$

C.4 A Fair Comparison of Stopping Rules

We investigate closely the stopping rules employed in existing linear BAI algorithm. We first make a synthesized table that resembles stopping rules and decision rules of all existing algorithms, including ours, in Table C.2. We denote by $\widehat{\mathscr{X}}_n$ the active arm set for elimination-based algorithms, and by $i_{\widehat{\mathscr{X}}_n}$ the only arm left in $\widehat{\mathscr{X}}_n$ when $|\widehat{\mathscr{X}}_n| = 1$.

We show that they are all the same up to the choice of the exploration rate. Note that in Table C.2, we have replaced all the exploration term by $d_{n,\delta}$, and we have also listed the original terms (with their original notation, thus may be in conflict with notation of the current paper). In the following, we always use the same exploration rate $d_{n,\delta}$ for all stopping rules.

Algorithm	Stopping rule	Decision rule
$\mathscr{XY}$-Static	$\exists \mathbf{x} \in \mathscr{X}, \forall \mathbf{x}' \neq \mathbf{x}, \left\|\mathbf{x}-\mathbf{x}'\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}-\mathbf{x}' \rangle$	$\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$
$\mathscr{XY}$-Adaptive	$\lvert\widehat{\mathscr{X}}_n\rvert = 1$, where all arms $\mathbf{x} \in \mathscr{X}$ s.t. $\exists \mathbf{x}' \in \mathscr{X}, \left\|\mathbf{x}-\mathbf{x}'\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}'-\mathbf{x} \rangle$ are discarded	$\mathrm{J}_n = i_{\widehat{\mathscr{X}}_n}$
ALBA	$\lvert\widehat{\mathscr{X}}_n\rvert = 1$, where all arms $\mathbf{x} \in \mathscr{X}$ s.t. $\frac{\left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)-\mathbf{x}\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}}}{\sqrt{1/2d_{n,\delta}}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)-\mathbf{x} \rangle$ are discarded	$\mathrm{J}_n = i_{\widehat{\mathscr{X}}_n}$
RAGE	$\lvert\widehat{\mathscr{X}}_n\rvert = 1$, where all arms $\mathbf{x} \in \mathscr{X}$ s.t. $\exists \mathbf{x}' \in \mathscr{X}, 2^{-t-2} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}'-\mathbf{x} \rangle$ are discarded	$\mathrm{J}_n = i_{\widehat{\mathscr{X}}_n}$
LinGapE	$\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_{j_n} - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) \rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_{j_n}\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}} < 0$ with $j_n = \arg\max_{j \in \mathscr{I}} \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_j - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) \rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_j\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}}$	$\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$
GLGapE	$\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_{j_n} - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) \rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_{j_n}\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}} < 0$ with $j_n = \arg\max_{j \in \mathscr{I}} \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_j - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) \rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_j\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}^{-1}} \sqrt{2d_{n,\delta}}$	$\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$
GLUCB	$\max_{i \in \mathscr{I}} \inf_{\boldsymbol{\theta}' \in \neg i} \frac{\left\|\widehat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\theta}'\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}}^2}{2} \geq d_{n,\delta}$	$\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$
LinGame	$\max_{i \in \mathscr{I}} \inf_{\boldsymbol{\theta}' \in \neg i} \frac{\left\|\widehat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\theta}'\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}}^2}{2} \geq d_{n,\delta}$	$\mathrm{J}_n = i_{n+1}$
LinGame-C	$\max_{i \in \mathscr{I}} \inf_{\boldsymbol{\theta}' \in \neg i} \frac{\left\|\widehat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\theta}'\right\|_{\mathbf{\Lambda}_{\mathrm{T}_n}}^2}{2} \geq d_{n,\delta}$	$\mathrm{J}_n = \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$

Table C.2: Stopping rules for different linear BAI algorithms.

LinGame. We first notice that using the same argument as the proves of Lemma C.1 and Proposition 4.1, the stopping rule of LinGame (and also the one of GLUCB) can be rewritten

as

$$\min_{i \neq I^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)} \frac{\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_i - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle^2}{2\left\|\mathbf{x}_i - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}}} \mathbb{1}\left\{\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)^\intercal \widehat{\boldsymbol{\theta}}_n^\lambda \geq \mathbf{x}_i^\intercal \widehat{\boldsymbol{\theta}}_n^\lambda\right\} > d_{n,\delta}\,.$$

Now we compare it with other stopping rules.

LinGame $\Rightarrow$ $\mathcal{XY}$-Static. If LinGame stops at time t, then for $\mathbf{x} = \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$, we have

$$\forall \mathbf{x}' \neq \mathbf{x}, \left\|\mathbf{x} - \mathbf{x}'\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x} - \mathbf{x}'\rangle\,,$$

and $\mathcal{XY}$-Static stops as well.

$\mathcal{XY}$-Static $\Rightarrow$ $\mathcal{XY}$-Adaptive. Suppose that $\mathcal{XY}$-Static stops at time t under its stopping rule, then

$$\exists \mathbf{x} \in \mathcal{X}, \forall \mathbf{x}' \neq \mathbf{x}, \left\|\mathbf{x} - \mathbf{x}'\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x} - \mathbf{x}'\rangle\,.$$

It is clear that if such $\mathbf{x}$ exists, then it can only be the empirical best arm $\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$. Thus,

$$\forall \mathbf{x}' \neq \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda), \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}'\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}'\rangle\,,$$

and all arms different from $\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$ would be discarded under $\mathcal{XY}$-Adaptive. Furthermore, $\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$ would never be discarded since

$$\forall \mathbf{x}' \neq \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda), \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}' - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle < 0 \leq \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}'\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}}\,,$$

and $\mathcal{XY}$-Adaptive stops.

$\mathcal{XY}$-Adaptive $\Rightarrow$ ALBA Now if $\mathcal{XY}$-Adaptive stops at time t, then all arms but $\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$ are discarded, and

$$\forall\, a \neq \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda), \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}'\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} = \frac{\left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}}}{\sqrt{1/2d_{n,\delta}}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}\rangle\,.$$

Therefore, those arms would also be discarded under ALBA, and ALBA stops.

ALBA $\Rightarrow$ LinGapE and GLGapE. Next, suppose that ALBA stops at time n under its stopping rule, then the only arm left would be $\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)$, and

$$\forall \mathbf{x} \neq \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda), \frac{\left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}}}{\sqrt{1/2d_{n,\delta}}} \leq \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}\rangle\,.$$

And in particular, we get

$$\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_{j_n} - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_{j_n}\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} < 0\,.$$

Thus LinGapE/GLGapE stops under its stopping rule.

