Bandit algorithms for recommender system optimization

Matthieu Jedor

Thèse de doctorat

NNT: 2020UPASM027

université PARIS-SACLAY

Bandit algorithms for recommender system optimization

Thèse de doctorat de l'Université Paris-Saclay

Ecole Doctorale de Mathématique Hadamard (EDMH) n° 574
Spécialité de doctorat : Mathématiques appliquées
Unité de recherche : Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, 91190, Gif-sur-Yvette, France
Référent : Ecole normale supérieure de Paris-Saclay

Thèse présentée et soutenue en visioconférence totale le 18 décembre 2020, par

Matthieu JEDOR

Au vu des rapports de :

Myriam Maumy-Bertrand Maître de conférences, UTT	Rapportrice
Philippe Preux Professeur, Université de Lille	Rapporteur

Composition du jury :

Romaric Gaudel Maître de conférences, ENSAI	Examinateur
Christophe Giraud Professeur, Université Paris-Saclay	Président
Jonathan Louëdec Docteur, Cdiscount	Coencadrant
Myriam Maumy-Bertrand Maître de conférences, UTT	Rapportrice
Philippe Preux Professeur, Université de Lille	Rapporteur
Vianney Perchet Professeur, ENSAE	Directeur


école
normale
supérieure
paris–saclay


université
PARIS-SACLAY


Fondation mathématique
FMJH
Jacques Hadamard


CENTRE
BORELLI


Cdiscount
N'ÉCONOMISEZ PAS VOTRE PLAISIR.

Abstract

In this Ph. D. thesis, we study the optimization of recommender systems with the objective of providing more refined suggestions of items for a user to benefit. The task is modeled using the multi-armed bandit framework. This thesis is divided into two parts: Chapter 3 and 4 study two problems encountered in recommender systems while Chapter 5 and 6 discuss about the practicality of bandit algorithms. Each chapter can be read independently of each other. The organization as follows:

- Chapter 1 presents the context and contributions of this thesis (in French).
- Chapter 2 is a general introduction to the stochastic multi-armed bandit problem.
- Chapter 3 introduces a new multi-armed bandit model where arms are grouped inside "ordered" categories with e-commerce as the motivating example. We prove instance-dependent lower bounds on the cumulative regret for three notions of dominance between categories, indicating how the complexity of the bandit problems increases with the generality of the ordering concept considered. We also provide algorithms that fully leverage the structure of the model with their associated theoretical guarantees. Finally, we have conducted an analysis on real data to highlight that those ordered categories actually exist in practice.
- Chapter 4 revisits the multi-armed bandit framework for online advertising in the pay-per-click model. In this setting, arms have a budget, a time of arrival and a lifetime. We present several bandit algorithms relevant to the setting and perform an empirical evaluation, both qualitative and quantitative, of these algorithms. Finally, we carry out a simulation with parameters from real data.
- Chapter 5 studies the problem of minimizing the total regret incurred over a series of tasks. While most bandit algorithms are designed to have a low worst-case regret, we examine here the average regret over bandit instances drawn from some prior distribution which may change over time. We specifically focus on confidence interval tuning of UCB algorithms. We further apply our solution to the mortal bandit problem, showing empirical improvement over previous work.
- Chapter 6 examines the greedy heuristic in multi-armed bandits. Although the Greedy algorithm is known to have a linear regret in the standard model, we show that it enjoys highly competitive performance in numerous settings. Theoretically, we prove that the Greedy algorithm run on a subset of arms verify near-optimal worst-case regret bounds in several models. Empirically, we perform experiments in numerous popular models that show that the Greedy algorithm outperforms the previous state-of-the-art in many cases.

Remerciements

Je tenais tout d'abord à remercier l'entreprise Cdiscount d'avoir financer mon doctorat et je pense en particulier à Matthieu Cornec qui est à l'origine de cette thèse. Grâce à vous, j'ai pu vivre une expérience exceptionnelle aussi bien sur le plan académique qu'industriel.

Vianney, je tiens à t'exprimer ma plus profonde gratitude. Depuis mon stage de M2, tu as toujours été présent quand j'en avais besoin et tu m'as toujours poussé à donner le meilleur de moi-même durant ces trois années (et le tout dans une ambiance très décontractée !). Je n'aurais pas pu rêver d'un meilleur directeur de thèse.

Un grand merci également à Jonathan, mon encadrant chez Cdiscount. Tu ne m'as jamais abandonné même quand mes recherches me menaient loin des problématiques de Cdiscount et tu t'es toujours démené pour trouver des projets qui me correspondaient. Je n'oublie pas également Bruno Goutorbe qui m'a encadré au début de ma thèse.

Je tiens aussi à remercier tout particulièrement Myriam Maumy-Bertrand et Philippe Preux d'avoir accepté de rapporter ma thèse et d'avoir supporté les (trop) nombreuses typographies présentes dans une version préliminaire de ce manuscrit. Vos retours ont grandement amélioré la qualité de ce manuscrit. Je remercie aussi très chaleureusement Christophe Giraud et Romaric Gaudel d'avoir accepté d'être membre de mon jury de thèse ainsi que l'intérêt que vous avez porté à mon manuscrit et à ma présentation.

Je pense aussi à tous les doctorants du CMLA/Centre Borelli, sans vous ces trois années n'auraient pas été aussi géniales. Je me souviens notamment de notre conférence à Toulouse tous ensemble qui ressemblait plus à des vacances. Je salue aussi l'ancien bureau 105 de Cachan qui était (de très loin) le meilleur bureau. Plus généralement, je salue toute l'équipe de Cachan : Mathilde, Alice, Firas, Etienne, Myrto, Théo, Antoine et Batiste. Merci également à la Team Vianney pour la qualité des discussions lors des group meetings (le flan est bien le meilleur dessert !). Je salue également l'équipe des "saints-pères" et l'équipe "image".

Je pense également aux équipes Pertinence et Data-Client que j'ai côtoyé chez Cdiscount. Je me souviens encore de ces nombreuses soirées foot ou encore badminton.

Je veux aussi saluer ma famille et mes amis qui m'ont soutenus (et supportés) durant ces années. J'ai une très grosse pensée pour Billy qui m'a accompagné pendant de nombreuses années. Je pense également à la Team 613/FP, vous m'avez toujours poussé à tout donné, même sur un tapis de course lancé à 10 km/h avec une inclinaison de 18%. Et pour finir, un petit clin d'œil à Vital qui peut maintenant voir le fruit de mon "emploi fictif".

Notation

Mathematics

$\mathbb{P}$	Probability
$\mathbb{E}$	Expectation
E^c	Complement of set E
$\mathbf{1}\{B\}$	Indicator of set B ($\mathbf{1}\{B\}(x) = 1$ if $x \in B$ and 0 otherwise)
$\lfloor x \rfloor, \lceil x \rceil$	Floor and ceiling functions of x
$e_1, \ldots, e_d$	Standard basis vectors of the d-dimensional Euclidean space
$\|x\|_p$	p-norm of vector x
$\langle x, y \rangle$	Inner product between x and y

Bandits

K	Number of arms
T	Time horizon
A_t	Choice of the learner at time t
X_t	Reward obtained at time t
$\mathcal{V}_k$	Distribution of arm k
μ_k	Mean reward of arm k
$\mu^\star$	Largest mean reward among all arms
Δ_k	Suboptimality gap of arm k ($\Delta_k = \mu^\star - \mu_k$)
$N_k(t)$	Number of pulls of arm k at time t
$R(T, \pi), R_T$	Regret of algorithm π at the end of time T (R_T when context is clear)
$\mathrm{BR}_Q(T, \pi)$	Bayesian regret of algorithm π at time T under prior distribution Q
$\widehat{\mu}_k(t)$	Empirical mean of rewards obtained for arm k at time t
$\mathrm{UCB}_k(t)$	Upper confidence bound of arm k at time t

Miscellaneous

$[n]$	$\{1, 2, \ldots, n-1, n\}$
$\mathrm{Beta}(\alpha, \beta)$	Beta distribution with parameters $\alpha, \beta > 0$
$f(n) = \mathcal{O}(g(n))$	Landau notation: $\limsup_{n \to \infty} \frac{f(n)}{g(n)} < \infty$

Contents

1 Introduction

Sommaire

Cette thèse s'inscrit dans le cadre d'une collaboration entre le Centre Borelli de l'École Normale Supérieure Paris-Saclay et l'équipe Pertinence de Cdiscount, responsable des résultats du moteur de recherche et des différentes recommandations présentes sur le site. Les modèles développés par l'équipe Pertinence sont principalement basés sur les différentes interactions passées entre les utilisateurs et le site Cdiscount. Leur objectif est d'améliorer l'expérience utilisateur en lui proposant des produits correspondant à son besoin.

Ce premier chapitre est l'occasion de revenir sur le contexte de cette thèse, ainsi que de rappeler l'histoire des systèmes de recommandation et des bandits multi-bras. Nous résumons également les problématiques et les contributions des différents chapitres.

1.1 Contexte de cette thèse

Deux problématiques sous-jacentes du site de commerce électronique Cdiscount motivent cette thèse.

La première problématique concerne la dynamique et la temporalité des informations traitées. Les approches actuelles se basent sur une représentation synthétique construite à partir d'informations disponibles une fois par jour. L'intégration de nouvelles informations en temps réel (flux de données) est un challenge pour une entreprise comme Cdiscount, qui interagit avec des millions d'utilisateurs chaque jour, avec l'arrivée de nouveaux utilisateurs et de nouveaux produits en permanence. Il s'agit de développer, dans des environnements non-stationnaires, des outils d'apprentissage séquentiel, où les nouvelles informations ont une influence directe sur les modèles estimés. En effet, Cdiscount a besoin de proposer des produits pertinents à ses utilisateurs (exploitation), tout en obtenant de nouvelles informations sur ses produits encore peu connus (exploration).

La seconde problématique est liée à l'adaptation des modèles au contexte. Le contexte fait référence à des connaissances implicites ou explicites concernant les besoins de l'utilisateur, son environnement et son comportement de navigation. Les produits du catalogue de Cdiscount étant nombreux et variés, nous sommes particulièrement intéressés par la personnalisation des produits présentés aux utilisateurs afin d'améliorer son expérience utilisateur. Ici il est important de souligner qu'avec des millions d'utilisateurs uniques chaque jour, les approches proposées devront prendre en considération un problème supplémentaire de volumétrie.

De tels problèmes sont au cœur des orientations récentes les plus actives prises par la recherche en machine learning. Les algorithmes de bandit sont connus pour offrir des solutions au dilemme exploration/exploitation. Cependant si ces algorithmes capturent l'essence de ce dilemme, ils s'avèrent insuffisants dans de nombreux cas, en particulier lorsque les données évoluent au fil du temps et sont contextualisées. Certains algorithmes de bandit prennent en compte l'un ou l'autre de ces deux verrous, mais ils sont conçus pour des modélisations simples et où chaque verrou est traité indépendamment de l'autre. De plus ils nécessitent d'être adaptés pour être utilisables sur des données réelles où de nombreux autres éléments entrent en jeu tels que des aspects multicritères (plusieurs métriques doivent être optimisées simultanément), ou des cas dans lesquels des contraintes fortes sont imposées (diversité notamment). Enfin peu d'expérimentations ont été effectuées sur un système en ligne avec un fort trafic comme celui de Cdiscount.

1.2 Systèmes de recommandation et bandits multi-bras

Dans cette section, nous rappelons ce qu'est un système de recommandation ainsi que le problème de base du bandit multi-bras. Nous passons également en revue les différents travaux réalisés dans ce modèle qui peuvent être utiles dans le cadre des systèmes de recommandation.

1.2.1 Systèmes de recommandation

De façon générale, les systèmes de recommandation sont des algorithmes qui visent à recommander des objets pertinents à un utilisateur. Le mot « objet » est ici un terme général et peut désigner une publicité dans le cadre de la publicité en ligne, un produit dans celui du commerce électronique ou encore un contenu dans les réseaux sociaux. Les systèmes de recommandation sont ainsi omniprésents dans nos vies de tous les jours. Dans la Figure 1.1, nous illustrons un système de recommandation du site de commerce électronique Cdiscount; il s'agit ici d'un carrousel de produits personnalisés présent sur la page d'accueil du site et basé sur l'historique de navigation d'un utilisateur. Signe supplémentaire de leur importance, l'entreprise Netflix a proposé il y a quelques années un concours où le but était de produire un algorithme capable d'améliorer de 10% les prédictions sur un jeu de données comparé à leur propre algorithme, avec au bout du compte un prix d'un million de dollars à gagner. Pour la petite histoire, ce prix a été remporté au bout de presque 3 ans d'efforts par une équipe composée de plusieurs chercheurs d'entreprises privées.

Données à disposition Commençons par préciser les données sur lesquelles sont entraînées les systèmes de recommandation. On distingue deux types de données sur un utilisateur : celle explicite et celle implicite. Les données explicites sont directement fournies par les utilisateurs. On peut par exemple demander à un utilisateur de noter son attrait pour un objet ou ce qu'il préfère

FIGURE 1.1 : Exemple de système de recommandation : carrousel de produits personnalisés sur la page d'accueil du site de commerce électronique Cdiscount.

entre deux choix. Les données implicites concernent elles les éléments relatifs à la navigation d'un utilisateur. Cela peut être les pages qu'il a consulté, sa fréquence de visite, ce sur quoi il a cliqué ou même le temps qu'il a passé sur une page.

APPROCHES CLASSIQUES On distingue principalement trois types d'approche des systèmes de recommandation pour suggérer des objets pertinents : la première est un filtrage utilisant les informations sur l'objet et l'utilisateur dit filtrage de contenu, la seconde repose sur un filtrage collaboratif et la dernière est une approche hybride.

Le filtrage de contenu prend en considération des informations supplémentaires que sont les préférences des utilisateurs et les descriptions des objets. L'idée est alors de construire un modèle basé sur toutes ces caractéristiques qui explique les interactions utilisateurs-objets passées. Par exemple, un algorithme très largement répandu est TF-IDF (de l'anglais Term Frequency-Inverse Document Frequency) qui consiste, dans ce cas, à pondérer les caractéristiques d'un objet en fonction des préférences d'un utilisateur. Le poids d'un objet augmente ainsi proportionnellement à son attrait par l'utilisateur. Cette méthode a le désavantage de devoir collecter des données sur les utilisateurs; cela peut notamment poser problème pour le respect de la vie privée. Un second problème est qu'elle va avoir tendance à proposer systématiquement des objets similaires à ceux auxquels un utilisateur a interagit; c'est ce que l'on appelle le problème de sur-spécialisation. Cette approche souffre ainsi d'un large biais mais d'une faible variance.

L'approche collaborative quant à elle se base uniquement sur les interactions utilisateurs-objets passées. L'idée est alors de détecter les objets similaires ainsi que les utilisateurs ayant des préférences communes. Il existe deux sous-catégories algorithmes dans cette approche. La première catégorie, nommée « model-based », assume l'existence d'un modèle sous-jacent qui explique les différentes interactions. Un exemple typique d'algorithme dans ce cas est la méthode de factorisation de matrices dont le but est de représenter les utilisateurs et les objets dans un espace de plus faible dimension. Au contraire, la seconde catégorie, communément appelée « memory-based », n'assume aucun modèle sous-jacent. Un algorithme classique dans ce cas est celui des plus proches voisins qui, pour un utilisateur donné, non seulement recommande des objets similaires à ceux qu'il a aimé mais va aussi recommander des objets aimés par des utilisateur avec des préférences communes. Cette approche souffre néanmoins du départ à froid : sans donnée sur un utilisateur, il est impossible de lui recommander quoi que ce soit. Il en est de même pour un nouvel objet. En théorie, cette seconde catégorie d'approches collaboratives souffre d'un faible biais mais d'une large variance et inversement pour la première catégorie.

La dernière approche est une approche hybride qui combine le filtrage de contenu et le filtrage collaboratif. C'est de nos jours l'approche la plus utilisée puisqu'elle permet de compenser certaines faiblesses de l'une ou l'autre approche. Il existe deux moyens de combiner les deux approches. Le premier consiste à entraîner un modèle de filtrage de contenu et un autre de filtrage collaboratif séparément puis de combiner leurs sorties. Le second moyen est d'entraîner un unique modèle en utilisant les interactions passées de l'approche collaborative ainsi que les données sur les utilisateurs et les objets de l'approche basé sur le contenu. Cette dernière méthode utilise généralement des réseaux de neurones pour se faire.

Évaluation des performances Un dernier point important est celui de l'évaluation des systèmes de recommandation. En effet, au delà de la précision de ses recommandations qui peut être mesurée de façon standard, par exemple à l'aide de l'erreur quadratique moyenne, d'autres critères plus abstraits entrent aussi en compte, comme la diversité et l'explicabilité des recommandations. Ces critères sont difficilement évaluables sur un jeu de données classique. Ainsi, la meilleure façon de comparer deux systèmes de recommandation est d'effectuer un test en ligne, dit « test A/B » qui consiste à proposer le premier algorithme à une partie des utilisateurs et le second à l'autre partie. Cependant, ce processus est généralement coûteux et demande un certain niveau de confiance dans les algorithmes à évaluer. Par ailleurs, l'évaluation hors-ligne d'algorithmes séquentiels pose de nombreuses problématiques en soit et est un domaine très actif de recherche.

1.2.2 Bandits multi-bras

Comme mentionné précédemment, nous modélisons ce problème à l'aide du cadre des bandits multi-bras. Ce modèle constitue également un sous-domaine de l'apprentissage par renforcement qui consiste pour un agent à apprendre, de par sa propre expérience, les actions à effectuer dans le but d'optimiser une certaine récompense. En effet, le problème du bandit multi-bras peut-être vu comme un processus décisionnel markovien avec un seul état. Bien que le nom « bandit » puisse prêter à sourire, il désigne en réalité une machine à sous avec un long manche qui, dans l'argot anglo-saxon, est communément appelé « one-armed bandit ». La motivation du premier article sur les bandits était d'ailleurs on ne peut plus sérieux puisqu'il s'agissait d'optimiser les essais cliniques ; ce travail de Thompson [137] date par ailleurs de 1933. Ce modèle a par la suite été formalisé en 1952 par Robbins [127].

Dans cette thèse, nous nous concentrons sur le modèle de bandit dit stochastique. De manière plus formelle, il s'agit d'un jeu de décision à temps discret où un agent interagit de manière séquentielle avec un jeu de $K \in \mathbb{N}$ distributions de probabilité $\mathcal{V}_1, \dots, \mathcal{V}_K$ aussi appelé bras. Soulignons ici que le nombre de bras K est connu mais la distribution $\mathcal{V}_k$ associée au bras $k \in \{1, \dots, K\}$ est elle inconnue. À un instant $t \in \mathbb{N}$, l'agent choisit un bras $A_t \in \{1, \dots, K\}$ et reçoit une récompense stochastique X_t tirée selon la distribution $\mathcal{V}_{A_t}$ du bras sélectionné. Cette étape est réitérée jusqu'à un instant $T \in \mathbb{N}$ que l'on appelle l'horizon et qui peut être connu ou non. Nous illustrons ces cycles dans la Figure 1.2. L'objectif le plus étudié est celui de maximiser la somme des récompenses obtenues, c'est à dire $\sum_{t=1}^{T} X_t$.

Il est clair que si les moyennes des bras étaient connues, l'algorithme optimale serait de jouer le bras avec la plus grande récompense moyenne à tous les pas de temps. Cet algorithme est d'ailleurs communément appelé « oracle ». Ces paramètres étant malheureusement inconnus, l'agent fait

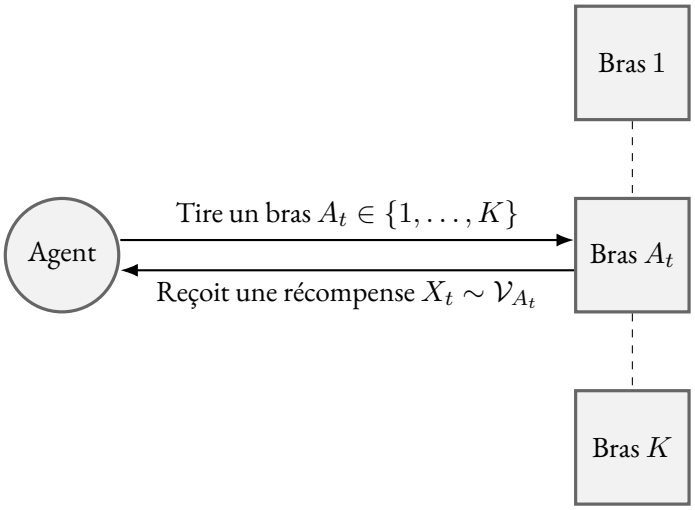


FIGURE 1.2 : Illustration d'un cycle dans un problème de bandit pour chaque tour $t = 1, \ldots, T$.

ainsi face à un dilemme que l'on appelle « exploitation vs. exploration ». En effet, l'agent doit choisir à chaque étape entre, explorer un bras de sorte à gagner de l'information sur celui-ci ou, exploiter le meilleur bras empiriquement.

Les problèmes de bandit sont un sujet de recherche très actif comme en témoigne les récents livres de BUBECK et CESA-BIANCHI [27], LATTIMORE et SZEPESVÁRI [100] et SLIVKINS [135]. Ils sont par ailleurs l'objet de nombreuses applications autre que les systèmes de recommandation comme aperçu par BOUNEFFOUF et RISH [24]. Nous proposons dans le Chapitre 2 une introduction plus détaillée du modèle de bandit.

1.2.3 BANDITS MULTI-BRAS APPLIQUÉS AUX SYSTÈMES DE RECOMMANDATION

La section précédente a décrit le modèle de base du bandit multi-bras. Depuis, plusieurs travaux se sont attardés à adapter ce modèle à différentes applications de sorte à améliorer les performances autant empiriquement que théoriquement. Nous détaillons ici les principales contributions dont l'objectif est l'optimiser des systèmes de recommandation.

Commençons tout d'abord par parler des bandits contextuels qui sont généralement utilisés en pratique. Pour les systèmes de recommandation, le contexte peut, par exemple, faire référence à des informations supplémentaires sur l'utilisateur et ce qu'il a fait par le passé. Du point de vue théorique, on suppose qu'à chaque pas de temps, le joueur observe un contexte avant de prendre une décision. L'idée est donc ici de tirer profit du contexte pour améliorer les recommandations. Cette prise en compte du contexte peut se marier avec n'importe quel modèle de bandit. LI, CHU, LANGFORD et SCHAPIRE [103], notamment, ont appliqué les bandits contextuels au modèle linéaire et ont évalué leur approche sur un jeu de données tiré de la page d'accueil du site d'actualités de l'entreprise Yahoo!.