LinGapE ⇒ LinGame Finally, we suppose that LinGapE stops at time n, then it comes

$$\begin{aligned} j_n &= \arg\max_{j\in\mathscr{I}} \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_j - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle + \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_j\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}} \\ &= \arg\min_{j\in\mathscr{I}} \langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_j\rangle - \left\|\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda) - \mathbf{x}_j\right\|_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}} \sqrt{2d_{n,\delta}}\,. \end{aligned}$$

By consequence,

$$\begin{aligned} \min_{i\neq \mathrm{I}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)} \frac{\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_i - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle^2}{2\left\|\mathbf{x}_i - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}}} \mathbb{1}\left\{\mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)^\intercal \widehat{\boldsymbol{\theta}}_n^\lambda \geq \mathbf{x}_i^\intercal \widehat{\boldsymbol{\theta}}_n^\lambda\right\} &= \frac{\langle \widehat{\boldsymbol{\theta}}_n^\lambda, \mathbf{x}_{j_n} - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\rangle^2}{2\left\|\mathbf{x}_{j_n} - \mathbf{x}^\star(\widehat{\boldsymbol{\theta}}_n^\lambda)\right\|^2_{\boldsymbol{\Lambda}_{\mathbf{T}_n}^{-1}}} \\ &\geq d_{n,\delta}\,, \end{aligned}$$

and LinGame stops as well.

In conclusion, all the stopping rules are equivalent if we set their exploration term to the same, though formulated in different manners.

Appendix D

Additional Proofs of Chapter 5

D.1 Notation

Table D.1: Table of notation for Chapter 5.

Notation	Meaning
$c_1 \triangleq (\rho/(3\nu))^{1/8}$	constant
$c \triangleq 2\sqrt{1/(1-\rho)}$	constant
$\mathscr{I}_h(n)$	set of nodes created by HCT at level h up to step n
$\mathscr{I}_h^+(n)$	subset of $\mathscr{I}_h(n)$ which contains only the internal nodes
(h_n, i_n)	node selected by the algorithm at each step n
$\mathscr{C}_{h,i} \triangleq \{n = 1, \cdots, \mathrm{N} : (h_n, i_n) = (h, i)\}$	all the time that (h,i) is selected by the algorithm
$\mathscr{C}_{h,i}^+ \triangleq \bigcup_{j \in \{0,\ldots,\mathrm{K}-1\}} \mathscr{C}_{h+1,\mathrm{K}i-j}$	set of child nodes of (h_n, i_n)
$\overline{n}_{h,i} \triangleq \max_{n \in \mathscr{C}_{h,i}} n$	last time (h,i) has been selected
$\widetilde{n}_{h,i} \triangleq \max_{n \in \mathscr{C}_{h,i}^+} n$	last time when one of its children has been selected
$n_{h,i} \triangleq \min\{n : \mathrm{T}_{h,i}(\mathrm{N}) \geq \tau_h(n)\}$	time when (h,i) is expanded
$\widehat{\mathbf{x}}_n$	arm pulled at time n
$\mathrm{T}_n^{\mathbf{x}} = \sum_{t=1}^n \mathbb{1}_{\{\widehat{\mathbf{x}}_t = \mathbf{x}\}}$	number of draws of arm $\mathbf{x}$ at time n
$\mathbf{T}_n = (\mathrm{T}_n^{\mathbf{x}})_{\mathbf{x} \in \mathscr{X}}$	vector of number of draws
$\mathrm{T}_n^{\mathbf{x},i} = \sum_{t=1}^n \mathbb{1}_{\{\widehat{\mathbf{x}}_t = \mathbf{x}, i_t = i\}}$	number of draws of arm $\mathbf{x}$ for a given answer i
λ	regularization parameter
$\widehat{\boldsymbol{\theta}}_n^{\lambda}$	regularized least square estimate

We further introduce some additional notation that are needed for the proof of Theorem 5.2.

- For any t, let $y_n \triangleq (r_n, x_n)$ be a random variable, we define the filtration $\mathscr{F}_n$ as a σ-algebra generated by $(y_1, \ldots, y_n)$.
- Another important notion in HCT is the threshold τ_h on the number of pulls needed before a node at level h can be expanded. The threshold τ_h is chosen such that the two confidence terms in $\mathrm{U}_{h,i}$ are roughly equivalent, that is,

$$\nu\rho^h \simeq c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\tau_h(n)}}.$$

Therefore, we choose

$$\tau_h(n) \triangleq \lceil \frac{c^2 \log(1/\widetilde{\delta}(n^+))}{\nu^2} \rho^{-2h} \rceil .$$

Since n^+ is defined as $2^{\lceil \log(n) \rceil}$, we have $n \leq n^+ \leq 2n$. In addition, log is an increasing function, thus we have

$$\frac{c^2}{\nu^2}\rho^{-2h} \leq \frac{c^2 \log(1/\widetilde{\delta}(n))}{\nu^2}\rho^{-2h} \leq \tau_h(n) \leq \frac{c^2 \log(2/\widetilde{\delta}(n))}{\nu^2}\rho^{-2h}, \tag{D.1}$$

where the first inequality follows from the fact that $0 < \widetilde{\delta}(n) \leq 1/2$.

D.2 Detailed regret analysis for HCT under Assumption 5.2

We begin our analysis by bounding the maximum depth of the trees constructed by HCT.

D.2.1 Maximum depth of the tree (proof of Lemma 5.1)

Lemma 5.1. *The depth of the covering tree produced by HCT after* N *function evaluations satisfies*

$$\mathrm{H}(\mathrm{N}) \leq \mathrm{H}_{\max}(\mathrm{N}) \triangleq \lceil \frac{1}{2(1-\rho)} \log\left(\frac{\mathrm{N}\nu^2}{c^2\rho^2}\right) \rceil .$$

Proof. The deepest tree that can be constructed by HCT is a linear one, where at each level one unique node is expanded. In such case, $|\mathscr{I}_h^+(\mathrm{N})| = 1$ and $|\mathscr{I}_h(\mathrm{N})| = \mathrm{K}$ for all $h < \mathrm{H}(\mathrm{N})$. Therefore, we have

$$\begin{aligned}
\mathrm{N} &= \sum_{h=0}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \mathrm{T}_{h,i}(\mathrm{N}) \\
&\geq \sum_{h=0}^{\mathrm{H}(\mathrm{N})-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \mathrm{T}_{h,i}(\mathrm{N}) \\
&\geq \sum_{h=0}^{\mathrm{H}(\mathrm{N})-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \mathrm{T}_{h,i}(n_{h,i}) \\
&\geq \sum_{h=0}^{\mathrm{H}(\mathrm{N})-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \tau_h(n_{h,i}) && \text{(D.2)} \\
&\geq \sum_{h=0}^{\mathrm{H}(\mathrm{N})-1} \frac{c^2}{\nu^2}\rho^{-2h} && \text{(D.3)} \\
&\geq \frac{(c\rho)^2}{\nu^2}\rho^{-2\mathrm{H}(\mathrm{N})}\mathrm{H}(\mathrm{N}) && \text{(D.4)} \\
&\geq \frac{(c\rho)^2}{\nu^2}\rho^{-2\mathrm{H}(\mathrm{N})} .
\end{aligned}$$

In the previous reasoning, (D.2) is based on the definition of $n_{h,i}$, (D.3) is due to inequality (D.1), and (D.4) holds since $h \leq \mathrm{H}(\mathrm{N}) - 1$.