Le modèle du bandit linéaire [1] est un moyen de tirer profit des similarités entre les bras, ce qui est utile quand leur nombre est grand. Dans ce modèle, les bras sont représentés par un vecteur de

caractéristiques et la récompense obtenue en tirant un bras est une fonction linéaire de son vecteur et d'un paramètre inconnu auquel s'ajoute un bruit.

Un autre article intéressant qui utilise le jeu de données précédemment évoqué est celui de Chapelle et Li [33]. Ils ont étudié, entre autres, l'impact du délai entre le choix du joueur et l'instant où il reçoit sa récompense. En effet, en pratique il se peut que cela ne soit pas immédiat à cause de diverses contraintes de temps d'exécution. Les algorithmes de bandit avec récompense différée sont un sujet actif de recherche.

Certaines approches visent à imiter les stratégies classiques des systèmes de recommandation vu précédemment. Gentile, Li et Zappella [60] ont proposé un algorithme basé sur le regroupement séquentiel des utilisateurs. Li, Karatzoglou et Gentile [106] ont généralisé le travail précédent en regroupant, en plus, les objets en fonction de la similitude des regroupements induits sur les utilisateurs. Maillard et Mannor [112] ont analysé un problème où les paramètres du modèle sont supposés être regroupés dans catégories inconnues. Kawale, Bui, Kveton, Tran-Thanh et Chawla [82] et Wang, Wu et Wang [147] ont eux proposé des algorithmes pour effectuer une factorisation de matrices de manière séquentielle. Les bandits stochastiques de rang 1 [76] sont un modèle particulier qui peuvent être appliqués aussi bien sur des regroupements d'utilisateurs et d'objets que dans des modèles de clics.

Des algorithmes de bandit ont par ailleurs été développés dans des modèles de clics. Kveton, Szepesvari, Wen et Ashkan [90] ont adapté des algorithmes standards dans le modèle de clics en cascade. Combes, Magureanu, Proutiere et Laroche [42] ont eux aussi considéré ce modèle de clics et ont développé un algorithme asymptotiquement optimal. Le modèle de clics basé sur la position a lui été étudié par Lagrée, Vernade et Cappe [93]. Lattimore, Kveton, Li et Szepesvari [97] et Zoghi, Tunys, Ghavamzadeh, Kveton, Szepesvari et Wen [155] ont considéré un modèle de clics plus général qui englobe les deux précédents. Katariya, Kveton, Szepesvari et Wen [77] ont étudié un modèle où plusieurs clics sont possibles.

D'autres travaux ont pris en compte le fait que les produits ont une durée de vie limitée. Ainsi, Chakrabarti, Kumar, Radlinski et Upfal [32] ont étudié un modèle où les bras sont régis par une durée de vie et de nouveaux bras apparaissent en permanence. Combes, Jiang et Srikant [40], Jiang et Srikant [72] et Slivkins [134] ont considéré un modèle où le tirage d'un bras induit un coût et un bras ne peut plus être choisit après que son budget soit épuisé.

Dans la même veine, certains travaux ont considéré un modèle non-stationnaire [59] où les récompenses des bras changent avec le temps. Louëdec, Rossi, Chevalier, Garivier et Mothe [109] ont discuté d'une décroissance de la récompense d'un bras en fonction du temps. Au contraire, Levine, Crammer et Mannor [102] ont étudié un décrochage de la récompense d'un bras sujet à son nombre de tirages.

1.3 Nos contributions

Cette thèse se divise en deux parties. Dans la première partie, nous nous attardons sur deux problématiques rencontrées dans le commerce électronique, mais que l'on rencontre aussi dans de nombreux systèmes de recommandation. La deuxième partie va elle se concentrer sur les performances des algorithmes de bandit en pratique. Plus précisément, comment on peut les améliorer et si il ne vaut pas mieux utiliser des heuristiques simples dans certains cas.

1.3.1 Problématiques du commerce électronique

Dans le Chapitre 3, nous considérons le problème du grand nombre de produits que doit traiter un site de commerce électronique. En effet, les bras du modèle de bandit sont ici les différents produits qui peuvent être affichés. Même si ce nombre est fini, il est en pratique prohibitif vu que le regret croît de manière linéaire avec le nombre de bras. Pour être exact, c'est la borne inférieure que l'on appelle « problème-dépendant » qui croît de façon linéaire ; le regret d'un algorithme ne pouvant être meilleur que cette borne, son regret croît, au mieux, linéairement avec le nombre de bras. Ainsi, un algorithme conventionnel va prendre un temps extrêmement long avant d'obtenir des performances satisfaisantes. Pour résoudre ce problème, nous allons exploiter une structure inhérente d'un site de e-commerce qui est celle des catégories dans lesquelles les produits sont classés. Comme un client est généralement intéressé par un faible nombre de catégories, le but est alors de regrouper les informations obtenues au sein d'une catégorie pour accélérer la phase d'apprentissage d'un algorithme de bandit et au final, proposer de meilleurs recommandations.

Sur le plan théorique, nous introduisons un nouveau modèle de bandit dans lequel les bras sont rangés dans des catégories dites « ordonnées », c'est à dire qu'il existe un ordre partiel, supposé connu, entre les catégories. Par conséquent, il existe une catégorie qui « domine » les autres mais cette catégorie est inconnue. Nous introduisons également trois concepts de dominance entre catégories qui sont progressivement plus faibles de sorte que de plus en plus de problèmes de bandit satisfont au moins l'un d'entre eux. Nous prouvons des bornes inférieures sur le regret pour chacun de ces concepts qui indiquent, entre autres, comment la complexité des problèmes de bandit augmente avec la généralité du concept de dominance considéré. Nous fournissons également deux algorithmes qui exploitent pleinement la structure du modèle et nous prouvons des garanties théoriques pour l'un d'entre eux. Enfin d'un point de vue plus appliqué, nous avons mené une analyse sur des données sur site de e-commerce Cdiscount pour souligner que l'on observe bien ces types de dominance dans la pratique.

Le Chapitre 4 traite le cas particulier des contenus sponsorisés qui représentent une source de revenue importante de nos jours pour les e-commerçants. On les repère facilement puisqu'ils sont souvent accompagnés du label « sponsorisé ». Le problème peut être décrit comme suit, pour un mot-clé donné, plusieurs annonceurs voudraient voir leurs produits s'afficher à une position stratégique et le moteur de recherche doit choisir lequel. L'annonceur sélectionné paie alors des frais uniquement lorsque son annonce a été cliquée. Ces contenus ont la spécificité d'être régis par un budget et une durée de vie. En effet, chaque annonceur dispose d'une somme maximale qu'il est prêt à dépenser et son annonce n'est disponible que sur une période donnée. Les budgets et les disponibilités sont généralement connus au début de la campagne, la seule inconnue est donc le taux de clics.

Dans ce Chapitre 4, nous modélisons un modèle de bandit où chaque bras dispose d'un budget, d'un temps d'arrivée et d'une durée de vie qui lui est propre. Ces caractéristiques ayant été étudiées séparément dans de précédents travaux, l'idée est ici de réunifier ces différents travaux. Ce modèle présente la particularité que la stratégie optimale n'est plus de tirer le meilleur bras à chaque étape. Pour s'en convaincre, considérons un exemple simple constitué de deux bras. Supposons qu'ils arrivent tous les deux en même temps et que le premier bras a une plus grande moyenne, dispose d'une plus longue durée de vie et d'un plus petit budget. Si on tirait systématiquement le premier bras, il se pourrait qu'une fois son budget épuisé, le second bras ne soit plus disponible. Si on

contraire, on commençait par tirer le second bras et ensuite le premier, on obtiendrait alors une meilleure récompense finale. Nous proposons alors plusieurs algorithmes de bandit et les évaluons de manière extensive sur de nombreuses simulations dont une tirée de données réelles.

1.3.2 Heuristiques en pratique

Le Chapitre 5 étudie l'optimisation d'algorithmes de bandit et spécifiquement ceux utilisant la borne supérieure d'un intervalle de confiance comme indice ; ce qui est notamment le cas du très populaire algorithme UCB (de l'anglais Upper Confidence Bound [9]). En effet, la plupart des travaux dans le littérature sur les bandits se concentrent sur l'obtention des meilleurs garanties possibles sur le regret. Pourtant, même si la théorie nous garantie un regret optimal dans le pire cas, les algorithmes optimaux vont être beaucoup trop conservateurs dans de nombreux cas et ainsi explorer inutilement, ce qui conduit à un large regret en pratique. Il a ainsi été montré que sur des problèmes relativement simples, de simples heuristiques obtiennent de meilleurs résultats que des algorithmes plus sophistiqués [89, 144] ; ce qui constitue un obstacle à leur usage en pratique.

Nous construisons alors un modèle où l'on résout de manière successive des problèmes de bandit. Ces problèmes sont tirés selon une distribution à priori, qui est inconnue et qui peut être stationnaire ou non. Le but est ainsi de construire un meta-algorithme qui va être en charge d'optimiser, par rapport à cette distribution, l'algorithme de bandit qui résout les tâches successives. Cette approche peut être vu comme un cas particulier de « meta learning » ou encore de « lifelong learning ». À titre d'exemple, si la distribution est connue et fixée, l'objectif est équivalent à celui de minimiser le regret Bayesien. Nous allons pour se faire opter pour un autre algorithme de bandit en charge de l'optimisation. Ce chapitre est principalement empirique et s'attarde sur l'intérêt et la façon d'optimiser un algorithme de bandit en pratique dans des environnements stationnaires et non-stationnaires. La méthode est ensuite appliquée au modèle du bandit mortel où les bras apparaissent et disparaissent constamment. Nous observons notamment des performances supérieures à l'état de l'art. Par ailleurs, nous montrons empiriquement qu'un simple algorithme glouton qui tire le meilleur bras empirique à chaque étape est plus performant que l'état de l'art dans le problème de bandit à bras continus.

Le Chapitre 6 va formellement prouver que l'algorithme glouton jouit d'un regret sous-linéaire dans ce cas. Plus globalement, nous y étudions l'heuristique gloutonne dans les problèmes de bandit. Notre analyse théorique se concentre sur l'utilisation de l'algorithme glouton sur un sous-échantillonnage de bras et nous prouvons des regrets sous-linéaires dans les cas de bandits continus, avec un nombre infinis de bras et avec un grand nombre de bras. Ces résultats théoriques sont complétés par de multiples expériences qui montrent les performances hautement compétitives de cet algorithme, notamment sur des horizons courts relativement à la complexité du problème considéré. Nous poursuivons également cette analyse empirique par plusieurs simulations dans divers modèles de bandit qui montre une nouvelle fois l'intérêt de l'algorithme glouton dans le cas où il existe de nombreux bras qui sont presque optimaux.

1.4 Plan du mémoire

Ce mémoire consacré à l'optimisation des systèmes de recommandation à l'aide du modèle de bandit multi-bras est ainsi divisé en deux parties. Les chapitres 3 et 4 étudient deux problèmes ren-

contrés dans les systèmes de recommandation alors que les chapitres 5 et 6 s'attardent sur la mise en pratique des algorithmes de bandit. Chaque chapitre peut être lu indépendamment des autres. Plus précisément :

- Le chapitre 2 est une introduction générale au problème du bandit multi-bras. Il contient ainsi tous le bagage nécessaire pour comprendre ce mémoire.
- Le chapitre 3 introduit un nouveau modèle de bandit où les bras sont rangés dans des catégories ordonnées. Le commerce électronique est ici l'exemple motivant.
- Le chapitre 4 revisite le problème de bandits pour la publicité en ligne. Nous prenons ainsi en compte le fait que les bras disposent d'un budget, d'une certaine durée de vie et que de nouveaux apparaissent constamment.
- Le chapitre 5 étudie l'optimisation d'un algorithme de bandit sur une série de problèmes.
- Le chapitre 6 examine l'heuristique gloutonne dans divers modèles de bandit. Nous nous posons notamment la question si un algorithme glouton peut être préférable à d'autres plus sophistiqués dans certains cas.

2 Stochastic multi-armed bandits

Contents

In this Ph. D. thesis, we are interested in the multi-armed bandit problem. This problem constitutes a sub-domain of reinforcement learning which consists, for an agent, in learning the actions to be carried out from his own experience in order to optimize a certain reward. More formally, the multi-armed bandit problem can be seen as a Markovian decision-making process with one state. The name "bandit" actually refers to a slot machine with a long handle informally called "one-armed bandit". This model was introduced without naming it in 1933 by Thompson [137] with the aim of optimizing clinical trials. It was subsequently formalized by Robbins [127].

In this chapter, we present the common base of the bandit theory. This is meant to be comprehensive enough to understand this thesis. Thus, we recall the stochastic multi-armed bandit model, the definition of the regret of an algorithm as well as lower bounds on achievable regret. We also review some commonly used bandit algorithms and present different applications beyond recommender systems. For readers interested in a broader introduction to bandit problems, we recommend the following books: Bubeck and Cesa-Bianchi [27], Lattimore and Szepesvári [100] and Slivkins [135]. We also recommend the article by Bouneffouf and Rish [24] for an overview of bandit applications.

2.1 Model

Here we focus on the stochastic multi-armed bandit model. Formally, it is a discrete-time decision-making game where a learner interacts sequentially with an unknown set of $K \in \mathbb{N}$ probability distributions $\mathcal{V}_1, \ldots, \mathcal{V}_K$ called arms. We emphasis that the number of arms K is known while the distribution $\mathcal{V}_k$ associated with arm k is unknown for each $k \in \{1, \ldots, K\}$. At time step t, the learner chooses an arm $A_t \in \{1, \ldots, K\}$ and receives a reward X_t drawn according to the distribution of the chosen arm $\mathcal{V}_{A_t}$. This step is reiterated for each time step $t = 1, \ldots, T$ where T is a time horizon which may or may not be known. We illustrate these cycles in Figure 2.1. We

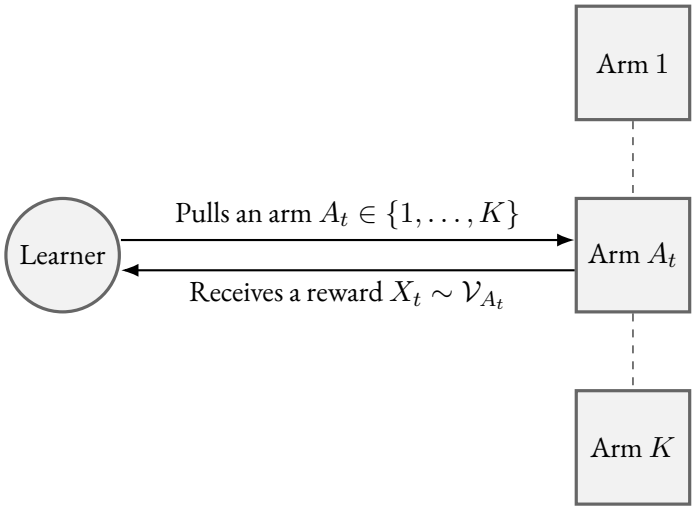


Figure 2.1: Illustration of a cycle in a bandit problem for a round $t \in \{1, \dots, T\}$.

also make use of a standard notation $[T] \coloneqq \{1, \dots, T\}$. The most commonly studied objective is to maximize the sum of the rewards obtained $\sum_{t=1}^{T} X_t$.

It is clear that if the mean rewards were known, the optimal algorithm would be to choose the best arm, that is the one with the largest expected reward. As usual in the literature, we assume that the best arm is unique. This algorithm is called the "oracle". The parameters of the problem being unknown, the learner thus faces a dilemma called "exploitation vs. exploration". Indeed, the learner must choose, at each time step, between exploring an arm to gain information on it and using the information collected so far to play the best arm in sight.

2.2 Regret of an algorithm

In the literature, the performance of an algorithm is usually measured by its regret, defined as the difference between the cumulative reward of the oracle and the one of the algorithm. Thus, maximizing the cumulative reward is equivalent to minimizing the regret. We denote by $\mu_1, \dots, \mu_K$ the expectation of the distributions associated with the arms and we define $\mu^\star = \arg\max_{k \in [K]} \mu_k$ the best arm, often called the optimal arm. The (expected) regret of an algorithm π is then written

$$\mathbb{E}[R(T, \pi)] = \mathbb{E}\left[T\mu^\star - \sum_{t=1}^{T} \mu_{A_t}\right] = T\mu^\star - \sum_{t=1}^{T} \mathbb{E}[\mu_{A_t}]$$

where the expectation is taken with respect to the randomness of the successive choices and the possible randomization of the algorithm. It is clear that the regret verifies $R(T, \pi) \geq 0$ and $R(T, \pi) \leq T\mu^\star$ for any horizon T and for any algorithm π. In particular $\mathbb{E}[R(T, \pi)] = \mathcal{O}(T)$ means that the algorithm π fails to find the best arm with a non negligible probability. Thus, we would like an algorithm which satisfies a sublinear regret, that is $\mathbb{E}[R(T, \pi)] = o(T)$.

If the context is clear, we will often drop the dependence on π and abbreviate the regret $R(T, \pi)$ of algorithm π by R_T.

There also exists a decomposition of the regret which is useful in theoretical analysis. Denote by $N_k(t) = \sum_{s=1}^{t} \mathbf{1}\{A_s = k\}$ the number of times the arm k has been played between the times 1 and t. We also define the suboptimality gap $\Delta_k = \mu^\star - \mu_k$ as being the difference between the mean of the best arm and of that of arm k. The regret can then be rewritten as

$$\mathbb{E}[R(T, \pi)] = \sum_{k=1}^{K} \Delta_k \mathbb{E}[N_k(T)] \,.$$

This decomposition is widely used to prove upper bounds on the regret of an algorithm. Indeed, thanks to this decomposition, it simply suffices to bound the expected number of times the algorithm draws a suboptimal arm k.

2.3 Assumption on reward distributions

An assumption often made in the literature, and which we too will do, is to assume that each distribution associated with an arm is subgaussian. We recall that we say a random variable is subgaussian if its tail distribution decreases as quickly as a Gaussian distribution with the same mean and the same variance. Formally,

Definition 2.1. Let $\sigma > 0$. A random variable X is σ^2-subgaussian if for all $\lambda \in \mathbb{R}$, we have

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\big(\lambda^2 \sigma^2 / 2\big) \,.$$

This implies, among other things, the following well-known theorem which explains the origin of the term and which we will use frequently.

Theorem 2.1. *If X is σ^2-subgaussian, then for any $\varepsilon \geq 0$, we have*

$$\mathbb{P}(X \geq \varepsilon) \leq \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right) .$$

A corollary which will be useful concerns the mean of subgaussian random variables.

Corollary 2.1. *Let $X_1, \ldots, X_n$ be n independent σ^2-subgaussian random variables. Then for any $\varepsilon \geq 0$, it holds that*

$$\mathbb{P}\big(\overline{X}(n) \geq \varepsilon\big) \leq \exp\left(-\frac{n\varepsilon^2}{2\sigma^2}\right)$$

where $\overline{X}(n) = \frac{1}{n} \sum_{i=1}^{n} X_i$.

Without loss of generality, we will assume thereafter that $\sigma = 1$ so as not to overload the notations.

Another common assumption is to assume that arms lie in a bounded interval, typically $[0, 1]$. Though that this implies that the distributions are $1/4$-subgaussian, tighter concentration bounds

should be picked. Additionally, similar concentration inequalities can be obtained with the Hoeffding inequality.

2.4 Regret lower bounds

We recall here two lower bounds on the regret that an algorithm can hope to achieve. For simplicity, we assume here that arms are Gaussian distributions with unit variance. Let us denote $\mathcal{I}$ the set of Gaussian bandits with $K \geq 2$ arms. In this section and in this section only, to specify the dependency on an instance $I \in \mathcal{I}$, we denote the regret of an algorithm π on instance I and horizon T by $R^I(T, \pi)$. First, we need the following definition to exclude algorithms that have a low regret on some instances and a linear regret on others. This is for example the case of algorithms that pull the same arm all the time.

Definition 2.2. An algorithm π is said to be consistent with $\mathcal{I}$ if for any instance $I \in \mathcal{I}$ and for all $\alpha \in (0, 1]$, its regret is negligible compared to T^α. In other words,

$$\sup_{\alpha \in (0,1)} \limsup_{T \to \infty} \frac{\mathbb{E}[R_I(T, \pi)])}{T^\alpha} = 0 \,.$$

Problem-dependent bound The first lower bound, called "problem-dependent", is due to Lai and Robbins [95]. As its name suggests, this lower bound is expressed as a function of the parameters of the problem that are the suboptimality gaps. It has had a considerable impact on the literature since most bandit algorithms are designed to reach this milestone.

Theorem 2.2. *An algorithm π consistent with $\mathcal{I}$ satisfies for all $I \in \mathcal{I}$*

$$\liminf_{T \to \infty} \frac{\mathbb{E}[R_I(T, \pi)]}{\log(T)} \geq c_I = \sum_{k : \Delta_k > 0} \frac{2}{\Delta_k^I}$$

where Δ_k^I is the suboptimality gap of arm k in instance I.

As an example, the UCB algorithm is asymptotically optimal for Gaussian bandits [100] and KL-UCB is also for Bernoulli bandits [56, 113]. The asymptotic regret is often indicative of finite-time performance. However, in practical regimes, the lower-order terms hidden by the asymptotics can be dominant. Garivier, Ménard, and Stoltz [58] highlighted what the second-order terms depend on; they additionally proved that the regret of any algorithm grows linearly in an initial phase.

Worst-case bound It may be that, for a fixed horizon T, a problem is so complex that this bound does not provide any information. This is for instance the case where an arm is arbitrarily close to the optimal one and thus difficult to distinguish given the time horizon. Thus, a second lower bound called "minimax" or "worst-case" has been proven by Auer, Cesa-Bianchi, Freund, and Schapire [10]. As indicated by its name, this bound is valid for each instance in a given set of bandit problems and is thus independent of the parameters of each instance, contrary to the first bound. We recall here the theorem of Lattimore and Szepesvári [100].