By solving this expression, we obtain

$$\begin{aligned}
\mathrm{H}(\mathrm{N}) &\leq \frac{1}{2}\log\left(\frac{\mathrm{N}\nu^2}{c^2\rho^2}\right)/\log(1/\rho) \\
&\leq \frac{1}{2(1-\rho)}\log\left(\frac{\mathrm{N}\nu^2}{c^2\rho^2}\right) \qquad \text{(D.5)} \\
&\leq \lceil \frac{1}{2(1-\rho)}\log\left(\frac{\mathrm{N}\nu^2}{c^2\rho^2}\right)\rceil \\
&\triangleq \mathrm{H}_{\max}(\mathrm{N})\,.
\end{aligned}$$

where (D.5) follows from $\log(1/\rho) \geq 1-\rho$. □

D.2.2 High-probability event

In Section 5.4.2, we described the favorable event ξ_n. We now define it precisely. We first define a set $\mathscr{L}_n$ that contains all possible nodes in trees of maximum depth $\mathrm{H}_{\max}(n)$,

$$\mathscr{L}_n \triangleq \bigcup_{\mathscr{T}:\mathrm{depth}(\mathscr{T})\leq \mathrm{H}_{\max}(n)} \mathrm{Nodes}(\mathscr{T})$$

and we recall the definition of the favorable event

$$\xi_n \triangleq \left\{ \forall (h,i) \in \mathscr{L}_n, |\widehat{\mu}_{h,i}(n) - \mu_{h,i}| \leq c\sqrt{\frac{\log(1/\widetilde{\delta}(n))}{\mathrm{T}_{h,i}(n)}} \right\}.$$

Next, we prove that our favorable event holds with high probability.

Lemma D.1. *With c_1 and c defined in Section 5.2, for any fixed round n,*

$$\mathbb{P}\left[\xi_n\right] \geq 1 - \frac{4\delta}{3n^6}\,.$$

Proof. Let $\widehat{\mu}_{h,i,s}$ be the empirical mean reward of the first s noisy evaluations of f in $x_{h,i}$, we upper-bound the probability of the complementary event ξ_n^c as

$$\begin{aligned}
\mathbb{P}\left[\xi_n^c\right] &\leq \sum_{(h,i)\in\mathscr{L}_n}\sum_{s=1}^{n}\mathbb{P}\left[|\widehat{\mu}_{h,i,s} - \mu_{h,i}| \geq c\sqrt{\frac{\log(1/\widetilde{\delta}(n))}{s}}\right] \qquad \text{(D.6)} \\
&\leq \sum_{(h,i)\in\mathscr{L}_n}\sum_{s=1}^{n} 2\exp\left(-2c^2\log(1/\widetilde{\delta}(n))\right) \qquad \text{(D.7)} \\
&= 2\exp\left(-2c^2\log(1/\widetilde{\delta}(n))\right) n|\mathscr{L}_n| \\
&= 2(\widetilde{\delta}(n))^{2c^2} n|\mathscr{L}_n| \\
&\leq 2(\widetilde{\delta}(n))^{2c^2} n 2^{\mathrm{H}_{max}(n)+1} \\
&= 2(\widetilde{\delta}(n))^{2c^2} n 2^{\lceil \frac{1}{2(1-\rho)}\log\left(\frac{n\nu^2}{c^2\rho^2}\right)\rceil+1} \qquad \text{(D.8)} \\
&\leq 8n(\widetilde{\delta}(n))^{2c^2}\left(\frac{t\nu^2}{c^2\rho^2}\right)^{\frac{1}{2(1-\rho)}}
\end{aligned}$$

$$
\begin{aligned}
&\leq 8n\left(\frac{\delta}{n}(\rho/(3\nu))^{1/8}\right)^{\frac{8}{1-\rho}}\left(\frac{n\nu^2(1-\rho)}{4\rho^2}\right)^{\frac{1}{2(1-\rho)}} && \text{(D.9)}\\
&= 8n\left(\frac{\delta}{n}\right)^{\frac{8}{1-\rho}}\left(\frac{\rho}{3\nu}\right)^{\frac{1}{1-\rho}} n^{\frac{1}{2(1-\rho)}}\left(\frac{\nu\sqrt{1-\rho}}{2\rho}\right)^{\frac{1}{1-\rho}}\\
&\leq \frac{4}{3}\delta n^{\frac{-2\rho-13}{2(1-\rho)}}\\
&\leq \frac{4\delta}{3n^6}.
\end{aligned}
$$

Here, (D.6) is derived using a union bound, and (D.7) is due to the Hoeffding inequality (see Appendix A.2 for details). (D.8) is a result of Lemma 5.1 and finally (D.9) is obtained by plugging in values of c and c_1. □

D.2.3 Failing confidence bound

We decompose the regret of HCT into two terms depending on whether ξ_n holds. Let us define $\Delta_n \triangleq f^\star - r_n$. Then, we decompose the regret as

$$
\mathrm{R}_\mathrm{N}^{\mathtt{HCT}} = \sum_{n=1}^{\mathrm{N}} \Delta_n = \sum_{n=1}^{\mathrm{N}} \Delta_n \mathbf{1}_{\xi_n} + \sum_{n=1}^{\mathrm{N}} \Delta_n \mathbf{1}_{\xi_n^c} = \mathrm{R}_\mathrm{N}^{\xi} + \mathrm{R}_\mathrm{N}^{\xi^c}.
$$

The failing confidence term $\mathrm{R}_\mathrm{N}^{\xi^c}$ is bounded by the following lemma.