Theorem 2.3. *Let $K > 1$ and $T \geq K - 1$. Then, for any algorithm π, there exists an instance $I \in \mathcal{I}$ with a mean vector $\mu = (\mu_1, \dots, \mu_K) \in [0,1]^K$ such that*

$$\mathbb{E}[R_I(T, \pi)] \geq \frac{1}{27}\sqrt{(K-1)T}\,.$$

As an example, the MOSS algorithm [7] attains this lower bound, up to constant factors.

Also note that some algorithms are both asymptotically and minimax optimal; this is for example the case of KL-UCB^{++} [115] and AdaUCB [96].

2.5 A brief overview of bandit algorithms

Here we briefly review bandit algorithms focusing on two families of policies that we will use frequently in this thesis. The first one is based on the principle of optimism in the face of uncertainty, while the second one uses the Bayesian point of view.

Algorithm 1: UCB

for $t \leftarrow 1$ **to** K **do**
  Pull arm $A_t = t$
for $t \leftarrow K+1$ **to** T **do**
  Pull arm $A_t \in \arg\max_{k \in [K]} \widehat{\mu}_k(t-1) + \sqrt{\dfrac{6 \log T}{N_k(t-1)}}$

The optimism principle In the first family, we describe the keystone that is the UCB algorithm [9]. This is an index policy which assumes that arms are as good as possible knowing the information collected so far and chooses the best of them at each time step. Formally, the algorithm begins by pulling each arm once. Then, the index of arm k at time t is defined as follows. We define the average reward $\widehat{\mu}_k(t) = \frac{1}{N_k(t)} \sum_{s=1}^{t} X_s \mathbf{1}\{A_s = k\}$ and the confidence radius $r_k(t) = \sqrt{\frac{6 \log T}{N_k(t)}}$. The UCB index of the arm k at time t is then written

$$\mathrm{UCB}_k(t-1) = \widehat{\mu}_k(t-1) + r_k(t-1)\,.$$

This index is therefore decomposed in two terms, one representing the exploitation and the other the exploration. The exploration term is chosen so that the true expected reward belongs to the confidence interval thus created with high probability (it actually holds with a confidence level $1/T^3$ for each arm at each round). We can notice that this interval is constructed with the concentration bound on subgaussian variables seen previously (originally it was constructed with the Hoeffding inequality). The UCB algorithm therefore naturally controls the exploration vs. exploitation dilemma: arm k is drawn either because it has a large empirical reward or because it has a large exploration term due to the fact that it was not pulled enough. We have actually presented here a (slightly) modified version of the original algorithm which will be more useful to understand the analysis developed in this thesis (usually, the time horizon T is replaced by the time step

t so that the algorithm is anytime). We now give an upper bound on the regret of this algorithm, which was originally proven by Auer, Cesa-Bianchi, and Fischer [9].

Theorem 2.4. *On a stochastic K-armed 1-subgaussian bandit problem, the regret of UCB verifies*

$$\mathbb{E}[R_T] \leq 24 \sum_{k:\Delta_k>0} \frac{\log T}{\Delta_k} + \sum_{k:\Delta_k>0} \Delta_k + 1\,.$$

For completeness, we prove this theorem as similar arguments will be of use in subsequent proofs.

Proof. Without loss of generality, we assume the first arm is optimal, that is $\mu^\star = \mu_1$. The theorem will be proven by bounding $\mathbb{E}[N_k(T)]$ for each suboptimal arm k and the result is obtained thanks to the regret decomposition. We make use of a standard technique which is to define a "good" event in order to distinguish the randomness of the distributions from the behavior of the algorithm.

Let E be the good event defined by

$$E = \left\{\forall t \in [T], \forall k \in [K] : |\widehat{\mu}_k(t) - \mu_k| \leq \sqrt{\frac{6\log T}{N_k(t)}}\right\}.$$

Let $k \in [K]$ be any suboptimal arm. We can thus further decompose its number of pulls by

$$\mathbb{E}[N_k(T)] \leq 1 + \mathbb{E}[\mathbf{1}\{E\}N_k(T)] + \mathbb{E}[\mathbf{1}\{E^c\}N_k(T)] \leq 1 + \mathbb{E}[\mathbf{1}\{E\}N_k(T)] + T\mathbb{P}(E^c)$$

where in the last inequality we used that $N_k(T) \leq T$ and the "1" term results from the initialization.

First, we bound $\mathbb{P}(E^c)$. With a slight abuse of notation, we denote $\overline{X}_k(s)$ the empirical mean of arm k after s pulls. Since it is $\frac{1}{s}$-subgaussian, we have

$$\mathbb{P}\left(\left|\overline{X}_k(s) - \mu_k\right| > \sqrt{\frac{6\log T}{s}}\right) \leq \frac{2}{T^3}\,.$$

Taking a union bound over $k \in [K]$ and $s \in [T]$, we obtain

$$\mathbb{P}\left(\exists s \in [T], \exists k \in [K] : \left|\overline{X}_k(s) - \mu_k\right| > \sqrt{\frac{6\log T}{s}}\right) \leq \frac{2}{T}\,.$$

In particular, this implies that $\mathbb{P}(E^c) \leq \dfrac{2}{T}$.

On the other hand, we bound the number of pulls $\mathbb{E}[\mathbf{1}\{E\}N_k(t)]$ for $t \leq T$. At time $t+1$, arm k is pulled if

$$\mathrm{UCB}_k(t) \geq \mathrm{UCB}_1(t)\,.$$

Since the good event holds, we also have

$$\begin{aligned}\mathrm{UCB}_k(t) &\leq \mu_k + 2r_k(t)\\ \mathrm{UCB}_1(t) &\geq \mu_1\end{aligned}$$

Putting the previous inequalities together yields

$$N_k(t) \leq \frac{24\log T}{\Delta_k^2}\,.$$

In particular, $\mathbb{E}[\mathbf{1}\{E\}N_k(T)] \leq \frac{24\log T}{\Delta_k^2}$ and the proof is conclude. □

This theorem tells us that the UCB algorithm is asymptotically optimal, with respect to the parameter-dependent lower bound, up to a constant factor. We mention that a different choice of confidence level and a more careful analysis lead to a tighter bound. We can also use this result to obtain a minimax upper bound on the regret of the UCB algorithm.

Theorem 2.5. *On a stochastic K-armed 1-subgaussian bandit problem, the regret of UCB verifies*

$$\mathbb{E}[R_T] \leq 10\sqrt{KT\log T} + \sum_{k:\Delta_k>0}\Delta_k + 1\,.$$

Proof. Let $\varepsilon > 0$ to be tuned later. Using the decomposition of the regret, we have

$$\begin{aligned}\mathbb{E}[R_T] &= \sum_{k=1}^{K}\Delta_k\mathbb{E}[N_k(t)]\\ &= \sum_{k:\Delta_k<\varepsilon}\Delta_k\mathbb{E}[N_k(t)] + \sum_{k:\Delta_k\geq\varepsilon}\Delta_k\mathbb{E}[N_k(t)]\\ &\leq T\varepsilon + \sum_{k:\Delta_k\geq\varepsilon}\frac{24\log T}{\Delta_k} + \sum_{k:\Delta_k\geq\varepsilon}\Delta_k + 1\\ &\leq T\varepsilon + \frac{24K\log T}{\varepsilon} + \sum_{k:\Delta_k>0}\Delta_k + 1\\ &\leq 2\sqrt{24KT\log T} + \sum_{k:\Delta_k>0}\Delta_k + 1\end{aligned}$$

where in the first inequality we have used $\sum_{k:\Delta_k<\varepsilon} N_k(t) \leq T$ and the result of the previous theorem, and the last row is obtained by taking $\varepsilon = \sqrt{\frac{24K\log T}{T}}$. □

We thus see that there exists a logarithmic factor with respect to the horizon T in excess compared to the minimax lower bound.

Many variants of the UCB algorithm have been proposed: UCB-V [8] uses variance estimates to tune the confidence radius, KL-UCB [56, 113] concerns Bernoulli bandits, KL-UCB^{++} [115]

and AdaUCB [96] are designed to be both asymptotically and minimax optimal, and many others. The difference between them concerns either the assumption on rewards distributions or the objective in terms of regret.

Algorithm 2: Thompson Sampling for Bernoulli bandits

Initialization: Set $S_k = 0$ and $F_k = 0$ for $k = 1, \dots, K$
for $t \leftarrow 1$ **to** T **do**
  for $k \leftarrow 1$ **to** K **do**
    Sample $\theta_k(t) \sim \text{Beta}(S_k + 1, F_k + 1)$
  Pull arm $A_t \in \arg\max_k \theta_k(t)$ and observe reward X_t
  if $X_t = 1$ **then**
    $S_{A_t} = S_{A_t} + 1$
  else
    $F_{A_t} = F_{A_t} + 1$

Bayesian principle We describe here the Thompson Sampling algorithm which is extremely popular in the literature. It is actually the first bandit algorithm introduced in 1933 by Thompson [137] (without any theoretical guarantee). It regains popularity recently due to its excellent empirical performance [33]. The theoretical analysis then followed [6, 81].

The Thompson Sampling algorithm consists in placing a prior distribution on each arm. At each step, a sample is drawn from each posterior distribution and the arm with the highest value is chosen. The distribution of the selected arm is then updated. Formally, let us denote by $\mathcal{F}(t) = \sigma(A_1, X_1, \dots, A_t, X_t)$ the information available after t steps. Let $\Pi_t = \mathbb{P}(\cdot|\mathcal{F}_t)$ be the posterior distribution of the means parameters at the end of step t. The algorithm then samples, at each time step, from the posterior distribution Π_{t-1} and pulls the arm with the best sample.

A more concrete example of Thompson Sampling is in the case of Bernoulli bandits. Indeed a natural choice of prior for a Bernoulli distribution is the Beta distribution. Initially, the prior is usually chosen to be uniform, that is a Beta(1, 1) prior. Then, after arm k has been pulled, its distribution is update to a Beta($S_k(t) + 1, N_k(t) - S_k(t) + 1$) distribution where $S_k(t)$ denotes the number of successes of arm k until time t. Thompson Sampling was proven to achieve asymptotical optimal problem-dependent bound in this case [6, 81]. We also mention that the choice of the prior has a significant effect on the performance of Thompson Sampling.

There exists another algorithm that use both the Bayesian point of view and the optimism principle: the Bayes-UCB algorithm [79]. Indeed, instead of sampling from the posterior distribution of each arm like Thompson Sampling, Bayes-UCB computes a given quantile value for each arm and pulls the best one like UCB. It also achieves an asymptotical optimal problem-dependent bound for Bernoulli bandits.

Explore or Exploit principle We regroup here three types of algorithms: Explore-Then-Commit [100], SuccessiveElimination [54] and Epsilon-Greedy [9]. Explore-Then-Commit pulls each arm successively until a predefined time step and then commit to the best arm

empirically. Its main drawback is that it must know the parameters of the problem to perform optimally. SuccessiveElimination also pulls each arm successively but it eliminates arms that are evaluated suboptimal over time. Epsilon-Greedy spreads the exploration more uniformly over time: at each time step t, it explores a random arm with probability ε_t, otherwise it pulls the best arm in sight. It has the same drawback as Explore-Then-Commit, it must know the parameters of the problem to perform optimally. All these algorithms share the fact that they either explore or exploit as opposed to UCB where this is a mix of both. They also are asymptotically optimal up to constant factors. However, their disadvantage is that the exploration is uniform over arms which hurt their performance in practice. Garivier, Lattimore, and Kaufmann [57] actually proved that Explore-Then-Commit is necessarily suboptimal by a factor 2.

Tracking principle The last family of algorithms worth mentioning is what we called tracking algorithms: DMED [67], IMED [68] and OSSB [41]. These algorithms share the common idea that is to pull arms that do not satisfy the inequality on their number of draws resulting from the problem-dependent lower bound. They pull the best empirical arm if all arms verify their conditions. They simply differ in the way they explore. Thereby they satisfy asymptotical optimal problem-dependent bounds.

2.6 Another performance index: the Bayesian regret

Another notion of regret that we will use later is the Bayesian regret. As the name suggests, we assume here the existence of a prior distribution which regulates the set of possible reward distributions. For instance, a simple case is when the prior distribution controls expected rewards. The idea is then to study the regret of an algorithm knowing this distribution. Since we study a subset of problems, we would ideally like to have stronger theoretical guarantees. Formally, in the parametric framework, a random vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)$ is generated according to a prior distribution Q, and the distribution $\mathcal{V}_k$ of arm k depends on the parameter θ_k. The Bayesian regret is then defined as follows

$$\mathrm{BR}_Q(T, \pi) = \int \mathbb{E}[R(T, \pi)] dQ(\boldsymbol{\theta}) \,.$$

Note that the optimal Bayesian regret is necessarily smaller than the minimax regret since we evaluate an algorithm an a subset of problems. And we cannot actually do better, up to constant factors, for all prior distributions [100].

An optimal dynamic-programming algorithm for Bayesian bandits is known as Gittins index [62] but unfortunately it is most of the time intractable. We refer the interested reader to Lattimore and Szepesvári [100, Chapter 35, and references therein] for an exhaustive overview on Bayesian bandits.

2.7 Applications

We conclude this section with a short description of several applications range bandit problems can be applied to. Even is the main motivation of this thesis concerns recommender systems, some results might be of independent interest.

Clinical trials This was the main motivation of the first article on bandit problems [137]. An arm here represents a potential treatment and the goal is to identify the best one in a minimal number of trials. Unfortunately, bandit algorithms still have not been used in clinical trials [145].

Black-box stochastic optimization Consider the problem of hyperparameter selection of an algorithm. If it returns noisy rewards, we can treat it at a bandit problem where arms are possible parameter values. We thus obtain strategies to refine the parameter over time. Chapter 5 in particular considers the optimization of the UCB algorithm. Another possibility is to fix the budget, that is the number of times we can run the algorithm and the goal is then is identify the best value at the end of this "test" phase.

Tree search One of the successful application of bandit algorithms comes from AlphaGo [133], an algorithm that become famous by outplaying world-class players at Go, a game known for its complexity. Indeed, one of its keystone is a tree search algorithm that make use of a bandit algorithm at each node [87].

Routing problem Consider the problem of sending packets from one vertex to another in a network represented by a graph with the objective of finding the shortest path. It can be modeled as a bandit problem where arms are the set of paths between the two points, each packet represent a time step and the feedback is the time taken by a packet to reached its destination. This application is one of the main motivations for combinatorial bandits.

Cognitive radio We can view the problem of communication between a wireless device and a gateway as a bandit problem where arms are the different possible channels. Actually, the problem is much more complicated as many devices may try to communicate at the same time; which lead to the study of multi-players bandits.

Part I

Bandit algorithms for e-commerce

3 Categorized multi-armed bandits

Contents

In this chapter, we introduce a new stochastic multi-armed bandit setting where arms are grouped inside "ordered" categories. The motivating example comes from e-commerce, where a customer typically has a greater appetence for items of a specific well-identified but unknown category than any other one. We introduce three concepts of ordering between categories, inspired by stochastic dominance between random variables, which are gradually weaker so that more and more bandit scenarios satisfy at least one of them. We first prove problem-dependent lower bounds on the cumulative regret for each of these models, indicating how the complexity of the bandit problem increases with the generality of the ordering concept considered. We also provide algorithms that fully leverage the structure of the model with their associated theoretical guarantees. Finally, we have conducted an analysis on real data to highlight that those ordered categories actually exist in practice.

3.1 Introduction

The traditional bandit model must be adapted to specific applications to unleash its full power. Consider for instance e-commerce. One of the core optimization problem is to decide which products to recommend, or display, to a user landing on a website, in the objective of maximizing the click-through-rate or the conversion rate. Arms of recommender systems are the different

products that can be displayed. The number of products, even if finite, is prohibitively huge as the regret, i.e. the learning cost, typically scale linearly with the number of arms. So agnostic bandit algorithms take too much time to complete their learning phase. Thankfully, there is an inherent structure behind a typical catalogue: products are gathered into well-defined categories. As customers are generally interested in only one or a few of them, it seems possible and profitable to gather information across products to speed up the learning phase and, ultimately, to make more refined recommendations.

Our results We introduce and study the idea of *categorized bandits*. In this framework, arms are grouped inside known categories and we assume the existence of a partial yet unknown order between categories. We aim at leveraging this additional assumption to reduce the linear dependency in the total number of arms. We present three different partial orders over categories inspired by different notions of stochastic dominance between random variables. We considered gradually weaker notions of ordering in order to cover more and more bandit scenarios. On the other hand, the stronger the assumption, the more "powerful" the algorithms can be, i.e. their regret is smaller. Those assumptions are motivated and justified by real data gathered on the e-commerce website Cdiscount. We first prove asymptotic problem-dependent lower bounds on the cumulative regret for each of these models, with a special emphasis on how the complexity of the bandit problems increases with the generality of the ordering concept considered. We then proceed to develop two generic algorithms for the categorized bandit problem that fully leverage the structure of the model; the first one is devised from the principle of optimism in the face of uncertainty [9] when the second one is from the Bayesian principle [137]. Finite-time problem-dependent upper bounds on the cumulative regret are provided for the former algorithm. Finally, we conduct numerical experiments on different scenarios to illustrate both finite-time and asymptotic performances of our algorithms compared to algorithms either agnostic to the structure or only taking it partly into account.

3.2 Related work

The idea of clustering is not novel in the bandit literature [26, 60, 88, 105, 117] yet they mainly focus on clustering users based on their preferences. Li, Karatzoglou, and Gentile [106] extended these work to the clustering of items as well. Katariya, Jain, Sengupta, Evans, and Nowak [74] considered a problem where the goal is to sort items according to their means into clusters. Similar in spirit are bandit algorithms for low-rank matrix completion [75, 83, 153]. Maillard and Mannor [112] studied a multi-armed bandit problem where arms are partitioned into latent groups. Valko, Munos, Kveton, and Kocák [143] and Kocák, Valko, Munos, and Agrawal [86] proposed algorithms where the features of items are derived from a known similarity graph over the items. However, none of these works consider the known structure of categories in which the items are gathered.

The model fits in the more general structured stochastic bandit framework i.e. where expected reward of arms can be dependent [2, 49, 91, 98, 123]. More recently, Combes, Magureanu, and Proutiere [41] proposed an asymptotically optimal algorithm for structured bandits relying on forced exploration (similarly to Lattimore and Szepesvari [99]) and a tracking mechanism on the

number of draws of sub-optimal arms. However, these approaches forcing exploration are too conservative as the linear dependency only disappears asymptotically.

There exist two other ways to tackle the bandit problem with arms grouped inside categories. The first one could rely on tree search methods, popularized by the celebrated UCT algorithm [87]. Alternative hierarchical algorithms [43] could also be used. The second one could be linear bandits [1, 45, 128] where we introduce a "categorical" feature that indicates in which category the arm belongs. However, these approaches are also not satisfactory as they do not leverage the full structure of the problem.

3.3 Model

We now present the variant of the multi-armed bandit model we consider. As usual, a decision maker sequentially selects (or pulls) an arm at each time step $t \in \{1, \dots, T\} =: [T]$. As motivated in the introduction, the total number of possible arms can be prohibitively large, but we assume that this large number of arms are grouped in a small number M of categories. For the sake of presentation, we are going to assume that each category has the same number of arms K, yet all of our assumptions and results immediately generalize to different number of arms. We emphasize again that the M categories of K arms each form a known partition of the set of arms (of cardinality MK). At time step $t \in [T]$, the agent selects a category C_t and an arm $A_t \in C_t$ in this category. This generates a reward $X_t = \mu_{A_t}^{C_t} + \eta_t$ where η_t is some independent 1-subgaussian white noise and μ_k^m is the unknown expected reward of the arm k of category m. For notational convenience, we will assume that arms are ordered inside each category, i.e. $\mu_1^m > \mu_2^m \geq \cdots \geq \mu_{K-1}^m > \mu_K^m$ for all category m and that category 1 is the best category, with respect to a partial order defined below. To be precise, since the order is only partial, some categories might not be pairwise comparable, but we assume that the optimal category is comparable to, and dominates, all the others. We stress out that, in the partial orders we consider, the maximum of μ_k^m over m and k is necessarily μ_1^1. As in any multi-armed bandit problem, the overall objective of an agent is to maximize her expected cumulative reward until time horizon T or identically, to minimize her expected cumulative regret $\mathbb{E}[R_T] = T\mu_1^1 - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}^{C_t}\right]$, or equivalently, $\mathbb{E}[R_T] = \sum_{m,k} \Delta_{m,k}\, \mathbb{E}[N_k^m(T)]$, where $\Delta_{m,k} := \mu_1^1 - \mu_k^m$ is the difference, usually called "gap", between the expected rewards of the best arm and the k^{th} arm of category m and $N_k^m(t) := \sum_{s=1}^{t} \mathbf{1}\{C_s = m, A_s = k\}$ denotes the number of times this arm has been pulled up to time step t. We also introduce here the notation $\Delta_{m,k}^{n,l} := \mu_l^n - \mu_k^m$ to compare two different arms.

Relations of dominance The main assumption to leverage is that the set of categories is partially ordered with a unique maximal element. Those partial orders are quite similar to the standard ones induced by stochastic dominance [16, 64] over random variables. We are going to consider three notions of dominance (inducing three different partial orders) that are gradually weaker so that the bandit setting is more and more general. Consequently, the regret should be higher and higher.

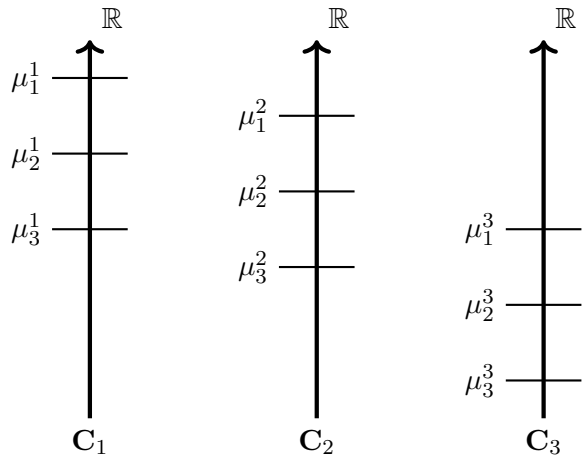


Figure 3.1: Illustration of dominances on three (imaginary) categories

Definition 3.1. Let $\mathbf{A} = \{\mu_1^{\mathbf{A}}, \dots, \mu_K^{\mathbf{A}}\} \subset \mathbb{R}$ and $\mathbf{B} = \{\mu_1^{\mathbf{B}}, \dots, \mu_K^{\mathbf{B}}\} \subset \mathbb{R}$ be a pair of categories,

Group-sparse dominance $\mathbf{A}$ group-sparsely dominates $\mathbf{B}$, denoted by $\mathbf{A} \succeq_s \mathbf{B}$, if each element of $\mathbf{A}$ are non-negative and at least one is positive, and each element of $\mathbf{B}$ are non-positive, i.e., $\max_{k\in[K]} \mu_k^{\mathbf{A}} > \min_{k\in[K]} \mu_k^{\mathbf{A}} \geq 0 \geq \max_{k\in[K]} \mu_k^{\mathbf{B}}$.