Lemma D.2. *With c_1 and c defined in Section D.1, when the favorable event does not hold, the regret of HCT is with probability $1-\delta/(5\mathrm{N}^2)$ bounded as*

$$
\mathrm{R}_\mathrm{N}^{\xi^c} \leq \sqrt{\mathrm{N}}.
$$

Proof. We split the term into rounds from 1 to $\sqrt{\mathrm{N}}$ and the rest,

$$
\mathrm{R}_\mathrm{N}^{\xi^c} = \sum_{n=1}^{\mathrm{N}} \Delta_n \mathbf{1}_{\xi_n^c} = \sum_{n=1}^{\sqrt{\mathrm{N}}} \Delta_n \mathbf{1}_{\xi_n^c} + \sum_{n=\sqrt{\mathrm{N}}+1}^{\mathrm{N}} \Delta_n \mathbf{1}_{\xi_n^c}.
$$

The first term can be bounded trivially by $\sqrt{\mathrm{N}}$ since $|\Delta_n| \leq 1$. Next, we show that the probability that the second term is non zero is bounded by $\delta/(5\mathrm{N}^2)$.

$$
\begin{aligned}
\mathbb{P}\left[\sum_{n=\sqrt{\mathrm{N}}+1}^{\mathrm{N}} \Delta_n \mathbf{1}_{\xi_n^c} > 0\right] &= \mathbb{P}\left[\bigcup_{n=\sqrt{\mathrm{N}}+1}^{\mathrm{N}} \xi_n^c\right]\\
&\leq \sum_{n=\sqrt{\mathrm{N}}+1}^{\mathrm{N}} \mathbb{P}\left[\xi_n^c\right] && \text{(D.10)}\\
&\leq \sum_{n=\sqrt{\mathrm{N}}+1}^{\mathrm{N}} \frac{\delta}{n^6} && \text{(D.11)}\\
&\leq \int_{\sqrt{\mathrm{N}}}^{\infty} \frac{\delta}{n^6}\,\mathrm{d}n\\
&= \frac{\delta}{5\mathrm{N}^{5/2}}
\end{aligned}
$$

$$\leq \frac{\delta}{5N^2}.$$

In the previous reasoning, (D.10) is achieved again by a simple union bound, and (D.11) can be obtained by Lemma D.1 □

D.2.4 Proof of Theorem 5.2

Theorem 5.2. *Assume that function f satisfies Assumption 5.2. Then, setting $\delta \triangleq 1/N$, the cumulative regret of HCT(ν,ρ) after N function evaluations is upper bounded as*

$$\mathbb{E}[R_N^{HCT(\nu,\rho)}] \leq \alpha C (\log N)^{1/(d(\nu,C,\rho)+2)} N^{(d(\nu,C,\rho)+1)/(d(\nu,C,\rho)+2)},$$

where α is a numerical constant and C is the constant associated to $d(\nu,C,\rho)$.

For the sake of simplicity, we denote $d(\nu,C,\rho)$ as d in the rest of this section. We study the regret under events $\{\xi_n\}_n$ and prove that

$$R_N^{\mathtt{HCT}(\nu,\rho)} \leq 2\sqrt{2N\log(\frac{4N^2}{\delta})} + 3\left(\frac{2^{3d+7}\nu^d KC\rho^d}{(1-\rho)^2}\right)^{\frac{1}{d+2}} \left(\log\left(\frac{2N}{\delta}\sqrt[8]{\frac{3\nu}{\rho}}\right)\right)^{\frac{1}{d+2}} N^{\frac{d+1}{d+2}}$$

holds with probability $1-\delta$. We decompose the proof into 3 steps.

Step 1: Decomposition of the regret. We start by further decomposing the instantaneous regret into two terms,

$$\Delta_n = f^\star - r_n = f^\star - f(x_{h_n,i_n}) + f(x_{h_n,i_n}) - r_n = \Delta_{h_n,i_n} + \widehat{\Delta}_n.$$

The regret of HCT when confidence intervals hold can thus be rewritten as

$$R_N^\xi = \sum_{n=1}^N \Delta_{h_n,i_n}\mathbf{1}_{\xi_n} + \sum_{n=1}^N \widehat{\Delta}_n\mathbf{1}_{\xi_n} \leq \sum_{n=1}^N \Delta_{h_n,i_n}\mathbf{1}_{\xi_n} + \sum_{n=1}^N \widehat{\Delta}_n = \widetilde{R}_N^\xi + \widehat{R}_N^\xi. \tag{D.12}$$

We notice that the sequence $\{\widehat{\Delta}_n\}_{n=1}^N$ is a bounded martingale difference sequence since $\mathbb{E}\left[\widehat{\Delta}_n|\mathscr{F}_{n-1}\right] = 0$ and $|\widehat{\Delta}_n| \leq 1$. Thus, we apply the Azuma's inequality on this sequence and obtain

$$\widehat{R}_N^\xi \leq \sqrt{2N\log\left(\frac{4N^2}{\delta}\right)} \tag{D.13}$$

with probability $1-\delta/(4N^2)$.

Step 2: Preliminary bound on the regret of selected nodes and their parents. Now we proceed with the bound of the first term $\widetilde{R}_N^\xi$. Recall that P_n is the optimistic path traversed by HCT at round t. Let $(h',i') \in P_n$ and (h'',i'') be the node which immediately follows (h',i') in P_n. By definition of B-values and U-values, we have

$$B_{h',i'}(n) \leq \max_{j\in\{0,\dots,K-1\}} \{B_{h'+1,Ki'-j}(n)\} = B_{h'',i''}(n), \tag{D.14}$$

where the last equality follows from the fact that the subroutine `OptTraverse` selects the node with the largest B-value. By iterating the previous inequality along the path P_n until the selected node (h_n, i_n) and its parent (h_n^p, i_n^p), we obtain

$$\forall (h', i') \in \mathrm{P}_n, \mathrm{B}_{h',i'}(n) \leq \mathrm{B}_{h_n,i_n}(n) \leq \mathrm{U}_{h_n,i_n}(n),$$

$$\forall (h', i') \in \mathrm{P}_n \setminus \{(h_n, i_n)\}, \mathrm{B}_{h',i'}(n) \leq \mathrm{B}_{h_n^p,i_n^p}(n) \leq \mathrm{U}_{h_n^p,i_n^p}(n).$$

Since the root, which is an optimal node, is in P_n, there exists at least one optimal node $(h^\star, i^\star)$ in path P_n. As a result, we have

$$\mathrm{B}_{h^\star,i^\star}(n) \leq \mathrm{U}_{h_n,i_n}(n), \tag{D.15}$$

$$\mathrm{B}_{h^\star,i^\star}(n) \leq \mathrm{U}_{h_n^p,i_n^p}(n). \tag{D.16}$$

We now expand (D.15) on both sides under ξ_n. First, we have

$$\begin{aligned}\mathrm{U}_{h_n,i_n}(n) &\triangleq \widehat{\mu}_{h_n,i_n}(n) + \nu\rho^{h_n} + c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h_n,i_n}(n)}} \\ &\leq f(x_{h_n,i_n}) + \nu\rho^{h_n} + 2c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h_n,i_n}(n)}}\end{aligned} \tag{D.17}$$

and the same holds for the parent of the selected node,

$$\mathrm{U}_{h_n^p,i_n^p}(n) \leq f(x_{h_n^p,i_n^p}) + \nu\rho^{h_n^p} + 2c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h_n^p,i_n^p}(n)}}.$$

By Lemma 5.2, we know that $\mathrm{U}_{h^\star,i^\star}(n)$ is a valid upper bound on $f^\star$. If an optimal node $(h^\star, i^\star)$ is a leaf, then $\mathrm{B}_{h^\star,i^\star}(n) = \mathrm{U}_{h^\star,i^\star}(n)$ is also a valid upper bound on $f^\star$. Otherwise, there always exists a leaf which contains the maximum for which $(h^\star, i^\star)$ is its ancestor. Now, if we propagate the bound backward from this leaf to $(h^\star, i^\star)$ through (D.14), we have that $\mathrm{B}_{h^\star,i^\star}(n)$ is still a valid upper bound on $f^\star$. Thus for any optimal node $(h^\star, i^\star)$, at round n under ξ_n, we have

$$\mathrm{B}_{h^\star,i^\star}(n) \geq f^\star. \tag{D.18}$$

We combine (D.18) with (D.15) and (D.17) to obtain

$$\Delta_{h_n,i_n} \triangleq f^\star - f(x_{h_n,i_n}) \leq \nu\rho^{h_n} + 2c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h_n,i_n}(n)}}.$$

The same result holds for its parent,

$$\Delta_{h_n^p,i_n^p} \triangleq f^\star - f(x_{h_n^p,i_n^p}) \leq \nu\rho^{h_n^p} + 2c\sqrt{\frac{\log(1/\widetilde{\delta}(n^+))}{\mathrm{T}_{h_n^p,i_n^p}(n)}}.$$

We now refine the two above expressions. The subroutine `OptTraverse` tells us that `HCT` only selects a node when $\mathrm{T}_{h,i}(\mathrm{N}) < \tau_h(n)$. Therefore, by definition of $\tau_{h_n}(n)$, we have

$$\Delta_{h_n,i_n} \leq 3c\sqrt{\frac{\log(2/\widetilde{\delta}(n))}{\mathrm{T}_{h_n,i_n}(n)}}. \tag{D.19}$$

On the other hand, `OptTraverse` tells us that $\mathrm{T}_{h_n^p,i_n^p}(n) \geq \tau_{h_n^p}(n)$, thus

$$\Delta_{h_n^p,i_n^p} \leq 3\nu\rho^{h_n^p},$$

which means that every selected node has a parent which is $(3\nu\rho^{h_n-1})$-optimal.

Step 3: Bound on the cumulative regret. We return to term $\widetilde{\mathrm{R}}_{\mathrm{N}}^{\xi}$ and split it into different depths. Let $1 \le \overline{\mathrm{H}} \le \mathrm{H}(\mathrm{N})$ be a constant that we fix later. We have

$$
\begin{aligned}
\widetilde{\mathrm{R}}_{\mathrm{N}}^{\xi} &\triangleq \sum_{n=1}^{\mathrm{N}} \Delta_{h_n,i_n} \mathbf{1}_{\xi_n} \\
&\le \sum_{h=0}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{n=1}^{\mathrm{N}} \Delta_{h,i} \mathbf{1}_{(h_n,i_n)=(h,i)} \mathbf{1}_{\xi_n} \\
&\le \sum_{h=0}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{n=1}^{\mathrm{N}} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(n))}{\mathrm{T}_{h,i}(n)}} \mathbf{1}_{(h_n,i_n)=(h,i)} \qquad \text{(D.20)} \\
&= \sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{n=1}^{\mathrm{N}} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(n))}{\mathrm{T}_{h,i}(n)}} \mathbf{1}_{(h_n,i_n)=(h,i)} \\
&+ \sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{n=1}^{\mathrm{N}} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(n))}{\mathrm{T}_{h,i}(n)}} \mathbf{1}_{(h_n,i_n)=(h,i)} \\
&\le \sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{s=1}^{\tau_h(\overline{n}_{h,i})} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(\overline{n}_{h,i}))}{s}} + \sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sum_{s=1}^{\mathrm{T}_{h,i}(\mathrm{N})} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(\overline{n}_{h,i}))}{s}} \\
&\le \sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \int_1^{\tau_h(\overline{n}_{h,i})} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(\overline{n}_{h,i}))}{s}} \, \mathrm{d}s \\
&+ \sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \int_1^{\mathrm{T}_{h,i}(\mathrm{N})} 3c \sqrt{\frac{\log(2/\widetilde{\delta}(\overline{n}_{h,i}))}{s}} \, \mathrm{d}s \\
&\le \sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} 6c \sqrt{\tau_h(\overline{n}_{h,i}) \log(2/\widetilde{\delta}(\overline{n}_{h,i}))} + \sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} 6c \sqrt{\mathrm{T}_{h,i}(\mathrm{N}) \log(2/\widetilde{\delta}(\overline{n}_{h,i}))} \\
&= 6c \left(\underbrace{\sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sqrt{\tau_h(\overline{n}_{h,i}) \log(2/\widetilde{\delta}(\overline{n}_{h,i}))}}_{\text{(a)}} + \underbrace{\sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H}(\mathrm{N})} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sqrt{\mathrm{T}_{h,i}(\mathrm{N}) \log(2/\widetilde{\delta}(\overline{n}_{h,i}))}}_{\text{(b)}} \right).
\end{aligned}
$$

In particular, (D.20) is due to inequality (D.19).

We bound separately the terms (a) and (b). Since $\overline{n}_{h,i} \le \mathrm{N}$, we have

$$
\begin{aligned}
\text{(a)} &\le \sum_{h=0}^{\overline{\mathrm{H}}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \sqrt{\tau_h(\mathrm{N}) \log(2/\widetilde{\delta}(\mathrm{N}))} \\
&\le \sum_{h=0}^{\overline{\mathrm{H}}} |\mathscr{I}_h(\mathrm{N})| \sqrt{\tau_h(\mathrm{N}) \log(2/\widetilde{\delta}(\mathrm{N}))}.
\end{aligned}
$$