Strong dominance $\mathbf{A}$ strongly dominates $\mathbf{B}$, denoted by $\mathbf{A} \succeq_0 \mathbf{B}$, if each element of $\mathbf{A}$ is bigger than any element of $\mathbf{B}$, i.e., $\min_{k\in[K]} \mu_k^{\mathbf{A}} \geq \max_{k\in[K]} \mu_k^{\mathbf{B}}$.

First-order dominance $\mathbf{A}$ first-order dominates $\mathbf{B}$, denoted by $\mathbf{A} \succeq_1 \mathbf{B}$, if $\sup_{x\in\mathbb{R}} F_{\mathbf{A}}(x) - F_{\mathbf{B}}(x) \leq 0$, where $F_{\mathbf{A}}(x) = \frac{1}{K}\sum_{k=1}^K \mathbf{1}\{\mu_k^{\mathbf{A}} \leq x\}$ is the cumulative distribution function of a uniform random variable over $\mathbf{A}$ (and similarly for $\mathbf{B}$).

The first notion of dominance is inspired by the classic (group-)sparsity concept in machine learning, that already emerged in variants of multi-armed bandits [29, 92]. It is quite a strong assumption as it implies the knowledge of a threshold[1] between two categories. The second notion weakens this assumption as the threshold is unknown. The third notion is even weaker. The second and third notions of dominance are similar to the zeroth (also called strong) and first-order of stochastic dominances between two random variables respectively uniform over $\mathbf{A}$ and $\mathbf{B}$. Hence, the three concepts of dominance immediately generalize to categories with different number of elements, with the very same definitions. Furthermore, one can weaken even more the dominance, e.g. introducing a second-order variant, but we will not consider it in this paper.

Example To illustrate the concepts of dominance, we have represented, in Figure 3.1, 3 (imaginary) categories of 3 arms each. It can be easily checked that, for the first-order dominance,

[1] This threshold is here fixed at 0 for convenience, but it could have any value.

$\mathbf{C}_1$	$\mathbf{C}_2$	$\mathbf{C}_3$	$\mathbf{C}_4$
0.0133	0.0140	0.0089	0.0069
0.0114	0.0088	0.0086	0.0063
0.0108	0.0083	0.0078	0.0053
0.0107	0.0082	0.0056	0.0051
0.0096	0.0078	0.0052	0.0051
0.0095	0.0078	0.0050	0.0044
0.0088	0.0078	0.0049	0.0042
0.0086	0.0077	0.0047	0.0041
0.0084	0.0076	0.0042	0.0040
0.0080	0.0074	0.0041	0.0038

Table 3.1: Click-through rates of the four categories on the dataset.

$\mathbf{C}_1 \succeq_1 \mathbf{C}_2 \succeq_1 \mathbf{C}_3$ as, if they have the same number of elements, $\mathbf{A}$ first-order dominates $\mathbf{B}$ if the k^{th} largest elements of $\mathbf{A}$ is greater than the k^{th} largest element of $\mathbf{B}$, for any k. Moreover, for the strong dominance, $\mathbf{C}_1 \succeq_0 \mathbf{C}_3$ since the worst mean of $\mathbf{C}_1$ is higher than the best mean of $\mathbf{C}_3$. Moreover, if this common value was known, then the dominance would even be group-sparse.

Lemma 3.1. *Let $\mathbf{C}_1, \ldots, \mathbf{C}_M$ be finite categories. If there is a category $\mathbf{C}^\star$ that dominates all the other ones for any of the partial orders defined above, then $\mathbf{C}^\star$ contains the maximal element of the union $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \ldots \cup \mathbf{C}_M$. Moreover, if $\mathbf{A}$ group-sparsely dominates $\mathbf{B}$, then the dominance also holds in the strong sense. Similarly, if $\mathbf{A}$ strongly dominates $\mathbf{B}$, then the dominance also holds in the first-order sense.*

The proof is almost immediate, hence omitted.

3.4 Empirical evidence of dominance

We illustrate these assumptions on a real dataset. We have collected the click-through rates of products in four different categories over one month on the e-commerce website Cdiscount, one of the leading e-commerce companies in France, gathered in Table 3.1. Categories $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ are three of the largest categories[2] in terms of revenue while $\mathbf{C}_4$ is a smaller category. The following dominances can be highlighted.

Strong dominance $\mathbf{C}_1$ strongly dominates $\mathbf{C}_4$ as its minimum CTR is 0.008 compared to the maximum CTR of 0.0069 for the other. Similarly, $\mathbf{C}_2$ strongly dominates $\mathbf{C}_4$.

First-order dominance $\mathbf{C}_2$ first-order dominates $\mathbf{C}_3$ as the CTR of each line of the second column are bigger than those of the third column. This dominance is not strong as 0.0074 is smaller than 0.0089. $\mathbf{C}_3$ first-order but not strongly dominates $\mathbf{C}_4$.

[2] For privacy reason, the exact content of the different categories cannot be revealed.

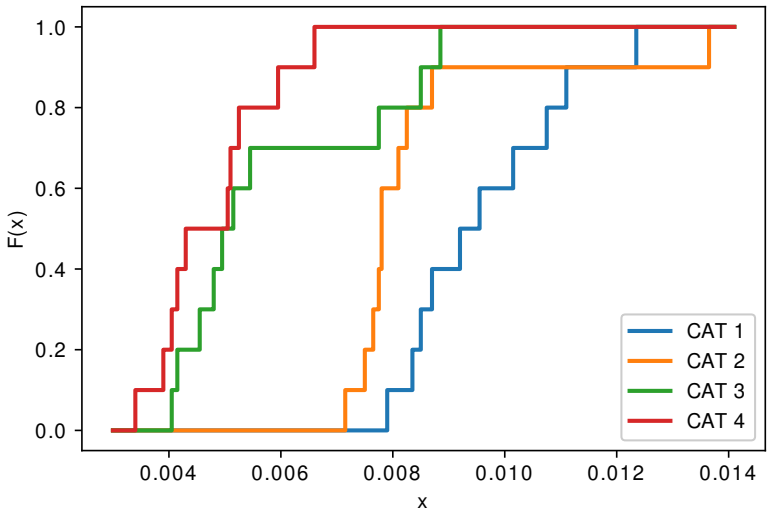


Figure 3.2: Cumulative distribution functions of the four categories on the dataset.

Uncomparable categories $\mathbf{C}_1$ and $\mathbf{C}_2$ are not comparable with respect to any partial order.

Notice that, had the first item of $\mathbf{C}_2$ performed only 5% worse than observed, then $\mathbf{C}_1$ would have been optimal with respect to the first-order dominance; the click-through rate of the best item of $\mathbf{C}_2$ is so higher than the second one, we could expect it is actually an outlier, i.e. an artefact of the choice of that specific month and category. So even if the dominance assumption is not satisfied during that specific month, assuming it would still give good empirical results.

The relations of dominance can be easier to determine based on the representation of the associated cumulative distribution functions of Figure 3.2. As the cumulative distribution function of the uniform random variable on category $\mathbf{C}_4$ is, pointwise, the biggest one, this means that this category is first-order dominated by all the other ones. Moreover, it reaches 1 while the cdf of $\mathbf{C}_1$ and $\mathbf{C}_2$ are still at 0. This implies that the dominance of these two categories is even strong. This analysis motivates and validates our assumption.

3.5 Lower bounds

In this section, we provide lower bounds on the regret that any "reasonable" algorithm (the precise definition is given below) must incur in a multi-armed bandit problem where arms are grouped into partially ordered categories (with a dominating one). To simplify the exposition, we assume here that noises are drawn i.i.d. from a Gaussian distribution with unit variance. The class of algorithms we consider are consistent [95] with respect to a given a class of possible bandit problems $\mathcal{M} = \{\mu = (\mu_1, \dots, \mu_{MK}) \in \mathbb{R}^{MK}\}$. We recall that an algorithm is consistent with $\mathcal{M}$ if, for any admissible reward vector $\mu \in \mathcal{M}$ and any parameter $\alpha \in (0, 1]$, the regret of that algorithm is asymptotically negligible compared to T^α, i.e., $\sup_{\alpha \in (0,1)} \limsup_{T \to \infty} \frac{\mathbb{E}_\mu[R_T]}{T^\alpha} = 0$. Graves and Lai [63] proved that any algorithm consistent with $\mathcal{M}$ has a regret scaling at least logarithmically in

T, with a leading constant c_μ depending on μ (and $\mathcal{M}$) i.e., $\liminf_{T\to\infty} \frac{\mathbb{E}_\mu[R_T]}{\log(T)} \geq c_\mu$; moreover, c_μ is the solution of some auxiliary optimization problem. In our setting, it rewrites as

$$c_\mu = \min_{N\geq 0} \sum_{m,k} N_k^m \Delta_{m,k} \quad \text{subject to} \quad \sum_{m,k} N_k^m (\mu_k^m - \lambda_k^m)^2 \geq 2, \forall\, \lambda \in \Lambda(\mu)\,,$$

where $\Lambda(\mu) = \left\{\lambda \in \mathcal{M}; \mu_1^1 = \lambda_1^1, \lambda_1^1 < \max_{m,k} \lambda_k^m\right\}$. We point out that the assumption of dominance is hidden in the class of bandit problem $\mathcal{M}$. Moreover, the square arises in the previous equation due to the KL divergence of the standard Gaussian distribution. In the remaining and with a slight abuse of notation, we are going to call an algorithm consistent with a dominance assumption if it is consistent with the set of all possible vectors of means satisfying this dominance assumption.

GROUP-SPARSE DOMINANCE In this case, the above optimization problem has a closed-form solution.

Theorem 3.1. *An algorithm consistent with the group-sparse dominance satisfies*

$$c_\mu = \sum_{k=2}^{K} \frac{2}{\Delta_{1,k}}\,.$$

Proof. The set $\Lambda(\mu)$ in the optimization problem can be decomposed into $\Lambda(\mu) = \Lambda_k(\mu) \sqcup \cdots \sqcup \Lambda_K(\mu)$ where $\Lambda_k(\mu)$ is the set of alternative parameters in which arm k of category 1 is optimal. Indeed, as we know that $\lambda_1^1 = \mu_1^1 > 0$, the best category is known and the regret incurred by suboptimal categories is non-existent. Thus, asymptotically, we fall back into deriving a lower bound on the regret in one category, i.e. in the classic multi-armed bandit setting. □

This lower bound indicates that all arms in the optimal category (and only those) should be pulled a logarithmic number of times, hence the regret should only scale asymptotically linearly in the number of arms in the optimal category instead of linearly with the total number of arms. We want to stress out here that Theorem 3.1 might have a misleading interpretation. Although the asymptotic regret scales with K and independently of M, the finite-stage minimax regret is still of the order of $\sqrt{MKT}$, as with usual bandits. This is simply because the lower-bound proof [27] of the standard multi-armed bandit case uses set of parameters of the form $(0, \ldots, 0, \varepsilon, 0, \ldots, 0)$ which respect the group-sparse assumption. As a result, the asymptotic lower bound of Theorem 3.1 is hiding some finite-time dependency in MK (possibly of the form of an extra-term in $\sum_{m,k} 1/\Delta_{m,k}$, yet independent of $\log(T)$) that non-asymptotic algorithms[3] would not be able to remove.

STRONG DOMINANCE In the case of strong dominance, a similar closed-form expression can be stated.

[3] We call an algorithm non-asymptotic if its worst-case regret is of the order of $\sqrt{MKT}$, maybe up to some additional polynomial dependency in M and K. In particular, classic algorithms for structured bandits [41, 99] are only asymptotical.

Theorem 3.2. *With strong dominance, a consistent algorithm verifies*

$$c_\mu = \sum_{k=2}^{K} \frac{2}{\Delta_{1,k}} + \sum_{m=2}^{M} \frac{2}{\Delta_{m,K}} \,.$$

Proof. Without loss of generality, we assume that we have $M = 2$ categories and that category 2 has a unique worst arm. The condition in the optimization problem can be written as

$$\sum_{k=2}^{K} N_k^1 (\mu_k^1 - \lambda_k^2)^2 + \sum_{k=1}^{K} N_k^2 (\mu_k^2 - \lambda_k^2)^2 \geq 2, \forall\, \lambda \in \Lambda(\mu) \,,$$

where $\Lambda(\mu) = \Lambda_2(\mu) \sqcup \cdots \sqcup \Lambda_K(\mu) \sqcup \Lambda^2(\mu)$ where $\Lambda_k(\mu)$ is the event in which the best arm is mistaken by arm k in the category 1, i.e.

$$\Lambda_k(\mu) = \{\mu_1^1\} \times]-\infty, \mu_1^1[\times \ldots \times]\mu_1^1, +\infty[\times \ldots \times]-\infty, \mu_1^1[\times]-\infty, \mu_1^1[\times \ldots \times]-\infty, \mu_1^1[$$

and $\Lambda^2(\mu)$ is the event in which we mistake category 2 as the optimal category, i.e.

$$\Lambda^2(\mu) = \{\mu_1^1\} \times]-\infty, \mu_1^1[\times \ldots \times]-\infty, \mu_1^1[\times]\mu_1^1, +\infty[\times \ldots \times]\mu_1^1, +\infty[\,.$$

On $\Lambda_k(\mu)$, the condition is equivalent to

$$N_k^1 \big(\mu_1^1 - \mu_k^2\big)^2 \geq 2 \,,$$

and on $\Lambda^2(\mu)$,

$$\sum_{k=1}^{K} N_k^2 \big(\mu_1^1 - \mu_k^2\big)^2 \geq 2 \,.$$

The minimization problem can thus be separated in two parts: the first part corresponds to finding the best arm in the optimal category and the second part to finding the optimal category.

For the first part, the solution is the same as in the multi-armed bandit setting and is given by $N_k^1 = \frac{2}{\big(\Delta_{1,k}\big)^2}$.

For the second part, let us prove that the solution is given by $N_K^2 = \frac{2}{\big(\Delta_{2,K}\big)^2}$ and $N_k^2 = 0$ for $k \neq K$. We have the following problem

$$\min_{N^2 \geq 0} \sum_{k=1}^{K} N_k^2 \Delta_{2,k} =: f(N^2) \qquad \text{subject to } \sum_{k=1}^{K} N_k^2 (\Delta_{2,k})^2 \geq 2 \,.$$

On one side, we have

$$\min_{N \geq 0} f(N) \leq \min_{n \geq 0} f(0, \ldots, 0, n) = f\left(0, \ldots, 0, \frac{2}{\big(\Delta_{2,K}\big)^2}\right) = \frac{2}{\Delta_{2,K}} \,,$$

and on the other side, since $\Delta_{2,k} < \Delta_{2,K}$, we have

$$\sum_{k=1}^{K} N_k^2 \Delta_{2,k} > \frac{1}{\Delta_{2,K}} \sum_{k=1}^{K} N_k^2 (\Delta_{2,k})^2 \geq \frac{2}{\Delta_{2,K}} \,.$$

Hence the solution of the optimization problem in the suboptimal category and the lower bound on the regret follows. □

This lower bound indicates that the dominance assumption can be leveraged to replace the asymptotic linear dependency in the total number of arms into a linear dependency in the number of arms of the optimal category plus the number of categories. With M categories of K arms each, the dependency in MK is replaced into $M + K$. However, as before and for the same reasons, the finite-time minimax lower bound will still be of the order $\sqrt{MKT}$. The lower bound of Theorem 3.2 seems to indicate that an optimal algorithm should be pulling only the arms of the optimal category and the **worst** arm (not the best!) of the other categories, at least asymptotically and logarithmically. Yet again, there is no guarantee that non-asymptotic algorithms can achieve this highly-demanding (and rather counter-intuitive) lower bound.

First-order dominance There are no simple closed form expression of c_μ with the first-order dominance assumption. We nonetheless provide a variational expression. By simplifying the optimization problem, we obtain the two following conditions

$$\forall\, k \neq 1, N_k^1 (\Delta_{1,k})^2 \geq 2 \,,$$

and $\forall\, k \in [K]$,

$$\begin{aligned} &\sum_{i=1}^{k-1} \Big[\Big(N_{i+1}^1 \big(\mu_{i+1}^1 - \widetilde{\mu}_i\big)^2 + N_i^2 \big(\mu_i^2 - \widetilde{\mu}_i\big)^2 \Big) \mathbf{1}\big\{\mu_i^2 < \mu_{i+1}^1\big\} \Big] + N_k^2 (\Delta_{2,k})^2 \\ &+ \sum_{j=k+1}^{K} \Big(N_j^1 \big(\mu_j^1 - \overline{\mu}_j\big)^2 + N_j^2 \big(\mu_j^2 - \overline{\mu}_j\big)^2 \Big) \geq 2 \,, \end{aligned} \tag{3.1}$$

where $\widetilde{\mu}_i = \dfrac{N_{i+1}^1 \mu_{i+1}^1 + N_i^2 \mu_i^2}{N_{i+1}^1 + N_i^2}$ and $\overline{\mu}_j = \dfrac{N_j^1 \mu_j^1 + N_j^2 \mu_j^2}{N_j^1 + N_j^2}$.

However, for the sake of illustration, we provide a closed-form solution for a specific case.

Theorem 3.3. *With first-order dominance and $M = K = 2$ and assuming that arms are intertwined, [4] i.e. $\mu_1^1 > \mu_1^2 > \mu_2^1 > \mu_2^2$, a consistent algorithm satisfies*

$$c_\mu = \frac{2}{\Delta_{1,2}} + \frac{2}{\Delta_{2,2}} + \frac{2}{\Delta_{2,1}} \left(1 - \frac{(\Delta_{2,2} - \Delta_{1,2})^2}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2} \right).$$

[4] With $K = M = 2$, if arms are not intertwined, then the strong assumption actually holds.

Proof. Assuming the arms are intertwined, the first term in the Equation (3.1) disappears since the condition in the indicator function is not verified. In the case of $M = 2$ categories and two arms per category $K = 2$, the following conditions are derived

$$N_2^1 \geq \frac{2}{(\Delta_{1,2})^2}, \qquad N_2^2 \geq \frac{2}{(\Delta_{2,2})^2},$$

and

$$N_1^2(\Delta_{2,1})^2 + N_2^1\left(\mu_2^1 - \overline{\mu}\right)^2 + N_2^2\left(\mu_2^2 - \overline{\mu}\right)^2 \geq 2,$$

where $\overline{\mu} = \dfrac{N_2^1\mu_2^1 + N_2^2\mu_2^2}{N_2^1 + N_2^2}$.

Since this is a minimization problem, it is clear that the regret is minimize on the lower bounds of N_2^1 and N_2^2. Putting this two quantities in the last inequality, we obtain

$$N_1^2 \geq \frac{2}{(\Delta_{2,1})^2}\left[1 - \left(\left(\frac{\mu_2^1 - \overline{\mu}}{\Delta_{1,2}}\right)^2 + \left(\frac{\mu_2^2 - \overline{\mu}}{\Delta_{2,2}}\right)^2\right)\right]. \tag{3.2}$$

Developing $\overline{\mu}$, we have

$$\overline{\mu} = \frac{\frac{2\mu_2^1}{(\Delta_{1,2})^2} + \frac{2\mu_2^2}{(\Delta_{2,2})^2}}{\frac{2}{(\Delta_{1,2})^2} + \frac{2}{(\Delta_{2,2})^2}} = \frac{\mu_2^1(\Delta_{2,2})^2 + \mu_2^2(\Delta_{1,2})^2}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2}.$$

Now developing $\frac{\mu_2^1 - \overline{\mu}}{\Delta_{1,2}}$, we get

$$\frac{\mu_2^1 - \overline{\mu}}{\Delta_{1,2}} = \frac{\Delta_{1,2}\left(\mu_2^1 - \mu_2^2\right)}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2} = \frac{\Delta_{1,2}\Delta_{2,2}^{1,2}}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2}.$$

Similarly,

$$\frac{\mu_2^2 - \overline{\mu}}{\Delta_{2,2}} = -\frac{\Delta_{2,2}\Delta_{2,2}^{1,2}}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2}.$$

Plugging this into Equation (3.2), we obtain

$$N_1^2 \geq \frac{2}{(\Delta_{2,1})^2}\left[1 - \frac{\left(\Delta_{2,2}^{1,2}\right)^2}{(\Delta_{1,2})^2 + (\Delta_{2,2})^2}\right].$$

The result follows by the decomposition of the expected regret. □

It is quite interesting to compare this lower bound to the corresponding ones with group-sparsity where $c_\mu = \frac{2}{\Delta_{1,2}}$, with strong dominance where $c_\mu = \frac{2}{\Delta_{1,2}} + \frac{2}{\Delta_{2,2}}$ and without structure at all where $c_\mu = \frac{2}{\Delta_{1,2}} + \frac{2}{\Delta_{2,2}} + \frac{2}{\Delta_{2,1}}$. Clearly, lower bounds are, as expected, decreasing with additional structure. More interestingly, the first-order lower bound somehow interpolates be-

tween this two by multiplying the term $\frac{2}{\Delta_{2,1}}$ by a factor $\rho \in (0,1)$; $\rho = 0$ corresponding to the stronger assumption of strong dominance and $\rho = 1$ to the absence of dominance assumption.

3.6 Algorithms and upper bounds

We introduce in this section two algorithms developed for the categorized multi-armed bandit problem. The first one is based on the principle of optimism in the face of uncertainty [9] while the second one is a variant of Thompson Sampling [137]. Regret upper bounds are given in each dominance hypothesis for the first algorithm.