Notice that the covering tree is K-ary and therefore $|\mathscr{I}_h(\mathrm{N})| \le \mathrm{K}|\mathscr{I}_{h-1}(\mathrm{N})|$. Recall that `HCT` only selects a node (h_n, i_n) when its parent is $3\nu\rho^{h_n-1}$-optimal. Therefore, by definition of the near-optimality dimension,

$$
|\mathscr{I}_h(\mathrm{N})| \le |\mathrm{K}.\mathscr{I}_{h-1}(\mathrm{N})| \le \mathrm{K}C\rho^{-d(h-1)},
$$

where d is the near-optimality dimension. As a result, for term (a), we obtain that

$$
\text{(a)} \le \sum_{h=0}^{\overline{\mathrm{H}}} \mathrm{K}C\rho^{-d(h-1)} \sqrt{\tau_h(\mathrm{N}) \log(2/\widetilde{\delta}(\mathrm{N}))}
$$

$$= \sum_{h=0}^{\overline{\mathrm{H}}} \mathrm{KC}\rho^{-d(h-1)} \sqrt{\frac{c^2 \log(2/\widetilde{\delta}(\mathrm{N}))}{\nu^2} \rho^{-2h} \log(2/\widetilde{\delta}(\mathrm{N}))} \tag{D.21}$$

$$= \mathrm{KC}\rho^{d} \frac{c \log(2/\widetilde{\delta}(\mathrm{N}))}{\nu} \sum_{h=0}^{\overline{\mathrm{H}}} \rho^{-h(d+1)}.$$

where (D.21) is deduced from inequality (D.1). Consequently, we bound (a) as

$$(\mathrm{a}) \leq \mathrm{KC}\rho^{d} \frac{c \log\left(2/\widetilde{\delta}(\mathrm{N})\right)}{\nu} \frac{\rho^{-\overline{\mathrm{H}}(d+1)}}{1-\rho}. \tag{D.22}$$

We proceed to bound the second term (b). By the Cauchy-Schwarz inequality,

$$\begin{aligned}(\mathrm{b}) &\leq \sqrt{\sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H(N)}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \log\left(2/\widetilde{\delta}\left(\overline{n}_{h,i}\right)\right)} \sqrt{\sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H(N)}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \mathrm{T}_{h,i}(\mathrm{N})} \\ &\leq \sqrt{n \sum_{h=\overline{\mathrm{H}}+1}^{\mathrm{H(N)}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \log\left(2/\widetilde{\delta}\left(\overline{n}_{h,i}\right)\right)},\end{aligned}$$

where we trivially bound the second square-root factor by the total number of pulls. Now consider the first square-root factor. Recall that the `HCT` algorithm only selects a node when $\mathrm{T}_{h,i}(n) \geq \tau_h(n)$ for its parent. We therefore have $\mathrm{T}_{h,i}(\widetilde{n}_{h,i}) \geq \tau_h(\widetilde{n}_{h,i})$ and the following sequence of inequalities,

$$\begin{aligned}\mathrm{N} &= \sum_{h=0}^{\mathrm{H(N)}} \sum_{i \in \mathscr{I}_h(\mathrm{N})} \mathrm{T}_{h,i}(\mathrm{N}) \\ &\geq \sum_{h=0}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \mathrm{T}_{h,i}(\mathrm{N}) \\ &\geq \sum_{h=0}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \mathrm{T}_{h,i}(\widetilde{n}_{h,i}) \qquad \text{(D.23)} \\ &\geq \sum_{h=0}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \tau_h(\widetilde{n}_{h,i}) \\ &\geq \sum_{h=\overline{\mathrm{H}}}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \tau_h(\widetilde{n}_{h,i}) \\ &= \sum_{h=\overline{\mathrm{H}}}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \frac{c^2 \log(1/\widetilde{\delta}(\widetilde{n}_{h,i}^+))}{\nu^2} \rho^{-2h} \\ &\geq \sum_{h=\overline{\mathrm{H}}}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \frac{c^2 \log(1/\widetilde{\delta}(\widetilde{n}_{h,i}^+))}{\nu^2} \rho^{-2\overline{\mathrm{H}}} \\ &= \frac{c^2 \rho^{-2\overline{\mathrm{H}}}}{\nu^2} \sum_{h=\overline{\mathrm{H}}}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \log(1/\widetilde{\delta}(\widetilde{n}_{h,i}^+))) \\ &= \frac{c^2 \rho^{-2\overline{\mathrm{H}}}}{\nu^2} \sum_{h=\overline{\mathrm{H}}}^{\mathrm{H(N)}-1} \sum_{i \in \mathscr{I}_h^+(\mathrm{N})} \log(1/\widetilde{\delta}(\left[\max(\overline{n}_{h+1,2i-1}, \overline{n}_{h+1,2i})\right]^+)) \qquad \text{(D.24)}\end{aligned}$$

$$= \frac{c^2\rho^{-2\overline{H}}}{\nu^2} \sum_{h=\overline{H}}^{H(N)-1} \sum_{i\in\mathscr{I}_h^+(N)} \log(1/\widetilde{\delta}(\max(\overline{n}^+_{h+1,2i-1}, \overline{n}^+_{h+1,2i}))) \tag{D.25}$$

$$= \frac{c^2\rho^{-2\overline{H}}}{\nu^2} \sum_{h=\overline{H}}^{H(N)-1} \sum_{i\in\mathscr{I}_h^+(N)} \max(\log(1/\widetilde{\delta}(\overline{n}^+_{h+1,2i-1})), \log(1/\widetilde{\delta}(\overline{n}^+_{h+1,2i})))$$

$$\geq \frac{c^2\rho^{-2\overline{H}}}{\nu^2} \sum_{h=\overline{H}}^{H(N)-1} \sum_{i\in\mathscr{I}_h^+(N)} \frac{\log(1/\widetilde{\delta}(\overline{n}^+_{h+1,2i-1})) + \log(1/\widetilde{\delta}(\overline{n}^+_{h+1,2i}))}{2}$$

$$= \frac{c^2\rho^{-2\overline{H}}}{\nu^2} \sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_{h-1}^+(n)} \frac{\log(1/\widetilde{\delta}(\overline{n}^+_{h,2i-1})) + \log(1/\widetilde{\delta}(\overline{n}^+_{h,2i}))}{2} \tag{D.26}$$

$$= \frac{c^2\rho^{-2\overline{H}}}{2\nu^2} \sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_h^+(N)} \log(1/\widetilde{\delta}(\overline{n}^+_{h,i}))\,.$$

Note that in (D.23), $\widetilde{n}_{h,i}$ is well defined for $i \in \mathscr{I}_h^+(N)$. In the above derivation, (D.24) holds since $\widetilde{n}_{h,i} = \max(\overline{n}_{h+1,2i-1}, \overline{n}_{h+1,2i})$, and (D.25) holds since $\forall n_1, n_2, [\max(n_1, n_2)]^+ = \max(n_1^+, n_2^+)$. Besides, (D.26) is simply due to a change of variables. Finally, the last equality relies on the fact that for any $h > 0$, $\mathscr{I}_h^+(N)$ covers all the internal nodes at level h and therefore its children cover $\mathscr{I}_{h+1}(N)$. We thus obtain

$$\sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_h^+(N)} \log(1/\widetilde{\delta}(\overline{n}^+_{h,i})) \leq \frac{2\nu^2\rho^{2\overline{H}}N}{c^2}. \tag{D.27}$$