3.6.1 Optimism principle

Our first algorithm is based on the principle of optimism in the face of uncertainty and is summarized in Algorithm 3. It behaves in three different ways depending on the number of categories that are called "active". The definition of an active category will depend on the assumption of dominance. Formally, let $\delta \in (0,1)$ be a confidence level (fixing the confidence level actually requires that the horizon T is known, but there exist well understood anytime version of all these results [48]). At time step t, it computes the set of active categories, denoted $\mathcal{C}(t,\delta)$. The three states of Algorithm 3 are then as follows:

Algorithm 3: CatSE(δ)

```
Pull each arm once
while t ≤ T do
    Compute set of active categories C(t, δ)
    if |C(t, δ)| = 0 then
        Pull all arms
    else if |C(t, δ)| = 1 then
        Perform UCB(δ) in the active category
    else
        Pull all arms in active categories
    end
end
```

1. $|\mathcal{C}(t,\delta)| = 0$: no category is active; the algorithm pulls all arms.
2. $|\mathcal{C}(t,\delta)| = 1$: only one category is active; the algorithm performs UCB(δ) in it.
3. $|\mathcal{C}(t,\delta)| > 1$: several categories are active; the algorithm pulls all arms inside those.

We now detail what we called an active category for each notion of dominance defined previously along with theorems upper bounding the regret of the CatSE algorithm.

GROUP-SPARSE DOMINANCE Under this assumption, we say a category is active if it has an active arm. Following the idea of sparse bandits [92] or bounded regret [29], we say that the arm k of category m is active if

$$\widehat{\mu}_k^m(t) := \frac{\sum_{s\leq t;(C_s,A_s)=(m,k)} X_s}{N_k^m(t)} \geq 2\sqrt{\frac{\log N_k^m(t)}{N_k^m(t)}}\,.$$

This condition ensures that the expected number of times an arm with positive mean is non active is finite. Similarly, the expected number of times an arm with non positive mean is active is also finite. Hence, the expected number of times a suboptimal category is pulled is also finite. Then, the set of active categories, denoted $\mathcal{C}(t)$ is simply

$$\mathcal{C}(t) := \left\{ m \in [M]; \exists\, k \in [K], \widehat{\mu}_k^m(t) \geq 2\sqrt{\frac{\log N_k^m(t)}{N_k^m(t)}} \right\}.$$

Theorem 3.4. *In the group-sparse dominance setting, the expected regret of* CATSE *verifies with probability at least* $1 - 2\delta KT$,

$$\mathbb{E}[R_T] \leq \sum_{k=2}^{K} \frac{8\log\frac{1}{\delta}}{\Delta_{1,k}} + \sum_{m,k} \Delta_{m,k} + \frac{40}{(\mu_1^1)^2}\log\frac{16}{(\mu_1^1)^2}\sum_{m,k}\Delta_{m,k} + (M-1)K\frac{\pi^2}{6}\sum_{m,k}\Delta_{m,k}\,.$$

The first term is the bound of the UCB algorithm while the third term is the regret incurred when the optimal category is non active and the last term comes from a suboptimal category being active. As a result, CATSE is asymptotically optimal, up to a multiplicative factor.

Proof. Consider the following clean event

$$E_1 = \left\{ \forall\, t \in [T], \forall\, k \in [K], |\widehat{\mu}_k^1(t) - \mu_k^1| \leq \sqrt{\frac{2\log\frac{1}{\delta}}{N_k^1(t)}} \right\}.$$

Using union bounds over t and k, one obtains thanks to the subgaussian assumption that $\mathbb{P}(E_1) \geq 2\delta KT$. In the following, we assume the clean event holds true. In the case in which only the optimal category is active, we get the regret of the UCB algorithm

$$\mathbb{E}[R_T] \leq \sum_{k=2}^{K} \frac{8\log\frac{1}{\delta}}{\Delta_{1,k}}\,.$$

On the other hand, the set of active categories is empty if the optimal category is non active. That means that $\forall\, k \leq s, \widehat{\mu}_k^1(N_k^1(t)) < 2\sqrt{\frac{\log N_k^1(t)}{N_k^1(t)}}$ where s is the number of arms with positive expected reward. Let E_2 denote this event. The number of times it happen is bounded. Indeed, since

$$E_2 \subseteq \left\{ \widehat{\mu}_1^1(N_1^1(t)) < 2\sqrt{\frac{\log N_1^1(t)}{N_1^1(t)}} \right\} =: E_3 \,,$$

and

$$n \geq 3 + \frac{32}{(\mu_1^1)^2} \log \frac{16}{(\mu_1^1)^2} \Rightarrow 2\sqrt{\frac{\log n}{n}} - \mu_1^1 \leq -\frac{\mu_1^1}{2} \,,$$

we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=MK+1}^{T} \mathbf{1}\{E_2\} \right] &\leq \mathbb{E}\left[\sum_{t=MK+1}^{T} \mathbf{1}\{E_3\} \right] \\
&\leq \left(3 + \frac{32}{(\mu_1^1)^2} \log \frac{16}{(\mu_1^1)^2} \right) + \sum_{u=1}^{T} \mathbb{P}\left(\widehat{\mu}_1^1(u) - \mu_1^1 < -\frac{\mu_k^1}{2} \right) \\
&\leq \left(3 + \frac{32}{(\mu_1^1)^2} \log \frac{16}{(\mu_1^1)^2} \right) + \sum_{u=1}^{T} \exp\left(-\frac{u}{8} (\mu_1^1)^2 \right) \\
&\leq 3 + \frac{32}{(\mu_1^1)^2} \log \frac{16}{(\mu_1^1)^2} + \frac{8}{(\mu_1^1)^2} \,.
\end{aligned}$$

Finally, the set of active categories has more than one element if a suboptimal category is active, i.e. $\exists\, m \neq 1, \exists\, k \in [K]; \widehat{\mu}_k^m(N_k^m(t)) \geq 2\sqrt{\frac{\log N_k^m(t)}{N_k^m(t)}}$. Let E_4 denote this event. The number of times it happen is also bounded. Indeed,

$$\begin{aligned}
\mathbb{E} \sum_{t=1}^{T} \mathbf{1}\{E_4\} &\leq \sum_{m,k} \sum_{u=1}^{T} \mathbb{P}\left(\widehat{\mu}_k^m(u) \geq 2\sqrt{\frac{\log u}{u}} \right) \\
&\leq \sum_{m,k} \sum_{u=1}^{T} \mathbb{P}\left(\widehat{\mu}_k^m(u) - \mu_k^m \geq 2\sqrt{\frac{\log u}{u}} \right) \\
&\leq \sum_{m,k} \sum_{u=1}^{T} \frac{1}{u^2} \leq (M-1)K \frac{\pi^2}{6} \,.
\end{aligned}$$

Combining the three inequalities, we conclude. □

A trick to improve empirically the performance of the algorithm is to replace the round-robin sampling phase (when $|\mathcal{C}(t)| = 0$) by choosing an arm with a higher probability the closer it is to be active. This idea was analyzed in Bubeck, Perchet, and Rigollet [29] with additional assumptions. Yet this can only improve the second term of the regret, which is already constant with respect to the time horizon T (so we chose to not focus on it). For example, a possibility is to pull arm

(m,k) at time t with probability [5] $p_k^m(t) \propto \left(\sqrt{\frac{4\log N_k^m(t)}{N_k^m(t)}} - \widehat{\mu}_k^m(t)\right)^{-2}$. Another possible improvement is to eliminate categories in which there exist an arm whose upper bound is less than 0. Again, this only improves a term constant with respect to T.

STRONG DOMINANCE In this case, CATSE will use the information gathered by all arms. The overall idea is to construct a confidence region for the mean vector and to eliminate a category as soon as it is clearly dominated by another one. The statistical test to perform in order to determine which categories to eliminate is based on the following alternative characterization of dominance.

Let $\mathcal{S}(K) \coloneqq \{\mathbf{x} \in \mathbb{R}_+^K; \|\mathbf{x}\|_1 = 1\}$ be the K-simplex and $\mu^m \coloneqq (\mu_k^m)_k$ denote the vector of means of category m.

Proposition 3.1. $\mathbf{C}_1$ strongly dominates $\mathbf{C}_2$ if and only if

$$\forall\, \mathbf{x} \in \mathcal{S}(K), \forall\, \mathbf{y} \in \mathcal{S}(K), \langle \mathbf{x}, \mu^1 \rangle \geq \langle \mathbf{y}, \mu^2 \rangle\,.$$

Proof. Let $(e_i)_i$ denotes the unit vectors. Taking $\mathbf{x} = e_k$ and $\mathbf{y} = e_l$ hands $\mu_k^1 \geq \mu_l^2$.

In the other direction, let $(\alpha, \beta) \in \mathcal{S}(K) \times \mathcal{S}(K)$. We have

$$\langle \alpha, \mu \rangle = \sum_{k=1}^{K} \alpha_k \mu_k = \sum_{k=1}^{K-1} \alpha_k \mu_k + \left(1 - \sum_{k=1}^{K-1} \alpha_k\right)\mu_K = \mu_K + \sum_{k=1}^{K-1} \alpha_k(\mu_k - \mu_K)\,.$$

Now, using the previous equality, we obtain

$$\begin{aligned}
\langle \alpha, \mu^1 \rangle - \langle \beta, \mu^2 \rangle &= \sum_{k=1}^{K} \alpha_k \mu_k^1 - \sum_{k=1}^{K} \beta_k \mu_k^2 \\
&= (\mu_K^1 - \mu_1^2) + \sum_{k=1}^{K-1} \alpha_k(\mu_k^1 - \mu_K^1) + \sum_{k=2}^{K} \beta_k(\mu_1^2 - \mu_k^2) \\
&\geq 0\,.
\end{aligned}$$

□

At the end of the p-th round of the phase of successive elimination of categories, each arm has been pulled p times. A natural estimator of $\mu^m \in \mathbb{R}^K$ is the coordinate wise empirical average of rewards, i.e., $\mu_k^m(p) = \frac{1}{p}\sum_{r=1}^{p} X_k^m(r)$, where (with a slight abuse of notation), $X_k^m(r)$ is the reward gathered by the r-th pull of arm k of category m. We now describe the statistical run at the end of round $p \in \mathbb{N}$; category $n \in [M]$ is eliminated by category $m \in [M]$ if it holds that

$$\begin{aligned}
L_m^+(p,\delta) &\coloneqq \max_{\mathbf{x}\in\mathcal{S}(K)} \langle \mathbf{x}, \widehat{\mu}^m(p) \rangle - \|\mathbf{x}\|_2 \beta(p,\delta) \\
&> \min_{\mathbf{y}\in\mathcal{S}(K)} \langle \mathbf{y}, \widehat{\mu}^n(p) \rangle + \|\mathbf{y}\|_2 \beta(p,\delta) =: L_n^-(p,\delta)\,,
\end{aligned} \tag{3.3}$$

[5] Other potential functions may lead to improvement.

where $\beta(p,\delta) = \sqrt{\frac{2}{p}\big(K\log 2 + \log\frac{1}{\delta}\big)}$. The set of active categories is then define as follows

$$\mathcal{C}(t,\delta) = \big\{m \in [M]; \forall\, n \neq m, L_n^+(t,\delta) \leq L_m^-(t,\delta)\big\}\,.$$

Theorem 3.5. *In the strong dominance case, the regret of* CatSE *satisfies w.p. at least* $1-\delta MT$,

$$R_T \leq \sum_{k=2}^{K} \frac{8\log\frac{1}{\delta}}{\Delta_{1,k}} + \sum_{m,k}\Delta_{m,k} + 8\big(K\log 2 + \log\frac{1}{\delta}\big)\sum_{m=2}^{M}\min_{\mathbf{x},\mathbf{y}\in\mathcal{S}(K)}\Big(\frac{\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2}{\langle \mathbf{x},\mu^1\rangle - \langle \mathbf{y},\mu^m\rangle}\Big)^2 \sum_{k=1}^{K}\Delta_{m,k}$$

To prove this theorem, we need the following Lemma.

Lemma 3.2. *With probability at least* $1-\delta$, *the following holds uniformly overall all* $\mathbf{x} \in \mathbb{R}^K$,

$$\langle \mathbf{x}, \widehat{\mu}^m(p) - \mu^m\rangle \leq \|\mathbf{x}\|_2\sqrt{\frac{2}{p}\left(K\log 2 + \log\frac{1}{\delta}\right)}\,.$$

Proof. Fix $x \in \mathbb{R}$ and $\delta \in (0,1)$ a confidence level. According to Lattimore and Szepesvári [100], we have with probability at least $1-\delta$,

$$\|\widehat{\mu}(t) - \mu\|_{V_t} \leq \sqrt{2\left(K\log 2 + \log\frac{1}{\delta}\right)}\,.$$

If an agent pulls each arm sequentially, we are in the fixed design setting. In this case, (assuming t is a multiple of K), we have $V_t = N(t)\mathbf{I}_K$, i.e. it is a diagonal matrix and we conclude. □

We are now ready to prove the theorem.

Proof. Let E_0 denote the clean event

$$E_0 = \big\{\forall\, t \in [T]; \forall\, m \in [M], \forall \mathbf{x} \in \mathbb{R}^K, \langle \mathbf{x}, \widehat{\mu}^m(t) - \mu^m\rangle \leq \|\mathbf{x}\|_2\beta(t,\delta)\big\}\,,$$

where $\beta(t,\delta) = \sqrt{\frac{2}{N^m(t)}\big(K\log 2 + \log\frac{1}{\delta}\big)}$.

Using union bounds over the time and the categories, and by Lemma 3.2, we obtain $\mathbb{P}(E_0^c) \leq \delta MT$.

Suppose we are in the clean event and let $m \neq 1$ be a suboptimal category and t be the last time when we did not invoke the stopping rule, i.e. that the category m is still active. First remark that category 1 is never eliminated by category m on the clean event since $\min_k \mu_k^1 \geq \max_k \mu_k^m$. By Equation (3.3), this means that $\forall\, \mathbf{x} \in \mathcal{S}(K), \forall\, \mathbf{y} \in \mathcal{S}(K)$,

$$\langle \mathbf{x}, \widehat{\mu}^1(t)\rangle - \langle \mathbf{y}, \widehat{\mu}^m(t)\rangle \leq (\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2)\sqrt{\frac{2}{N(t)}\left(\log\frac{1}{\delta} + K\log 2\right)}\,,$$

where $N(t)$ denotes the number of times each category have been pulled. As we are in the clean event, we have $\forall \mathbf{x} \in \mathcal{S}(K), \forall \mathbf{y} \in \mathcal{S}(K)$,

$$\langle \mathbf{x}, \mu^1 \rangle - \langle \mathbf{y}, \mu^m \rangle \leq 2(\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2) \sqrt{\frac{2}{N(t)} \left(\log \frac{1}{\delta} + K \log 2 \right)} \,.$$

Inverting this equation, we obtain the following upper bound on $N(t)$

$$\forall \mathbf{x} \in \mathcal{S}(K), \forall \mathbf{y} \in \mathcal{S}(K), N(t) \leq 8 \left(K \log 2 + \log \frac{1}{\delta} \right) \left(\frac{\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2}{\langle \mathbf{x}, \mu^1 \rangle - \langle \mathbf{y}, \mu^m \rangle} \right)^2 .$$

The proof is conclude with the analysis of the UCB algorithm [9]. □

First-order dominance CatSE will proceed with first-order dominance as with strong dominance, the major difference is the statistical test. Let us first characterize the notion of first-order dominance.

Proposition 3.2. $\mathbf{C}_1$ first-order dominates $\mathbf{C}_2$ if and only if

$$\forall \mathbf{x} \in \mathcal{S}(K), \langle \mathbf{x}, \mu^1 \rangle \geq \langle \mathbf{x}, \mu^2 \rangle \,.$$

Proof. Taking $\mathbf{x} = e_k$ hands $\mu_k^1 \geq \mu_k^2$. In the other direction, let $\mathbf{x} \in \mathcal{S}(K)$. We have

$$\langle \mathbf{x}, \mu^1 - \mu^2 \rangle = \sum_{k=1}^{K} \mathbf{x}_k (\mu_k^1 - \mu_k^2) \geq 0 \,.$$

□

The statistical test is then defined as follows: category $n \in [M]$ is eliminated by category $m \in [M]$ at round p if

$$D_{m,n}(p, \delta) := \max_{\mathbf{x} \in \mathcal{S}(K)} \frac{\langle \mathbf{x}, \widehat{\mu}_\sigma^m(p) - \widehat{\mu}_\tau^n(p) \rangle}{\|\mathbf{x}\|_2} > 2\gamma(p, \delta) \,, \tag{3.4}$$

where $\widehat{\mu}_\sigma^m(p)$ and $\widehat{\mu}_\tau^n(p)$ represent respectively the reordering of $\widehat{\mu}^m(p)$ and $\widehat{\mu}^n(p)$ in decreasing order and $\gamma(p, \delta) = \frac{1}{\sqrt{2p}} \left(\sqrt{K \log \frac{1}{\delta}} + \sqrt{1 + (K+1) \log K} \right)$. We emphasis the permutation is specific to both a category and a round. This statistical test yields the following set of active categories

$$\mathcal{C}(t, \delta) = \{ m \in [M]; \forall\, n \neq m, D_{m,n}(t, \delta) \leq 2\gamma(t, \delta) \} \,.$$

Theorem 3.6. *Under the additional assumption that $X_t \in [0,1]$, in the first-order dominance, the regret of CatSE verifies with probability at least $1-\delta MT$,*

$$R_T \leq \sum_{k=2}^{K} \frac{8\log\frac{1}{\delta}}{\Delta_{1,k}} + \sum_{m,k} \Delta_{m,k} + 16\left(K\log\frac{1}{\delta} + K\log K + \log K + 1\right)\sum_{m=2}^{M} \frac{\sum_{k=1}^{K}\Delta_{m,k}}{\|\mu^1-\mu^m\|_2^2}\,.$$

To prove this result we need the following Lemma.

Lemma 3.3. *With probability at least $1-\delta$,*

$$\|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\|_2 \leq \frac{1}{\sqrt{2t}}\left(\sqrt{K\log\frac{1}{\delta}} + \sqrt{1+(K+1)\log K}\right),$$

where $\widehat{\mu}^m_{\sigma^m_t}(t)$ denotes the vector $\widehat{\mu}^m(t)$ ordered in decreasing order.

Proof. The McDiarmid inequality gives the following

$$\mathbb{P}\Big\{\|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\| \geq \mathbb{E}\|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\| + \varepsilon\Big\} \leq \exp(-2t\varepsilon^2/K)\,.$$

Now we just have to bound $\mathbb{E}\|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\|_2$. If $Y_1,\dots,Y_N$ are σ^2-subgaussian, then

$$\mathbb{P}\left\{\max_{i=1,\dots,N} Y_i \geq \varepsilon\right\} \leq N\exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right).$$

This give, by a careful integration, that

$$\mathbb{E}\left(\max_{i=1,\dots,N} Y_i\right)^2 \leq 2\sigma^2(\log(N)+1)\,.$$

In our case, we have $\sigma^2 = \frac{1}{4t}$. Using that the expectation of the k^{th} maximum of N random variables is smaller than the expectation of the maximum of $N-(k-1)$ random variables [46], we obtain

$$\begin{aligned}\mathbb{E}\|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\|_2^2 &\leq \frac{1}{2t}\sum_{k=1}^{K}(1+\log(K-(k-1))) \\ &\leq \frac{1}{2t}(K+\log K!) \\ &\leq \frac{1+(K+1)\log K}{2t}\,,\end{aligned}$$

where the last inequality comes from the Stirling formula. The result follows. □

Proof. Let define the clean event

$$E_1 = \Bigg\{ \forall\, t \in [T], \forall\, m \in [m], \\ \|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\|_2 \leq \frac{1}{\sqrt{2t}} \left(\sqrt{K \log \frac{1}{\delta}} + \sqrt{1 + (K+1)\log K} \right) \Bigg\}$$

By Lemma 3.3 and with union bounds over t and m, we have $\mathbb{P}(E_1^c) \leq \delta MT$. Let $m \neq 1$ and t be the last time we pulled category m.

By Equation (3.4), we have

$$\forall\, \mathbf{x} \in \mathcal{S}(K), \langle \mathbf{x}, \widehat{\mu}^1_{\sigma^1_t}(t) - \widehat{\mu}^m_{\sigma^m_t}(t) \rangle \leq 2\|\mathbf{x}\|_2 \gamma(t, \delta)\,.$$

Moreover, notice that after t samples, $\forall\, \mathbf{x} \in \mathcal{S}(K)$,

$$\begin{aligned} \frac{1}{\|\mathbf{x}\|_2} \left| \langle \mathbf{x}, \widehat{\mu}^1_{\sigma^1_t}(t) - \widehat{\mu}^m_{\sigma^m_t}(t) \rangle - \langle \mathbf{x}, \mu^1 - \mu^m \rangle \right| &\leq \|\widehat{\mu}^1_{\sigma^1_t}(t) - \mu^1\|_2 + \|\widehat{\mu}^m_{\sigma^m_t}(t) - \mu^m\|_2 \\ &\leq 2\gamma(t, \delta)\,, \end{aligned}$$

where the last inequality holds true with probability at least $1 - \delta MT$. Combining the two inequalities, one obtains with probability at least $1 - \delta MT$,

$$\begin{aligned} N^m(t) &\leq \frac{8}{\|\mu^1 - \mu^m\|_2^2} \left(\sqrt{K \log \frac{1}{\delta}} + \sqrt{1 + (K+1)\log K} \right)^2 \\ &\leq \frac{16}{\|\mu^1 - \mu^m\|_2^2} \left(K \log \frac{1}{\delta} + K \log K + \log K + 1 \right) \end{aligned}$$

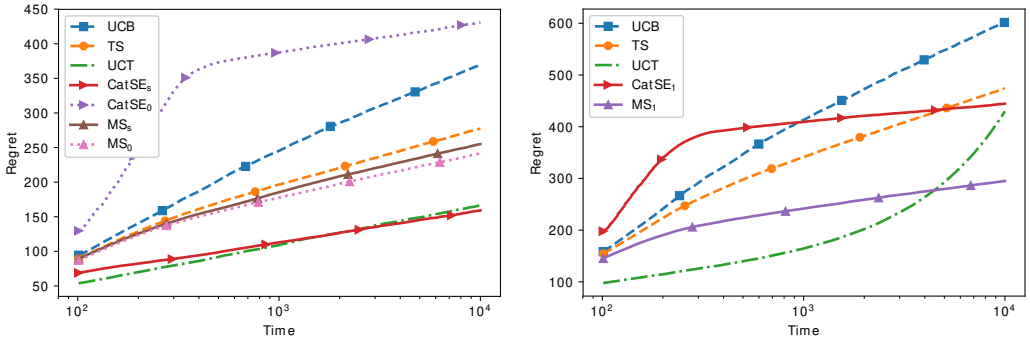
where in the last inequality we used the Cauchy–Schwarz inequality. Hence the result. □

3.6.2 Bayesian principle

Algorithm 4: Murphy Sampling

while $t \leq T$ **do**
  Sample $\theta(t) \sim \Pi_{t-1}(\cdot|\mathcal{H}_d)$
  Pull $(C_t, A_t) \in \arg\max_{(m,k)} \theta^m_k(t)$
end

The Murphy Sampling (MS) algorithm [80] was originally developed in a pure exploration setting. Conceptually, it is derived from Thompson Sampling (TS) [137], the difference is that the sampling respects some inherent structure of the problem. To define MS, we denote by $\mathcal{F}(t) = \sigma(A_1, X_1, \ldots, A_t, X_t)$ the information available after t steps and $\mathcal{H}_d$ the assumption of dominance considered. Let $\Pi_t = \mathbb{P}(\cdot|\mathcal{F}_t)$ be the posterior distribution of the means param-



(a) Sparse and strong dominance scenario

(b) First-order dominance scenario

Figure 3.3: Regret of various algorithms as a function of time.

eters after t rounds. The algorithm samples, at each time step, from the posterior distribution $\Pi_{t-1}(\cdot|\mathcal{H}_d)$ and then pulls the best arm, which, by definition, is in the best category sampled at this time step. In comparison, TS would sample from Π_{t-1} without taking into account any structure. To implement this algorithm, we use that independent conjugate priors will produce independent posteriors, making the posterior sampling tractable. The required assumption, i.e. the structure of our problem, is then attained using rejection sampling. We do not provide theoretical guarantees on its regret but we will illustrate empirically on simulated data that it is highly competitive compared to the other algorithms, as is TS in the standard multi-armed bandit setting [33].

3.7 Experiments

We finally present numerical experiments illustrating the performance of the algorithms we have introduced. We compare them with two families of algorithms. The first one is algorithms for the multi-armed bandit framework, namely UCB [9] and TS [137]; they are agnostic to the structure of the arms. The second family of algorithms is adapted to tree search, namely UCT [87]; they partially take into account the inherent structure. Specifically, they will just use the fact that arms are grouped into categories but not that one category dominates the others. We consider two scenarios for the different dominance hypothesis. In all experiments, rewards are drawn from Gaussian distributions with unit variance and we report the average regret as a function of time, in log-scale. To implement TS and MS, we pulled each arm once and then sampled using a Gaussian prior. The simulations were ran until time horizon $T = 10000$ and results were averaged over 100 independent runs.

Group-sparse & strong dominance We start by grouping the experiments in the group-sparse and strong dominance setting, as we recall that the only difference between the two concepts is the knowledge of a threshold between the best category and the others. In this first scenario, we analyze a problem with five categories and five arms per category. Precisely, in the first category the optimal arm has expected reward 1, and the four suboptimal arms consist of one group of three

(stochastically) identical arms each with expected reward 0.5 and one arm with expected reward 0. The four suboptimal category are identical and are composed of two arms with expected rewards 0 and -1, respectively and a group of three arms with expected reward -0.5. We used the subscript s and 0 to denote the assumption of dominance the algorithm exploited. CatSE$_s$ and CatSE$_0$ were run with $\delta = \frac{1}{t}$ and $\delta = \frac{1}{Mt}$, respectively. Results are presented on Figure 3.3a. In the case of group-sparse dominance, CatSE$_s$ (implement here with the potential sampling improvement) outperforms both UCB and UCT; MS$_s$ asymptotically performs as well yet with a slightly higher regret. Interestingly, UCT performs well in the beginning; thanks to the lack of an exploration phase compared to CatSE$_s$. In the case of strong dominance, MS$_0$ and CatSE$_0$ asymptotically perform alike and slightly better than UCT. However, the regret of CatSE$_0$ is much higher due to its round-robin sampling phase; this can be seen in the beginning as CatSE$_0$ is still in the search of the optimal category. If we compare the two versions of each algorithm between them, we can notice two points. Firstly, for CatSE, the result of the potential sampling improvement is significant. Secondly, for MS, the regret in the group-sparse case is slightly worse than in the strong dominance case even though it is stronger. This is simply due to our implementation and the difficulty of the posterior sampling, specifically the rejection sampling phase.

First-order dominance Finally, we consider the first-order dominance setting. In this scenario, we look upon a problem with five categories and ten arms per category. Precisely, in the optimal category, the best arm has expected reward 5 while the nine suboptimal arms consist of three group of five, three and one arms, with expected rewards 4, 3 and 2, respectively. The four suboptimal categories are composed of two arms with expected rewards 4.5 and 0, respectively, and eight arms with expected reward 3. CatSE was run with $\delta = \frac{1}{Mt}$ and the results are presented on Figure 3.3b. Once again, MS and CatSE outperform baseline algorithms and both appear to have the same slope asymptotically with a significant difference between their regret, again due to the exploration phase of CatSE. It is interesting to observe that UCT performed poorly; as noticed in [43], the convergence can be sluggish. Indeed, the main issue occurs when the best arm is underestimated. In that case, it is pulled a logarithmic number of times the optimal category is pulled, which is a logarithmic number of times, since the second best arm overall is in suboptimal categories. Hence, it would take an exponential of exponentials number of time for the optimal arm to become the best again.

3.8 Conclusion

In this chapter, we have introduced a novel structured bandit framework inspired by e-commerce applications. In our setting, the arms are assumed to belong to ordered categories. We have presented three different relations of dominance between categories and we confirmed the veracity of our model on real data. For each dominance, we derived asymptotic regret lower bound and we devised two generic algorithms to solve the categorized bandit problem.

Two problems remain open: the first one is a better exploration phase in CatSE since it heavily impacts the regret and as noted in [57], ETC algorithms are necessarily suboptimal; and the second is an upper bound on the regret of the MS algorithm since it is highly competitive in practice. We

believe that it is asymptotically optimal and that it can be applied to other setting of structured bandits.

3.9 A suboptimal algorithm in the strong dominance case

In this section, we analyze a natural, yet suboptimal, algorithm in the case of the strong dominance assumption. Indeed, the lower bound stated in Theorem 3.2 seems to indicate that the best way to identify a suboptimal category in this case is by pulling its worst arm. The algorithm in question, which is essentially CatSE with another definition of active category, follows this idea. We call this algorithm MinMaxCatSE. Let $\delta \in (0,1)$ be a confidence level. At each time step t, it computes the best lower bound $C_m^+(t,\delta)$ and the worst upper bound $C_m^-(t,\delta)$ inside each category. Formally, for a category m

$$C_m^+(t,\delta) = \max_{k\in[K]} \widehat{\mu}_k^m(t) - \sqrt{\frac{2\log\left(\frac{1}{\delta}\right)}{N_k^m(t)}}$$
$$C_m^-(t,\delta) = \min_{k\in[K]} \widehat{\mu}_k^m(t) + \sqrt{\frac{2\log\left(\frac{1}{\delta}\right)}{N_k^m(t)}}$$

Then it rejects suboptimal categories if their worst arm is statistically worse than the best arm of another category. Similarly to CatSE, we define $\mathcal{C}(t,\delta)$ the set of active categories at time t as follows

$$\mathcal{C}(t,\delta) = \left\{m \in [M]; \forall n \neq m, C_n^+(t,\delta) \leq C_m^-(t,\delta)\right\}$$

We prove the following theorem bounding its regret.

Theorem 3.7. *With probability at least* $1 - 2\delta MKT$*, the regret of MinMaxCatSE satisfies*

$$\mathbb{E}[R_T] \leq 8\sum_{k=2}^{K} \max\left\{\frac{1}{\Delta_{1,k}}, \frac{4\Delta_{1,k}}{\Delta_{\sharp,K}^2}\right\}\log\frac{1}{\delta} + \sum_{m=1}^{M}\sum_{K=1}^{K}\Delta_{m,k} + \sum_{m=2}^{M}\left(\frac{32\log\frac{1}{\delta}}{(\Delta_{m,K})^2}\left(\sum_{k=1}^{K}\Delta_{m,k}\right)\right)$$

where $\Delta_{\sharp,K} = \min_{m=2,\dots,M}\Delta_{m,K}$.

MinMaxCatSE outplays, theoretically, UCB since $\frac{\Delta_{m,k}}{(\Delta_{m,K})^2} \leq \frac{1}{\Delta_{m,k}}$ ($\Delta_{m,k} \leq \Delta_{m,K}$), so the dependency with respect to the gaps of arms in suboptimal categories is greatly reduced. Its major drawback is that it only uses the information of the best and worst arms in categories to discover suboptimal ones; and thus does not use the information obtained on intermediate arms like CatSE. Notice that the MinMaxCatSE is obtained by choosing $\mathbf{x} = (1,0,\dots,0)$ and $\mathbf{y} = (0,\dots,0,1)$ in Equation (3.3) while CatSE instead optimize over these values. As a result MinMaxCatSE is a special case of CatSE and the latter is always better in the worst-case.

Proof. Let $\mathcal{E}$ be the good event defined by

$$\mathcal{E} = \left\{ \forall m \in [M], \forall k \in [K], \forall u \in [T], |\widehat{\mu}_k^m - \mu_k^m| < \sqrt{\frac{2\log\frac{1}{\delta}}{u}} \right\}.$$

Using the subgaussian assumption and a union bound, one obtains $\mathbb{P}(\mathcal{E}^c) \leq 2\delta MKT$.

On the good event $\mathcal{E}$, the optimal category is never eliminated. Otherwise, we would have

$$\mu_1^m \geq \widehat{\mu}_1^m(t) - \sqrt{\frac{2\log\frac{1}{\delta}}{N_1^m(t)}} > \widehat{\mu}_K^1(t) + \sqrt{\frac{2\log\frac{1}{\delta}}{N_K^1(t)}} \geq \mu_K^1$$

which is impossible by assumption.

Now let us upper bound the number of times a suboptimal category is active. By definition of the algorithm we have to upper bound the number of times we pull its worst arm. Let t be the last time category m is active. At time t, we have

$$\widehat{\mu}_1^1(t) - \sqrt{\frac{2\log\frac{1}{\delta}}{N_1^1(t)}} \leq \widehat{\mu}_K^m(t) + \sqrt{\frac{2\log\frac{1}{\delta}}{N_K^m(t)}}$$

which implies that

$$\Delta_{m,K} \leq 2\left(\sqrt{\frac{2\log\frac{1}{\delta}}{N_1^1(t)}} + \sqrt{\frac{2\log\frac{1}{\delta}}{N_K^m(t)}}\right) = 4\sqrt{\frac{2\log\frac{1}{\delta}}{N_K^m(t)}}$$

where the last equality comes from the fact that $N_1^1(t) = N_K^m(t)$. Hence,

$$N_K^m(t) \leq \frac{32\log\frac{1}{\delta}}{(\Delta_{m,K})^2}$$

The result follows with the classic analysis of the UCB algorithm [9], the max term arising from the fact that UCB pulls more times a suboptimal arm. □

3.10 Full information feedback

In this section, we study heuristics derived under the full information feedback in the categorized bandit model. We recall that with this feedback, the learner observes a reward for each arm. She still does not know their expected rewards. This was intended to serve as a benchmark for the bandit feedback. Thereby, we also make the additional assumption that the reward distributions are Gaussian random variables with unit variance.

3.10.1 Heuristics

We begin by providing algorithms in the full information feedback. This setting allows us to focus on the choice of the category at each time step rather than which arm to choose. Consequently, we use the Follow-The-Leader (FTL) algorithm [48] to choose which arm to pull once the choice of the category is set. We recall that after a uniformly random pull in the first time step, the FTL algorithm pulls the arm with the highest average reward. Hence, our problem is reduced to finding the best category empirically i.e. determining the category which is more likely to satisfy the specified stochastic dominance assumption.

Extension of the FTL algorithm

The first algorithm, called HFTL (the H stands for hierarchical) is summarized in Algorithm 5. It is a generalization of the FTL algorithm to our framework. We know that for two random variables, if X stochastically dominates Y (for any order) then the expectation of X is greater than the expectation of Y. Back to our problem, let K_m be the number of arms of category m, $\overline{\mu}^m := \frac{1}{K_m}\sum_{k\in[K_m]}\mu_k^m$ be the mean of category m and $\widehat{\mu}^m(t)$ be its empirical estimate based on the samples observed until time step t. At time t, the HFTL algorithm pulls category $C_t \in \arg\max_{m\in[M]}\widehat{\mu}^m(t)$ i.e. the HFTL algorithm differentiates categories based on the average reward across all its arms and consequently, aggregates the information of a category to a unique value.

Algorithm 5: HFTL

Choose a category and an arm randomly
for $t = 2, \dots, T$ **do**
  Pull category $C_t \in \arg\max_{m\in[M]}\widehat{\mu}^m(t)$
  Pull arm $A_t \in \arg\max_{k\in[K_{C_t}]}\widehat{\mu}_k^{C_t}(t)$
end

The Greedy Approach

Another point of view of the FTL algorithm is that it is a greedy algorithm: it pulls the arm with the best empirical mean which is none other than the maximum likelihood estimator (MLE) of the expected reward. Let us employ the same principle in our problem. Considering $M = 2$ categories, one has to determinate which one is more likely to be better.

First, we assume that one category is optimal, for instance category $\mathcal{A}$ is better than category $\mathcal{B}$, and we determinate the MLE of the means given our constraints. In the case of Gaussian rewards with unit variance, we must solve the following optimization problem

$$\min_{\mu:\mathcal{A}\succeq\mathcal{B}} \sum_{k\in[K_{\mathcal{A}}]} \left(\mu_k^{\mathcal{A}} - \widehat{\mu}_k^{\mathcal{A}}\right)^2 + \sum_{k\in[K_{\mathcal{B}}]} \left(\mu_k^{\mathcal{B}} - \widehat{\mu}_k^{\mathcal{B}}\right)^2. \tag{3.5}$$

Then we solve again Equation (3.5) but this time assuming that category $\mathcal{B}$ is optimal. Once our two constrained solutions obtained, we evaluate them and our best empirical category will be the one maximizing the likelihood, i.e. the quantity in Equation (3.5).

The problem with this approach is that the solution to Equation (3.5) is, most of the time, intractable. One simple solution can, nonetheless, be obtained in the case of first-order stochastic dominance with the same number of arms in the two categories. Recall that category $\mathcal{A}$ first-order dominates category $\mathcal{B}$ if the k^{th} best arm of $\mathcal{A}$ is better than the k^{th} best arm of $\mathcal{B}$ and the difficulty is reduce to compare one arm with another. Consider the k^{th} best arm empirically of category $\mathcal{A}$ with empirical mean $\widehat{\mu}^{\mathcal{A}}_{(k)}$ and the one of category $\mathcal{B}$ with empirical mean $\widehat{\mu}^{\mathcal{B}}_{(k)}$, we then have two cases: if $\widehat{\mu}^{\mathcal{A}}_{(k)} \geq \widehat{\mu}^{\mathcal{B}}_{(k)}$, the constraint is already satisfied; in the other case, if $\widehat{\mu}^{\mathcal{A}}_{(k)} < \widehat{\mu}^{\mathcal{B}}_{(k)}$, the solution to the constrained problem is simply the mean of the two empirical means: $\mu^{\mathcal{A}}_{(k)} = \mu^{\mathcal{B}}_{(k)} = \frac{\widehat{\mu}^{\mathcal{A}}_{(k)} + \widehat{\mu}^{\mathcal{B}}_{(k)}}{2} =: \overline{\mu}_{(k)}$. Then, the optimal category empirically is simply the category which minimizes the divergence between the constrained and the unconstrained estimations, which in the case of category $\mathcal{A}$ better than $\mathcal{B}$ is denoted by $d_{\mathcal{A}\succeq\mathcal{B}}$ and defined by

$$d_{\mathcal{A}\succeq\mathcal{B}} := \sum_{k\in[K]} \left[\frac{\left(\widehat{\mu}^{\mathcal{A}}_{(k)} - \overline{\mu}_{(k)}\right)^2}{2} + \frac{\left(\widehat{\mu}^{\mathcal{B}}_{(k)} - \overline{\mu}_{(k)}\right)^2}{2} \right] \mathbf{1}\{\widehat{\mu}^{\mathcal{A}}_{(k)} < \widehat{\mu}^{\mathcal{B}}_{(k)}\} \tag{3.6}$$

This algorithm, which we called Greedy1, is summarized in Algorithm 6.

Algorithm 6: Greedy1

Choose a category and an arm randomly
for $t = 2, \dots, T$ **do**
  Compute $d_{1\succeq 2}$ and $d_{2\succeq 1}$ according to (3.6)
  Pull the category minimizing the divergence
  Pull arm $A_t \in \arg\max_{k\in[K_{C_t}]} \widehat{\mu}^{C_t}_k(t)$
end

A pseudo-Greedy Approach

We discern that the Greedy approach chooses the category which is the closest to being optimal by pushing the empirical means of all categories in order to satisfy the constraints. Our idea is to mimic this technique but this time we push only the arms of one category.

In the case of strong dominance, we compute the divergence between the empirical means of the supposed optimal category and the ones of the other category in order to satisfy the constraints. More formally, let $\widetilde{\mu}^{\mathcal{B}} := \max_{k\in[K_{\mathcal{B}}]} \widehat{\mu}^{\mathcal{B}}_k$ be the best empirical mean of category $\mathcal{B}$, we denote by $d^0_{\mathcal{A}\succeq\mathcal{B}}$ the divergence

$$d^0_{\mathcal{A}\succeq\mathcal{B}} := \sum_{k\in[K_{\mathcal{A}}]} \frac{\left(\widetilde{\mu}^{\mathcal{B}} - \widehat{\mu}^{\mathcal{A}}_k\right)^2}{2} \mathbf{1}\{\widehat{\mu}^{\mathcal{A}}_k < \widetilde{\mu}^{\mathcal{B}}\}$$

In the case of first-order dominance with two categories, we already know the hierarchy in the worst case by definition. Hence, for categories with the same number of arms, the divergence is given by

$$d^1_{\mathcal{A}\succeq\mathcal{B}} := \sum_{k\in[K]} \frac{\left(\widehat{\mu}^{\mathcal{B}}_{(k)} - \widehat{\mu}^{\mathcal{A}}_{(k)}\right)^2}{2} \mathbf{1}\{\widehat{\mu}^{\mathcal{B}}_{(k)} > \widehat{\mu}^{\mathcal{A}}_{(k)}\}$$

Our algorithm named pGreedy is then summarized in Algorithm 7.

Algorithm 7: pGreedy-p

Choose a category and an arm randomly
for $t = 2, \dots, T$ **do**
  Compute $d^p_{1\succeq 2}$ and $d^p_{2\succeq 1}$ given dominance p
  Pull the category C_t minimizing the divergence
  Pull arm $A_t \in \arg\max_{k\in[K_{C_t}]} \widehat{\mu}^{C_t}_k(t)$
end

3.10.2 Experiments

We present some numerical experiments of the proposed policies to show their strengths and weaknesses. We compare them with FTL since it does not take into account the structure of the arms.

We consider three scenarios with two categories in which the rewards are Gaussian distributions with unit variance. As previously, we report the regret as a function of time. Results were averaged over 50000 independent runs.

Scenario 1: Strong dominance

In the first scenario, we analyze a simple problem with two arms per category in the case of strong dominance. More precisely, the expected rewards are 0.6 and 0.5 for the first category and 0.4 and 0.3 for the second category. The simulations were ran until time horizon $T = 2500$. Results are presented on Figure 3.4.

In this scenario, HFTL and pGreedy0 have the same regret and perform slightly better than FTL. They ruled out more easily the arms in the sub-optimal category. However, in the case of strong dominance, the main difficulty lie in the optimal category hence the small difference between their regrets.

Scenario 2: First-order dominance

In the second scenario, we analyze the same problem as previously but this time we swap two arms to be in the case of first-order dominance. Specifically, expected rewards are 0.6 and 0.4 for the first category and 0.5 and 0.3 for the second one. The simulations were also run until time horizon $T = 2500$. Results are presented on Figure 3.5.

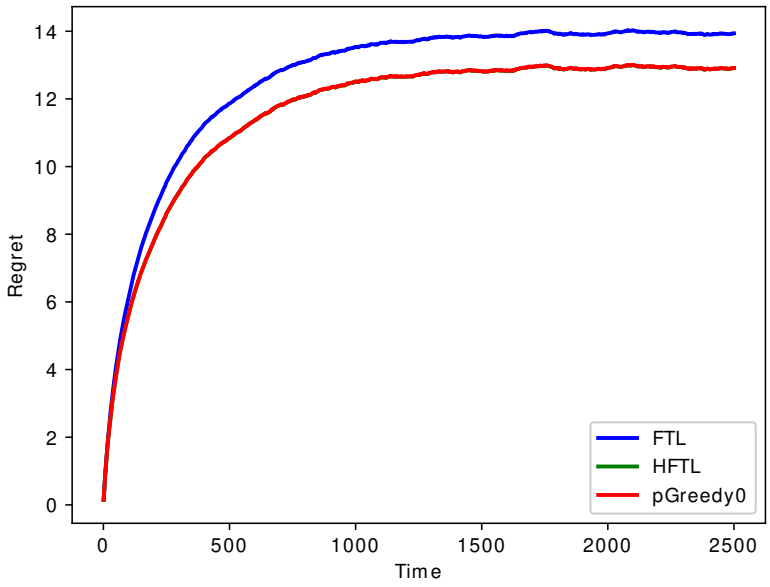


Figure 3.4: Regret of various algorithms as a function of time in the full information framework and strong dominance scenario. Curves of HFTL and pGreedy0 are similar.