On the other hand, we have

$$\begin{aligned}
(b) &\leq \sqrt{n \sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_h(N)} \log(2/\widetilde{\delta}(\overline{n}_{h,i}))} \\
&\leq \sqrt{n \sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_h(N)} 2\log(1/\widetilde{\delta}(\overline{n}_{h,i}))} \\
&\leq \sqrt{n \sum_{h=\overline{H}+1}^{H(N)} \sum_{i\in\mathscr{I}_h(N)} 2\log(1/\widetilde{\delta}(\overline{n}^+_{h,i}))}\,.
\end{aligned}$$

The last seep holds in the above since $\overline{n}_{h,i} \leq \overline{n}^+_{h,i}$. By plugging (D.27) into the expression above, we get

$$(b) \leq \frac{2\nu\rho^{\overline{H}}N}{c}. \tag{D.28}$$

Now if we combine (D.28) with (D.22), we bound $\widetilde{R}^{\xi}_N$ as

$$\widetilde{R}^{\xi}_N \leq 6\nu \left[KC\rho^d \frac{c^2 \log(2/\widetilde{\delta}(N))}{\nu^2} \frac{\rho^{-\overline{H}(d+1)}}{1-\rho} + 2\rho^{\overline{H}}N \right]. \tag{D.29}$$

We choose $\overline{H}$ to minimize the above bound by equalizing the two terms in the sum and obtain

$$\rho^{\overline{H}} = \left(\frac{KC\rho^d c^2 \log(2/\widetilde{\delta}(N))}{2n(1-\rho)\nu^2} \right)^{\frac{1}{d+2}}, \tag{D.30}$$

which after being plugged into (D.29) gives

$$\widetilde{R}_N^{\xi} \le 24\nu \left(\frac{KC\rho^d c^2 \log(2/\widetilde{\delta}(N))}{2(1-\rho)\nu^2} \right)^{\frac{1}{d+2}} N^{\frac{d+1}{d+2}}. \tag{D.31}$$

Finally, combining (D.31), (D.13), and Lemma D.2, we obtain

$$\begin{aligned} R_N^{\mathtt{HCT}} &\le \sqrt{N} + \sqrt{2N\log(\frac{4N^2}{\delta})} + 24\nu \left(\frac{2KC\rho^d}{(1-\rho)^2\nu^2} \right)^{\frac{1}{d+2}} \left(\log\left(\frac{2N}{\delta} \sqrt[8]{\frac{3\nu}{\rho}} \right) \right)^{\frac{1}{d+2}} N^{\frac{d+1}{d+2}} \\ &= \sqrt{N} + \sqrt{2N\log(\frac{4N^2}{\delta})} + 3 \left(\frac{2^{3d+7}\nu^d KC\rho^d}{(1-\rho)^2} \right)^{\frac{1}{d+2}} \left(\log\left(\frac{2N}{\delta} \sqrt[8]{\frac{3\nu}{\rho}} \right) \right)^{\frac{1}{d+2}} N^{\frac{d+1}{d+2}} \\ &\le 2\sqrt{2N\log(\frac{4N^2}{\delta})} + 3 \left(\frac{2^{3d+7}\nu^d KC\rho^d}{(1-\rho)^2} \right)^{\frac{1}{d+2}} \left(\log\left(\frac{2N}{\delta} \sqrt[8]{\frac{3\nu}{\rho}} \right) \right)^{\frac{1}{d+2}} N^{\frac{d+1}{d+2}} \end{aligned}$$

with probability $1-\delta$.

D.3 General analysis of POO

We prove Proposition 5.1 in this section. The analysis of POO originally proposed by Grill et al. [2015] consists in two main parts, that can be adapted to any base algorithm satisfying assumption (5.6) on its cumulative regret. In the following, we assume that $\nu^\star \le \nu_{\max}$ and $\rho^\star \le \rho_{\max}$.

The first part of the analysis consists in proving that there exists a parameter $\overline{\rho}$ such that $(\nu_{\max}, \overline{\rho}) \in \mathcal{G}$ and the instance $\mathscr{A}(\nu_{\max}, \overline{\rho})$ has its simple regret bounded in terms of the *true parameters* $(\nu^\star, \rho^\star)$. One important ingredient is the following lemma, which upper bounds the difference between the near-optimality dimension $d(\nu_{\max}, C, \rho)$ and $d(\nu^\star, C^\star, \rho^\star)$ for $\rho > \rho^\star$.

Lemma D.3 (Appendix B.1 of Grill et al. 2015)**.** *Under Assumption 5.2, for any choice of* $\rho^\star$ *and* ρ *s.t.* $0 < \rho^\star < \rho < 1$*, we have*

$$d(\nu_{\max}, C, \rho) - d(\nu^\star, C^\star, \rho^\star) \le \log K \left(\frac{1}{\log(1/\rho)} - \frac{1}{\log(1/\rho^\star)} \right).$$

Lemma D.3 endorses the choice of grid $\mathcal{G} = \{(\nu_{\max}, \rho_{\max}^{2M/(2i+1)})_i\}$, which ensures that

$$\overline{\rho} \triangleq \operatorname*{arg\,min}_{\rho_i \ge \rho^\star} \left[d(\nu_{\max}, C_i, \rho_i) - d(\nu^\star, C^\star, \rho^\star) \right].$$

satisfies $d(\nu_{\max}, \overline{C}, \overline{\rho}) - d(\nu^\star, C^\star, \rho^\star) \le D_{\max}/N$, where $\overline{C}$ is associated to $\overline{\rho}$. A close examination of Appendix B.2 and B.3 of Grill et al. [2015] shows that under the assumption

$$\log \mathbb{E}\left[S_n^{\mathscr{A}(\nu_{\max}, \overline{\rho})} \right] \le \log \alpha + \frac{\log C(\nu_{\max}, \overline{\rho})}{d(\nu_{\max}, \overline{C}, \overline{\rho}) + 2} - \frac{\log(n/\log n)}{d(\nu_{\max}, \overline{C}, \overline{\rho}) + 2}, \tag{D.32}$$

the simple regret of $\mathscr{A}(\nu_{\max}, \overline{\rho})$ can also be related to $(\nu^\star, C^\star, \rho^\star)$: for some constant α',

$$\mathbb{E}\left[S_n^{\mathscr{A}(\nu_{\max}, \overline{\rho})} \right] \le \alpha' D_{\max} (\nu_{\max}/\nu^\star)^{D_{\max}} \left((\log^2 n)/n \right)^{1/(d(\nu^\star, C^\star, \rho^\star)+2)} \tag{D.33}$$

under assumption described by (5.6) on the cumulative regret of the base algorithms. Note that (D.32) holds as the recommendation rule ensures that $\mathbb{E}[S_n] = \mathbb{E}[R_n]/n$.