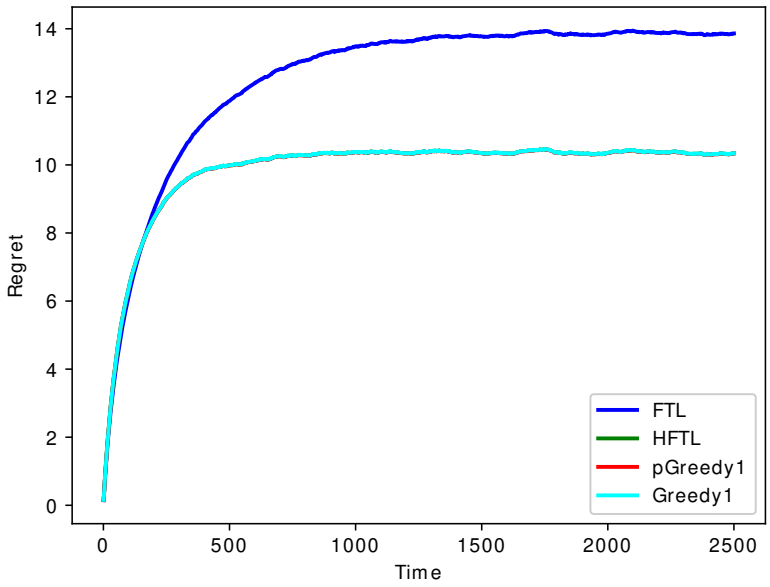


Figure 3.5: Regret of various algorithms as a function of time in the full information framework and first-order dominance scenario. Curves of HFTL, pGreedy1 and Greedy1 are similar.

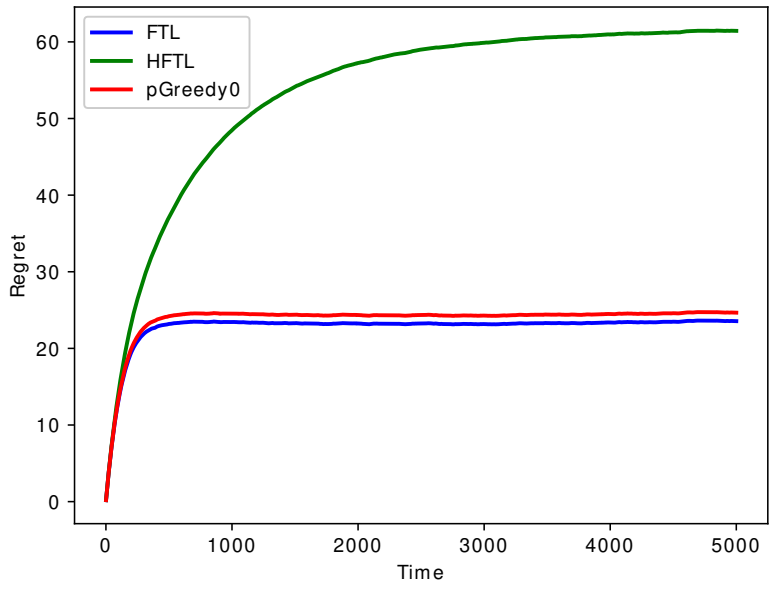


Figure 3.6: Regret of various algorithms as a function of time in the full information framework and strong dominance scenario.

In this scenario, HFTL, Greedy1 and pGreedy1 have similar performance and clearly outperform FTL. They ruled out the overall second best arm by taking into account the structure of the arms. Our pseudo-Greedy algorithm imitates excellently the Greedy one, and interestingly, HFTL matches up with these two algorithms.

Scenario 3: Worst-case

In the last scenario, we consider a more challenging problem in which the structure of the arms are less discernible. Precisely, there are ten arms per category with expected rewards 0.4 except for one arm in one category which have the expected mean of 0.6. The simulations were run until time horizon $T = 5000$. Results are presented on Figure 3.6.

The goal of this experiment is to show why one cannot be satisfied with HFTL; as soon as the means of our categories are close, HFTL performs poorly. In this scenario, FTL has the lowest regret, closely followed by pGreedy0. This is not surprising as nearly all arms were selected to be noise in this scenario.

4 Bandits for online advertising

Contents

In this chapter, we revisit the multi-armed bandit framework for online advertising in the pay-per-click model. In this setting, several advertisers would like to display an ad on a given search query and the search engine must choose which ad to show. The selected advertiser pays only when her ad is clicked. Sponsored contents have two distinctive characteristics: they have both a budget and a lifetime. Previous works on the bandit framework looked at the setting by addressing either the issue of the budget, in the so-called bandits with budgets framework, or the lifetime constraint, in the mortal bandit framework. When in fact, both issues are verified for sponsored contents in practice. This chapter aims at bridging the gap between the two literature by providing a general framework that consider both these constraints. We present several multi-armed bandit algorithms relevant to the setting and perform an empirical evaluation, both qualitative and quantitative, of these algorithms. Finally, we carry out a simulation with parameters taken from real data.

4.1 Introduction

The problem of online advertising can be described as follows. Given a search query, several advertisers would like to be display their product on a prominent position and the search engine must pick which ad to show. The selected advertiser then pays a fee only when her ad is clicked; this is the so-called pay-per-click model. The peculiarity of these contents is that they have both a budget and a lifetime. Each advertiser has a finite amount of money she is willing to pay, hence the amount of times her ad can be displayed is also finite. Furthermore, each advertiser usually define a specific period of time for her ad to be displayed. This may be due to seasonal events, a change

in the advertiser campaigns or any other reason in which the search engine has no control over. Both the budget of an ad and its lifetime are generally known in advance since they are set at the beginning of the ad campaign. The only uncertainty is that the click-through rate (henceforth, CTR) of an ad, i.e. the probability that it will be clicked is unknown. Nonetheless, the estimates of these click probabilities can be refined over time.

In the standard multi-armed bandit framework, the optimal solution is always to pull the best arm, i.e. the one with the highest expected reward. However in the ad allocation setting, this is no more the case. To be convinced of this, consider a simple example with two ads, one with a long lifetime and a small budget and the other with a short lifetime and a large budget. If the first ad yields a better payoff, one may be tempt to only show this ad. Yet, as its budget is so small, it may run out quickly; so quickly that as this time comes, the second arm is no more available. Thus there is a strong probability that the cumulative reward culminates to a lower amount that if we had exhaust the second ad first and then the first one.

In this chapter, we study a bandit setting where the number of pulls of an arm is restricted by its budget and/or its lifetime. These features have previously been dealt with separately, under the name of bandits with budgets and mortal bandits, respectively. Though both models differ, the motivation remains exactly the same: the optimization of ad allocation. Thus we present algorithms adapted from the literature and perform extensive empirical evaluation on simulated and real data.

4.2 Related work

As previously mentioned, this chapter is at the junction of bandits with budgets and mortal bandits.

In the bandit with budgets model, arms have a (limited) budget that restrict their number of pulls. Jiang and Srikant [72] and Slivkins [134] proposed natural extensions of the UCB algorithm to this setting. Combes, Jiang, and Srikant [40] studied an extension of KL-UCB and proved that it is asymptotically optimal. They further presented two algorithms adapted from the knapsack bandit problem. This framework should not be confused with the budgeted bandit setting [110] where the pull of an arm implies a random cost and the learner aims at maximizing her cumulative reward with a constrained budget.

On the contrary in the mortal bandit model, arms have a specific time of arrival and a lifetime; they cannot be played outside their windows. Thus, new arms may appear all the time and an algorithm needs then to continuously explore new arms. Chakrabarti, Kumar, Radlinski, and Upfal [32] proposed AdaptiveGreedy, an algorithm based on Epsilon-Greedy in which the probability of exploration depends on the performance of the best arm available. Recently, Traca, Rudin, and Yan [140] argued to limit the exploration of dying arms and modified AdaptiveGreedy and UCB in that sense.

As we will see later, the core optimization problem resulting from taking into account both the constraint of budget and lifetime will be a knapsack problem [114]. Moreover, the bandit with budgets setting is actually a special case of bandit with knapsack problem. Bandits with knapsack have already been studied [4, 13, 70, 142]. Unfortunately, they treat only the "simplest" knapsack problem while in the case of ad allocation, the problem becomes more complex as we will see.

4.3 Problem setup

We now formalize the setting. The decision maker sequentially selects a functional arm among K to be displayed at each time step $t \in \{1, \dots, T\} =: [T]$, where T denotes the time horizon. For an arm to be functional, it needs on one hand to be available and on the other hand to have enough budget in case of a click. Each arm $k \in [K]$ is characterized by the following quantities: a CTR $\mu_k \in [0, 1]$, a payment per click b_k, a budget B_k, a time step of arrival $s_k \in [T]$ and a lifetime $l_k \in \mathbb{N}^\star$. We point out that only the CTRs are uncertain, since the budget and payment per click are known and we often know in advance when ads will disappear. We further emphasis that we consider the pay-per-click model, meaning that the advertiser of arm k pays b_k to the decision maker if and only if her arm obtained a reward of 1, i.e. the ad has been clicked. It is also natural to assume that arms are Bernoulli distributed.

While arms can come and go, there exist periods of time, which we call batches, where the set of arms remains the same. Formally, there exists $M \in \mathbb{N}^\star$ such that $(s_k, l_k) \in \{t_0, \dots, t_M\}^2$ with $1 = t_0 < t_1 < \cdots < t_M = T$ for all ad k, where with a slight abuse of notation, arms die after the time horizon, i.e. $s_k + l_k \leq T$ for all arm $k \in [K]$. We denote by $T_m = t_m - t_{m-1}$ for $m \in [M]$ the length of batch m, $\mathcal{A}_m$ the set of arms available at the beginning of batch m and $\mathcal{A} \coloneqq \cup_{m \in [M]} \mathcal{A}_m$ the set of all arms. Notice that we recover the bandit with budgets and the mortal framework by setting $M = 1$ and $B_k = \infty$, respectively.

The goal of the decision maker is to maximize the total reward $\sum_{m=1}^{M} \sum_{t=1}^{T_m} w_{A_t}$, where $w_k \coloneqq b_k \mu_k$ is the expected value of one impression of arm k and A_t is the arm selected at time t. As usual in the bandit literature, the goal is equivalent to minimize the regret, by comparing with an "oracle" strategy. Unfortunately in this model, the oracle has no closed-form solution and is furthermore tedious to evaluate. For the sake of completeness however, in the rest of this section, we describe the oracle along with some remarks. The oracle is the solution of the following optimization problem,

$$\max_{\mathbf{x}} \sum_{m=1}^{M} \sum_{k \in \mathcal{A}_m} w_k x_k^m \tag{4.1}$$

subject to

$$\begin{aligned}
&\sum_{m=s_k}^{s_k+l_k-1} x_k^m \leq \frac{B_k}{w_k}\,, \quad \forall\, k \in \mathcal{A} \\
&\sum_{k \in \mathcal{A}_m} x_k^m \leq T_m\,, \quad \forall\, m \in [M] \\
&x_k^m \geq 0 \text{ and integer}, \quad \forall\, m \in [M], \forall\, k \in \mathcal{A}_m
\end{aligned}$$

The oracle actually solves a bounded multiple knapsack problem with assignment restrictions and unit weights. With words, that means that we have several knapsacks (the batches in which arms neither appear nor disappear) with several items (the ads), where each items have a given number of copies (this represents the budget of an ad), a given item cannot go in some knapsack (ad not available) and each item have a unit weight (we display one ad at each time step). The problem being a generalization of the multiple knapsack problem is NP-hard. It is itself a special case of the

generalized assignment problem. There exists some work on multiple knapsack problem with assignment restrictions focusing more on approximation algorithms [44, 47, 118]. As the items have a unit weight, the solution of the greedy algorithm is optimal in the standard knapsack problem. Hence the greedy strategy is a 2-approximation algorithm in this setting [39].

4.4 Algorithms

In this section, we go through the algorithms relevant to the dynamic ad allocation setting. We also make (in this section only) a slight abuse of terminology by describing the score of an arm by its CTR. Actually, we should multiply this quantity by the payment per click but since this a known constant, it would only hinder the notation. All algorithms automatically pull arms that are available and never selected before.

KL-UCB The first algorithm is a natural extension of KL-UCB [56]. It consists of pulling the arm with the largest KL-UCB index among available arms. It has further been analyzed by Combes, Jiang, and Srikant [40] in the bandit with budgets setting. We recall that the KL-UCB index of arm k at time t is defined by

$$\sup\{q \in [\widehat{\mu}_k(t), 1] : N_k(t) d(\widehat{\mu}_k(t), q) \leq \log(t - s_k) + 3\log(\log(t - s_k))\}$$

where $\widehat{\mu}_k(t)$ and $N_k(t)$ denote respectively the empirical mean and number of pulls of arm k up to time t and $d(a, b)$ represent the Kullback–Leibler divergence between Bernoulli distributions with parameter a and b, respectively.

KUBE The KUBE (knapsack–based upper confidence bound exploration and exploitation) algorithm is inspired by Tran-Thanh, Chapman, Rogers, and Jennings [142] and is depicted in Algorithm 8. The idea behind KUBE is to compute a solution of the knapsack problem that results from available arms. Since some parameters of the optimization problem are unknown, we use upper confidence bound of the estimate. Formally, at each time step t, KUBE solves the following problem

$$\begin{aligned} &\max_{\mathbf{x}} \sum_{k=1}^{K_t} x_k \, \mathrm{UCB}_k(t) && (4.2)\\ &\text{such that} \quad \sum_{k=1}^{K_t} x_k \leq T - t + 1 \\ &\qquad x_k \leq \min\left\{ \frac{B_k(t)}{\mathrm{LCB}_k(t)}, L_k(t) \right\} \quad \forall k \in [K_t] \\ &\qquad x_k \geq 0 \text{ and integer } \forall k \in [K_t] \end{aligned}$$

where K_t denotes the number of arms available at time t, $B_k(t)$ and $L_k(t)$ are respectively the remaining budget and the remaining lifetime of arm k at time t; $\mathrm{UCB}_k(t)$ and $\mathrm{LCB}_k(t)$ denote respectively an upper and lower bound on the expected reward of arm k computed with the infor-

mation gathered until time t. We make use of the standard confidence radius $\sqrt{\frac{\log(t-s_k)}{2\,N_k(t)}}$. Since x_k represents the number of times arm k must be pull to obtain the optimal solution, KUBE randomly selects an arm according to the solution of the knapsack problem, i.e. $\mathbb{P}(A_t = k) = \frac{x_k}{\sum_{j=1}^{K_t} x_j}$. Since the knapsack problem is NP-hard, we make use of a greedy approximation to compute the solution; we rank the arms by decreasing **UCB** index and from the top to the bottom, we increment x_k as much as possible with respect to the constraints.

Algorithm 8: KUBE

Input: Horizon T, budget and lifetime of arms when they arrive
for $t \leftarrow 1$ **to** T **do**
  if *arm k is available and* $N_k(t) = 0$ **then**
    $A_t = k$
  else
    Compute solution $(x_k)_k$ of Equation (4.2)
    Pull randomly A_t with $\mathbb{P}(A_t = k) = \frac{x_k}{\sum_{j=1}^{K_t} x_j}$

BalancedExploration BalancedExploration is an adaptation of the first algorithm proposed by Badanidiyuru, Kleinberg, and Slivkins [13] and is summurizd in Algorithm 9. The idea is to simultaneously exhaust the budget of best arms at the time horizon T. Precisely, the time is divided into batches and the length of a batch is the number of arms available at the beginning of it. At the start of a batch, we compute the UCB index of arms along with a bound on the number of times each of arms can be pulled over the remaining time steps. Then we construct a probability distribution over the available arms. Specifically, starting from the best available arm to the worst, we assign a probability mass which is the estimated number of times the arm can be pulled divided by the remaining number of time steps. We do so until the accumulated probability reached 1. We make use of the following confidence radius

$$\mathrm{rad}(\widehat{\mu}, N) = \sqrt{\frac{C_{\mathrm{rad}}\,\widehat{\mu}}{N}} + \frac{C_{\mathrm{rad}}}{N}$$

where $C_{\mathrm{rad}} = \log(TK)$, $\widehat{\mu}$ is the average empirical reward of an arm and N denotes its number of pulls. The estimated number of pull of arm k at time t is thus $\min\left\{T - t + 1, L_k(t), \frac{B_k(t)}{\mathrm{LCB}_k(t)}\right\}$. Then for each time step in the phase, BalancedExploration chooses an arm according to this distribution. If the selected arm is not available anymore, it passes to the next round.

PrimalDualBwk PrimalDualBwK is an adaptation of the second algorithm proposed by Badanidiyuru, Kleinberg, and Slivkins [13] and is depicted in Algorithm 10. The idea is to greedily pull arms with the greatest bang-per-buck, i.e. reward per unit of resource consumption. To do so, the algorithm treats the budget and lifetime of an arm as resources and puts a fictitious

Algorithm 9: BalancedExploration

Input: Horizon T, budget and lifetime of arms when they arrive
for *phase* $p \leftarrow 1$ **to** ... **do**
  for *arm* $k \leftarrow 1$ **to** K_t **do**
    Compute $\text{LCB}_k(t)$ and $\text{UCB}_k(t)$
  Compute a distribution $\mathcal{D}$ over arms
  for *the next* K_t *rounds* **do**
    Choose an arm k as an independent sample from $\mathcal{D}$
    if *arm* k *is available* **then**
      Pull arm k
    else
      Pass

price on each arm. Additionally, we also consider the remaining number of rounds as a resource. PrimalDualBwK considers the same confidence radius as BalancedExploration.

4.5 Warm-up: A toy model

To illustrate that the greedy oracle, i.e. playing the best arm available, is no longer the optimal solution, look at this simple example. Consider a problem with two arms whose CTR are 0.2 and 0.1, a payment per click of 1 for both, and a budget of 1000 and 5000 respectively. The time horizon $T = 10000$ is divided into two batches with 5000 time steps each. Both arms are at hand at the beginning but the second arm is only available in the first batch while the first one is all time long. In this example, the first arm is better than the second one but it does not have enough budget to be played all along the time horizon. Actually, in average, it has roughly enough budget for one batch. Thus, an oracle algorithm must play the second arm in the first batch then the first arm in the second batch. We compare the previously described algorithms with a random and a greedy algorithms. The former selects randomly an available arm while the latter pulls the arm with the best empirical reward. Results are averaged over 1000 iterations and are presented on Figure 4.1.

As expected, KL-UCB performs poorly as it concentrates on the optimal arm on the first batch. PrimalDualBwK performs similarly. KUBE and BalancedExploration manage to leverage the knowledge of the budget and availability to achieve a better final reward. We can see that their cumulative rewards is lower than the one of KL-UCB in the first batch indicating that they pull less the optimal arm. Interestingly, Random and Greedy achieve great final rewards. The former was expected since in expectation the optimal arm in only pulled half the time; while for the latter, it is actually its failure of finding the optimal arm that improve its reward in the end.

Algorithm 10: PrimalDualBwK

Input: Horizon T, budget and lifetime of arms when they arrive

for $t \leftarrow 1$ **to** T **do**

  if $t = 1$ *or a new arm arrive* **then**

    Set $v_t = \mathbf{1}_{K_t+1}$

  for *arm* $k \leftarrow 1$ **to** K_t **do**

    if *arm* k *is available and* $N_k(t) = 0$ **then**

      $A_t = k$

    else

      Compute $\mathrm{LCB}_k(t)$ and $\mathrm{UCB}_k(t)$

  Set $y_t = v_t / \|v_t\|_1$

  Pull arm $k \in \arg\min_{j \in [K_t]} \dfrac{y_{K+1} + y_j \mathrm{LCB}_j(t)}{\mathrm{UCB}_j(t)}$

  Set $\varepsilon = \sqrt{\dfrac{\log(K_t+1)}{B}}$ where $B = \min\left\{T - t + 1, \min_k L_k(t), \min_k \dfrac{B_k(t)}{\mathrm{LCB}_k(t)}\right\}$

  Update $v_{t+1,K_t+1} = v_{t,K_t+1} \cdot (1+\varepsilon)$ and

  for *arm* $j \leftarrow 1$ **to** K_t **do**

    if $L_j(t) \le B_j(t)/LCB_j(t)$ **then**

      $v_{t+1,j} = v_{t,j} \cdot (1+\varepsilon)$

    else if $j = k$ **then**

      $v_{t+1,k} = v_{t,k} \cdot (1+\varepsilon)^{\mathrm{LCB}_k(t)}$

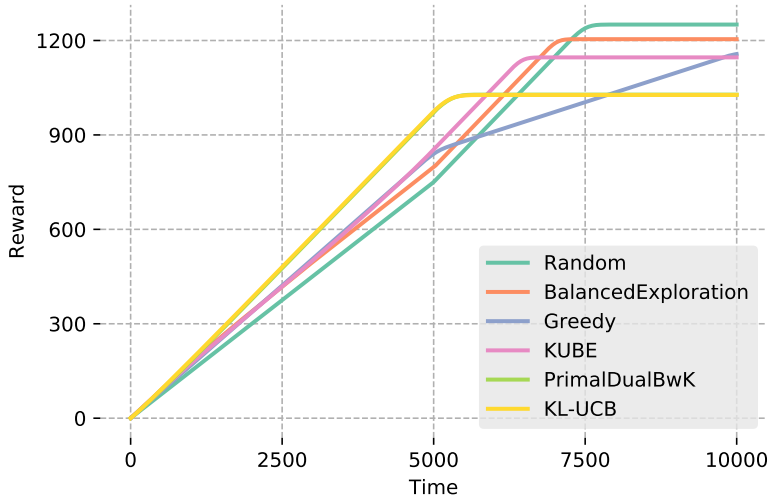
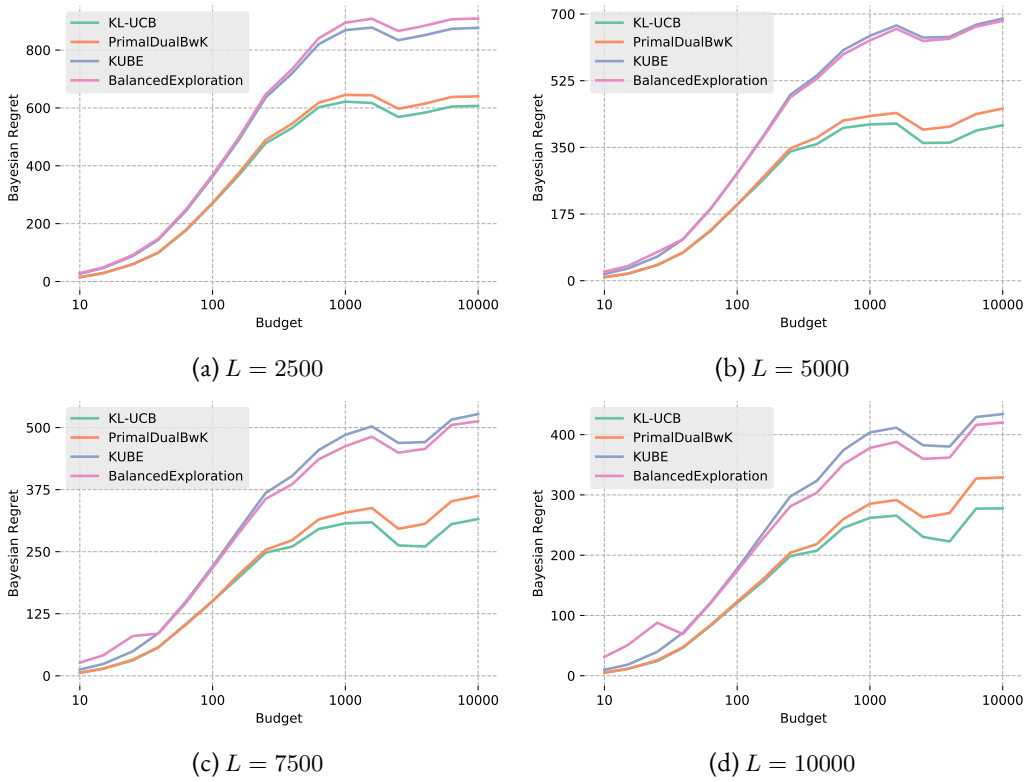


Figure 4.1: Reward of various algorithms as a function of time in a toy model. Curves of KL-UCB and PrimalDualBwK are similar.