The second part of the analysis controls the simple regret of $\texttt{POO}(\mathscr{A})$ by showing that the error made when choosing $s^\star \neq (\nu_{\max}, \overline{\rho})$ is negligible. We highlight that for this part, having cumulative regret guarantees is crucial. Denoting by $(x_{i,j})_{1\leq i\leq N/M}$ the successive points selected by algorithm j and $(r_{i,j})_{1\leq i\leq N/M}$ the reward observed, the final output of $\texttt{POO}(\mathscr{A})$ can be written as

$$\widehat{x} = x_{I,\widehat{j}} \text{ where } I \sim \mathscr{U}(\{1,\dots,N/M\}) \text{ and } \widehat{j} = \arg\max_j \widehat{\mu}_j$$

with

$$\widehat{\mu}_j = \frac{M}{N}\sum_{i=1}^{N/M} r_{i,j}\,.$$

One can also define $\widetilde{j} = \arg\max_j \mu_j$ with

$$\mu_j = \frac{M}{N}\sum_{i=1}^{N/M} f(x_{i,j})$$

and $\overline{j}$ to be the index of the instance such that $\rho_{\overline{j}} = \overline{\rho}$. First, some concentration results (see Appendix B.4 of Grill et al. 2015) show that for all j, $\mathbb{E}\left[|\widehat{\mu}_j - \mu_j|\right] \leq C/\sqrt{N/M}$. The simple regret can then be upper bounded as

$$\begin{aligned}\mathbb{E}\left[S_N^{\texttt{POO}(\mathscr{A})}\right] &= \mathbb{E}\left[f^\star - f(\widehat{x})\right] = \mathbb{E}\left[f^\star - \frac{M}{N}\sum_{i=1}^{N/M} f(x_{i,\widehat{j}})\right] = \mathbb{E}\left[f^\star - \mu_{\widehat{j}}\right] \\ &= \mathbb{E}\left[f^\star - \mu_{\overline{j}}\right] + \mathbb{E}\left[\mu_{\overline{j}} - \mu_{\widetilde{j}}\right] + \mathbb{E}\left[\mu_{\widetilde{j}} - \widehat{\mu}_{\widetilde{j}}\right] + \mathbb{E}\left[\widehat{\mu}_{\widetilde{j}} - \widehat{\mu}_{\widehat{j}}\right] + \mathbb{E}\left[\widehat{\mu}_{\widehat{j}} - \mu_{\widehat{j}}\right]\end{aligned}$$

The second and fourth terms in this sum are negative by definition of $\widetilde{j}$ and $\widehat{j}$ respectively, while the third and last terms are $O(\sqrt{N/n})$ using the concentration result mentioned above. As for the first term, one has

$$\mathbb{E}\left[f^\star - \mu_{\overline{j}}\right] = \frac{M}{N}\mathbb{E}\left[\sum_{t=1}^{T}(f^\star - r_{i,\overline{j}})\right] = \frac{M}{N}\mathbb{E}\left[R_{N/M}^{\mathscr{A}(\nu_{\max},\overline{\rho})}\right] = \mathbb{E}\left[S_{N/M}^{\mathscr{A}(\nu_{\max},\overline{\rho})}\right],$$

where again the recommendation rule matters. Using the upper bound (D.33) obtained in the first part of the analysis permits to conclude by noting that the first term is actually the leading term.

Appendix E

Acronyms

AutoML *automated machine learning*. 88, 89

BAI *best-arm identification*. 4–7, 16–25, 43, 49, 62, 64, 65, 67, 70, 88, 92–94, 104, 106, 107

BBO *black-box optimization*. 3, 7, 72, 88

BO *Bayesian optimization*. 7, 91

BPI *best-policy identification*. 107

CASH *combined algorithm selection and hyper-parameter optimization*. 90

DL *deep learning*. 4, 88, 91, 107

EBA *empirical best arm*. 19

EDP *empirical distribution of plays*. 19

FMS *full model selection*. 90

GLRT *generalized likelihood ratio test*. 18

GO *global optimization*. 7, 72, 75, 88, 91

HCT High Confidence Tree. 73, 79–85, 179, 180, 182–186, 188

HOO Hierachical Optimistic Optimization. 73, 75, 78, 79, 81, 83–85

HPO *hyper-parameter optimization*. 3, 7, 43, 88–93, 104, 106, 107

iid *independent and identically distributed*. 13, 19

MAB *multi-armed bandits*. 3–6, 10, 15, 88, 107, 124

MDP *Markov decision process*. 3, 107

MPA *most played arm*. 19

NAS *neural architecture search*. 90, 107

OFU *optimism in the face of uncertainty*. 15

PAC *probably approximately correct*. 24

POO Parallel Optimistic Optimization. 72, 75, 76, 78, 81–85, 106

D-TTTS Dynamic Top-Two Thompson Sampling. 94, 104, 106

GPO General Parallel Optimization. 73, 76–78, 80, 81, 83, 85, 106

LinGame Linear Game. 62–67, 106

L-T3C Linear-Top-Two Transportation Cost. 56

L-T3S Linear-Top-Two Thompson Sampling. 56

PCT Parallel Confidence Tree. 83–85

SLinGapE Saddle-Point Linear Gap-Based Exploration. 69

SL-T3C Saddle-Point Linear-Top-Two Transportation Cost. 69

T3C Top-Two Transportation Cost. 28, 56, 106

RL *reinforcement learning*. 4, 107

SIAB *stochastic infinitely-armed bandits*. 88, 92, 97

TS Thompson Sampling. 6, 14, 28, 43

TTTS Top-Two Thompson Sampling. 28, 29, 43, 56, 89, 93–95, 106, 131, 135–137, 140, 142, 143, 152

UCB Upper-Confidence Bound. 14–16

ZO *zeroth-order optimization*. 7

Appendix F

Glossary

alternative set . 24, 49

asymptotic optimality . 6

Bayesian stopping rule . 19

continuum-armed bandits . 7, 23, 72

cumulative regret . 13

decision rule . 16

differential entropy . 126

entropy . 126

exploration-exploitation dilemma . 4

finitely-armed bandits . 7

fixed-budget setting . 16

fixed-confidence setting . 5, 16

generalized likelihood ratio . 18

generalized linear bandits . 22

G-optimality . 6

hyper-parameters . 3, 88

infinitely-armed bandits . 7, 23, 88, 106

natural sufficient statistic . 13

near-optimality dimension . 74, 75, 81

one-dimensional exponential family . 13

probability density function . 13

probability mass function . 13

pure exploration . 4

regression parameter . 6

regularized least-square estimation . 48

reservoir . 23, 88, 92

sample complexity . 6

sampling rule . 12, 16

sequential optimization . 3

simple regret . 7, 22, 25, 94

single-parameter exponential family . 13

stopping rule . 16

sub-optimality gap . 15

$\mathcal{X}$-armed bandits . 23