(a) $L = 2500$ (b) $L = 5000$

(c) $L = 7500$ (d) $L = 10000$

Figure 4.2: Bayesian regret of various algorithms as a function of the budget of arms for diverse expected lifetime L.

4.6 Empirical evaluation

In this section, we evaluate the proposed algorithms on several simulated setups. Specifically, we firstly consider a setting close to bandits with budgets and study the influence of the lifetime of arms; then, we consider a setting close to mortal bandits and study the impact of the budget of arms.

4.6.1 Bandit with budgets setting with lifetimes

Like the previous section, we study a framework with a single set of arms that arrive at the same time. This setting is similar to the bandit with budgets model, with the additional assumption that arms have a lifetime. Specially, we consider problems with $K = 10$ arms with expected rewards drawn i.i.d. from a Beta(1, 9) distribution. Arms have the same budget and the same expected lifetime L. Lifetime of arms are draw i.i.d. from a Geometric distribution with expectation L. We compare the performance of the previously described algorithms (Random and Greedy are this time far from optimal) for different expected lifetime and we vary the budget of arms. Results are averaged over 500 iterations and are displayed on Figure 4.2.

We see that KL-UCB performs slightly better than PrimalDualBwK, though the difference is significant for large budget and long lifetime, and they both outplay KUBE and BalancedExploration. It is interesting to mention that the latter two algorithms aimed at achieving the optimal solution at time T while the others two are more greedy and pull the best "bang-per-buck" arm. While this principle works great on the toy model, it seems that the uncertainty on more complicated models hinders its efficiency.

4.6.2 Mortal bandit setting with budgets

We now study a setting closer to the mortal bandit model, where arms have a (limited) budget. The number of arms available remains fixed throughout the time horizon T, that is when an arm dies, it is immediately replaced by another one. We emphasis that there can be arms available that have exhausted their budget. The lifetime of arm k, denoted L_k, is drawn i.i.d. from a Geometric distribution with expected lifetime L; this arm died after being available for L_k rounds. Specially, we again consider problems with $K = 10$ arms with expected rewards drawn i.i.d. from a Beta(1, 9) distribution. Arms have the same budget and the same expected lifetime L. We compare the performance of the previously described algorithms for different budget and we vary the expected lifetime of arms. We slightly modify BalancedExploration and KUBE to optimize the solution with respect to the expected lifetime instead of the time horizon, i.e. in both algorithms, the probability distribution over arms is computed as if the time horizon has been divided in batches of length the expected lifetime of arms and we want to optimize the cumulative reward in those batches. In practice, this considerably improves the performance of both algorithms. Results are averaged over 500 iterations and are displayed on Figure 4.3.

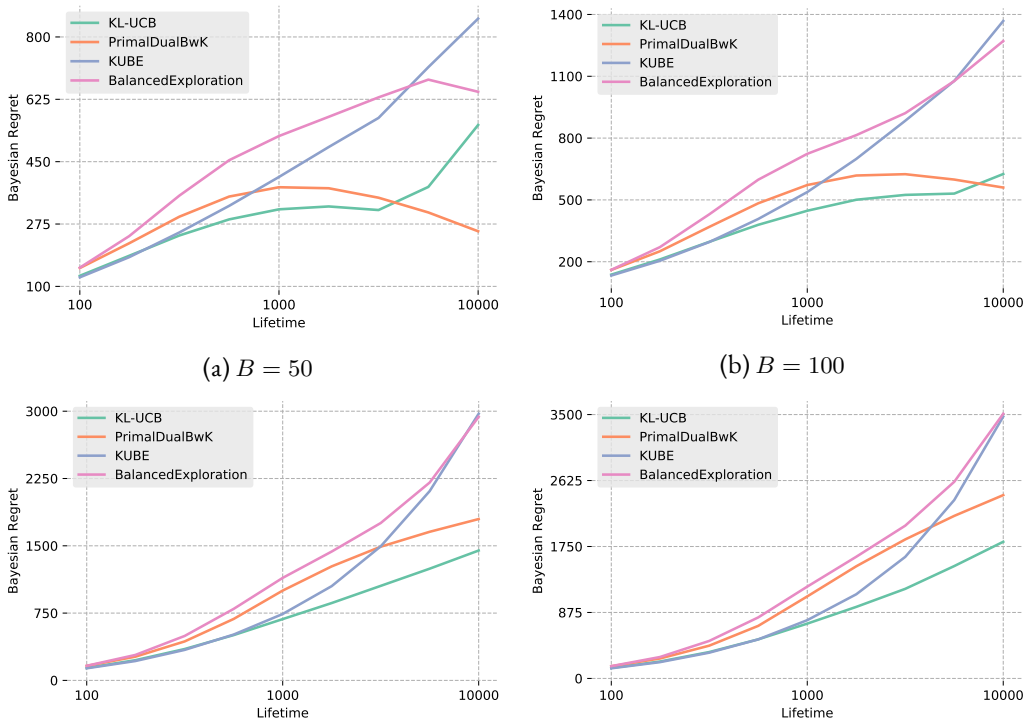
For large budget, KL-UCB outperforms other algorithms. It is interesting to notice that its regret increase roughly logarithmically as a function of the lifetime, just as PrimalDualBwK while for KUBE and BalancedExploration the increase is more linear. For small budget, we again observe that KL-UCB performs best but only for relatively short lifetime while PrimalDualBwK outmatches it for long lifetime; its regret even decreases compared to moderate lifetime. KUBE and BalancedExploration are once again outplayed.

4.7 Dynamic ad allocation

We now compare the performance of the proposed algorithms on a simulation with real-world parameters from an ad allocation problem. We start with a brief description of the problem on the Cdiscount website, one of the leading e-commerce companies in France. For each search query inputs, the search engine on the Cdiscount website outputs a list of products, in the order of ten, on some slots. Among these slots, a small number is generally saved for sponsored contents. The objective is to optimize the revenue on these slots.

We have collected the data of auctions for a specific search query[1] over a period of two months. These data contains for each ad, the budget of the advertiser, the date the campaign starts, the date of its end and its bid, i.e. the maximum amount the advertiser is willing to pay. As auctions are usually second-price, this bid is not the pay-per-click. Moreover, some ads can be the subject

[1]The keyword and the resulting data will not be revealed to protect business-sensitive information.



(a) $B = 50$

(b) $B = 100$

(c) $B = 500$

(d) $B = 1000$

Figure 4.3: Bayesian regret of various algorithms as a function of the lifetime of arms for diverse budget B.

Table 4.1: Relative performance of various algorithms on a simulation with real-world parameters.

KL-UCB	KUBE	BalancedExploration	PrimalDualBwK
2.035	1.348	1.950	1.990

of different bids, for different campaigns. We make use of the mean pay-per-click in these cases. Each ad has been subject to some number of impressions and clicks for the search query. We discarded any ad with less than 100 impressions for better estimate of the CTRs and less bias. This results in $K = 52$ total arms. The number of time step, i.e. the number of times the keyword has been searched, has also been collected for each day. To protect business-sensitive information, we report the relative performance of the proposed algorithms, which is the cumulative reward of an algorithm divided by the one of the random policy, similarly to Li, Chu, Langford, and Schapire [103]. Results are averaged over 500 iterations and are presented on Table 4.1.

Once again and as expected after the previous section, KL-UCB performs better than the other algorithms even if BalancedExploration and **PrimalDualBwK** are close behind; **KUBE** is unsurprisingly outplayed.

4.8 Conclusion

In this chapter, we have studied a bandit model where arms have both a budget and a lifetime to tackle the problem of online advertising. Since these features have been dealt with separately in the literature, we have presented several algorithms and evaluated them on numerous experiments. We have showed that, despite the fact that the greedy oracle is no more optimal, conventional algorithms achieve better performance that algorithms that try to leverage the additional information due to the complexity and the uncertainty in real-world setups. Nonetheless, knapsack-based algorithms remain competitive empirically.

Part II

Bandit algorithms in practice

5 Lifelong learning in multi-armed bandits

Contents

Continuously learning and leveraging the knowledge accumulated from prior tasks in order to improve future performance is a long standing machine learning problem. In this paper, we study the problem in the multi-armed bandit framework with the objective to minimize the total regret incurred over a series of tasks. While most bandit algorithms are designed to have a low worst-case regret, we examine here the average regret over bandit instances drawn from some prior distribution which may change over time. We specifically focus on confidence interval tuning of UCB algorithms. We propose a bandit over bandit approach with greedy policies and we perform extensive experimental evaluations in both stationary and non-stationary environments. We further apply our solution to the mortal bandit problem, showing empirical improvement over previous work.

5.1 Introduction

Most of the work in the stochastic bandit literature focuses on developing algorithms with optimal worst-case regret on some problem class, typically on bounded or subgaussian rewards. While the theory guarantees that these algorithms will have a sublinear regret in all instances within the problem class, they will be overly conservative for the majority of these instances, leading to a large regret. Additionally, on these "easier" instances, sophisticated algorithms are outperformed by simpler heuristics [89, 144], which is an obstacle to their implementation in practice. In this chapter, we consider that the learner successively interacts with a task sampled from a problem instance with some prior distribution, which may or may not be stationary over time. Within each

task, there is a learning problem which is a multi-armed bandit problem with a fixed time horizon. The learning agent does not know the model parameters of each bandit problem, nor does she know the prior distribution within the bandit instance. Yet, it is critical for the learner be to able to "track" this probability distribution to build lifelong learning agent and thus achieve the best possible performance. In a stationary environment, the learner experiences the same bandit problem over and over again, and an efficient policy, in order to improve the performance when it faces the same problem again, should leverage the information acquired from previous tasks. Conversely in a non-stationary environment, the distribution within the bandit instance may change over time and the previously learned solution may fail to keep its good performance under the new distribution. An efficient policy must thus continuously learn to improve itself. The goal of the learning agent is then to learn a policy that selects a bandit algorithm for each task, and achieves a low lifelong regret, that is a low cumulative regret across all tasks. This approach can be regarded as a special case of meta-learning [131, 139] or lifelong learning [36, 132, 138].

Our results In this chapter, we focus on a tractable instance of this problem, the tuning of the confidence interval width of UCB-like policies. We first evaluate the impact of the choice of algorithm on empirical performance over several environments with different prior and different number of arms. On a side note, we prove a bound on the Bayesian regret of a tuned UCB algorithm showing that we do not lose all theoretical guarantees. We then concentrate on learning in a stationary environment. We first investigate the influence of the various initializations on performance. Next, we discuss about the main part of this chapter, i.e., the learning of the optimal algorithm. For this purpose, we consider a bandit over bandit approach, that is using a bandit algorithm to choose the optimal parameter. We show empirically that the Greedy algorithm as meta-algorithm performs extremely well. We next concentrate on learning in a non-stationary environment, precisely we look at both an abruptly changing and a slowly changing environment. To this effect, we adapt the Greedy algorithm using methods from non-stationary bandits. Finally, we apply our method to a more realistic setting: the mortal bandit problem. By decomposing the time horizon into episodes according to the expected lifetime of arms, we show great empirical improvement compared to previous work.

5.2 Related work

Bayesian bandits [19, 61, 62] have been studied extensively with the goal of developing optimal algorithms in the Bayesian sense. Lower bounds on the Bayesian regret are also given by Kaufmann [78] and Lai [94]. Unfortunately, computing the Bayesian optimal algorithm is generally intractable [100].

The closest to our work is that of Lazaric, Brunskill, et al. [101] who considered a framework which closely resembles to ours except they studied the stochastic setting and further assumed a finite set of models; whereas we consider the Bayesian setting and do not make any assumption on the number of changes in the prior distribution. Deshmukh, Dogan, and Scott [51] extended the previous work in the contextual framework. Also related to our work, Maes, Wehenkel, and Ernst [111] tuned existing algorithms and also learned index policies of historical features. Hsu, Kveton, Meshi, Mladenov, and Szepesvari [69] proposed a best-arm identification algorithm for tun-

ing the confidence interval of the UCB algorithm and the posterior distribution of the THOMPSON SAMPLING algorithm. Boutilier, Hsu, Kveton, Mladenov, Szepesvari, and Zaheer [25] focused on "differentiable" algorithms and optimized them by gradient ascent. In comparison, the setting of these works are offline while we are more interested in the lifelong regret incurred in potentially non-stationary environments.

Several works also use bandit algorithms as meta-algorithm. Li, Jamieson, DeSalvo, Rostamizadeh, and Talwalkar [104] introduced a bandit-based approach to hyperparameter optimization using SEQUENTIAL HALVING, a pure-exploration bandit algorithm, as a subroutine. The celebrated UCT algorithm [87] makes use of the UCB algorithm applies on trees. In the non-stationary framework, a few methods involve bandit algorithms in a hierarchical way: Hartland, Gelly, Baskiotis, Teytaud, and Sebag [66] considered a meta-bandit to decide whether to accept the change detection or not; Cheung, Simchi-Levi, and Zhu [37] used the EXP3 algorithm to decide the window size of the SW-UCB algorithm; and Wu, Iyer, and Wang [149] and Wu, Wang, Li, and Wang [150] adopted a hierarchical bandit algorithm where a master bandit manages some slave bandits.

The model can also be viewed as a special case of mortal bandits [22, 32, 140] in which arms show up by batch where the time of death is the same for each arm in a given batch.

5.3 SETTING

In the stochastic multi-armed bandit model, an agent interacts sequentially with a set of K distributions $\mathcal{V}_1, \dots, \mathcal{V}_K$, called arms. At time t, the agent chooses an arm A_t, which yields a reward X_t drawn from the associated probability distribution $\mathcal{V}_{A_t}$. The objective is to design a sequential strategy maximizing the expected cumulative reward up to some time horizon T. Let $\mu_1, \dots, \mu_K$ denote the mean rewards of arms, and $\mu^\star := \max_{k \in [K]} \mu_k$. The goal is equivalent to minimizing the regret, defined as the difference between the expected reward accumulated by an oracle strategy always playing the best arm at each round, and the one accumulated by an algorithm π,

$$\mathbb{E}[R(T, \pi)] = \mathbb{E}\left[\sum_{t=1}^{T} (\mu^\star - \mu_{A_t})\right]$$

where the expectation is taken with respect to the randomness in the sequence of successive rewards from each arm and the possible randomization of the algorithm. Furthermore, we assume the following Bayesian parametric bandit setting: a random vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)$ is drawn from a prior distribution Q, and the distribution of arm k depends on the parameter θ_k. This leads to the notion of Bayesian regret which depends on the prior distribution Q,

$$\mathrm{BR}_Q(T, \pi) = \int \mathbb{E}[R(T, \pi)] dQ(\boldsymbol{\theta}) \,.$$

We consider a lifelong learning setting where at each episode j the learner interacts with a task $\mathcal{V}_{\theta_1^j}, \dots, \mathcal{V}_{\theta_K^j}$, where each θ_k^j is drawn i.i.d. from an unknown prior distribution Q_j. The objective is to find a series of algorithms minimizing the lifelong Bayesian regret over J episodes,

$$\mathrm{LBR}_{\boldsymbol{Q}}(T, \boldsymbol{\pi}) = \sum_{j=1}^{J} \mathrm{BR}_{Q_j}(T, \pi_j)$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_J)$ and $\boldsymbol{Q} = (Q_1, \dots, Q_J)$ denote respectively the algorithms and the prior distributions for each episode. We further assume that the horizon for each task T and the number of episodes J are known. An unknown number of episodes can be handled as usual [48]; while the knowledge of the time horizon is mostly for convenience in the choice of the sub-algorithm and similar results can be achieved with an anytime algorithm.

In order to minimize the lifelong regret, our method consists of designing a meta-algorithm which, at each episode j, selects an algorithm π_j from a set of algorithms, which can be finite or infinite, that aims at minimizing the Bayesian regret with respect to the prior distribution Q_j.

We especially focus on a tractable set of algorithms: the UCB algorithm [9] with a parameter γ controlling the width of the confidence interval. We emphasis that similar results can be obtained with any UCB-like algorithm. Formally, the UCB(γ) index of arm $k \in [K]$ in round t is

$$\mathrm{UCB}_k(t) = \widehat{\mu}_k(t-1) + \gamma \sqrt{\frac{2\log(1/\delta)}{N_k(t-1)}}$$

where $\widehat{\mu}_k(t)$ is the average reward of arm k in the first t rounds, $N_k(t)$ is the number of times that arm k have been pulled in the first t rounds and $\delta = 1/T$ is the probability that the confidence interval fails. Note that the case $\gamma = 1$ corresponds to the theoretical value that minimizes the worst-case regret, while $\gamma = 0$ corresponds to the Greedy algorithm. Hence for this class of algorithms, the meta-algorithm faces a new trade-off between being conservative and being aggressive on a class of bandit instances. Thereby, the objective that is to minimize the lifelong regret is equivalent to finding the optimal values of $\gamma \in [0, 1]$ that minimizes the Bayesian regret in each episode.

Theoretically, any minimax optimal algorithm will also be optimal, up to constant factors, in the Bayesian setting [100]; and thus optimal in the lifelong setting. Yet, we will see that we can empirically improve these algorithms. We end this section with an analysis of this tuned UCB. More generally, Russo and Van Roy [129] noticed that the Bayesian regret of any UCB-like algorithm satisfies

$$\mathrm{BR}_Q(T, \mathrm{UCB}) \leq \mathbb{E}\left[\sum_{t=1}^{T} (\mu^\star - \mathrm{UCB}_{A^\star}(t)) + \sum_{t=1}^{T} (\mathrm{UCB}_{A_t}(t) - \mu_{A_t})\right] \tag{5.1}$$

where $A^\star$ denotes the optimal arm, which is a random variable. Specifically for the UCB algorithm we state the following theorem.

Theorem 5.1. *For* 1*-subgaussian distributions with mean in* $[0, 1]$*, UCB(γ) with* $\gamma > 0$ *satisfies*

$$\mathrm{BR}_Q(T, \mathit{UCB}(\gamma)) \leq 4KT^2\delta^{4\gamma^2} + 2\gamma\sqrt{2KT\log(1/\delta)}\,.$$

As an example, for $\gamma = 1$ and $\delta = 1/T$, we get the known bound $\mathcal{O}\big(\sqrt{KT\log T}\big)$.

Proof. Let E be the event that for all $t \in [T]$ and $k \in [K]$,

$$|\widehat{\mu}_k(t-1) - \mu_k| < \gamma \sqrt{\frac{2\log(1/\delta)}{N_k(t-1)}}\,.$$

Using the subgaussian assumption and a union bound, we get that $\mathbb{P}(E^c) \leq 2KT\delta^{4\gamma^2}$. On the event E^c, the terms inside the expectation of Equation (5.1) are bounded by $2T$, while on the event E, the first sum is bounded by 0 and for the second term, standard computations yield

$$\sum_{t=1}^{T} (\mathrm{UCB}_{A_t}(t) - \mu_{A_t}) \mathbf{1}\{E\} \leq 2\gamma\sqrt{2KT\log(1/\delta)}\,.$$

Putting together the pieces completes the proof. □

5.4 Choice of the class of algorithms

We begin by showing that the choice of the algorithm in the class of algorithms, i.e. the set of all algorithms in which we seek to find the one that maximize the average cumulative reward over bandit instances, is of the utmost importance in order to have the best possible empirical performance, and the most favorable algorithm depends on a number of factors. In Figure 5.1, we evaluate the Bayesian regret of several UCB-like policies as a function of their parameter γ in different settings. In the first scenario, the distribution of each arm is a Bernoulli distribution where the expected reward is drawn i.i.d. from a uniform distribution over $[0, 1]$. In the second, we change the problem to Gaussian bandits and in the third scenario, we change the prior to a Beta$(1, 3)$ distribution. In all scenarios, the horizon is fixed at $T = 1000$ and we vary the number of arms K. Results, averaged over 5000 iterations, are presented on Figure 5.1.

Let us concentrate on the first scenario one moment. For $K = 5$, i.e. for a small number of arms, we compare different UCB-like algorithms: the original UCB algorithm [9], MOSS [7], which is known to be minimax optimal contrary to UCB, and AdaUCB [96] which is simultaneously minimax optimal, asymptotically optimal, and never worse than UCB in the worst case. Interestingly, the later statement is also verified on the Bayesian regret for all values of γ. The same can also be said almost everywhere for MOSS over UCB. For large values of γ, AdaUCB outperforms MOSS while for small values, MOSS only improves slightly over AdaUCB. As a result, if one is purely interested in great empirical performance, a default choice would be AdaUCB for any value of γ. However in the lifelong setting, we want to optimize the parameter γ and it is not clear which choice of algorithm will perform better through all the episodes; AdaUCB may be better but this also makes it harder for the meta-policy to find the optimal value of γ, meaning a potentially larger lifelong regret compared to UCB for example. Additionally, the best Bayesian regret of the optimally tuned version of each algorithm is roughly the same for all three. Hence our choice of the UCB algorithm, which is simple enough for illustration and may also be suitable for practical applications. We finally note that for all algorithms, the gain of their tuned version over the default choice ($\gamma = 1$) is substantial even though MOSS and AdaUCB are minimax optimal up to constant factors, supporting our choice to optimize algorithms for practical purpose.

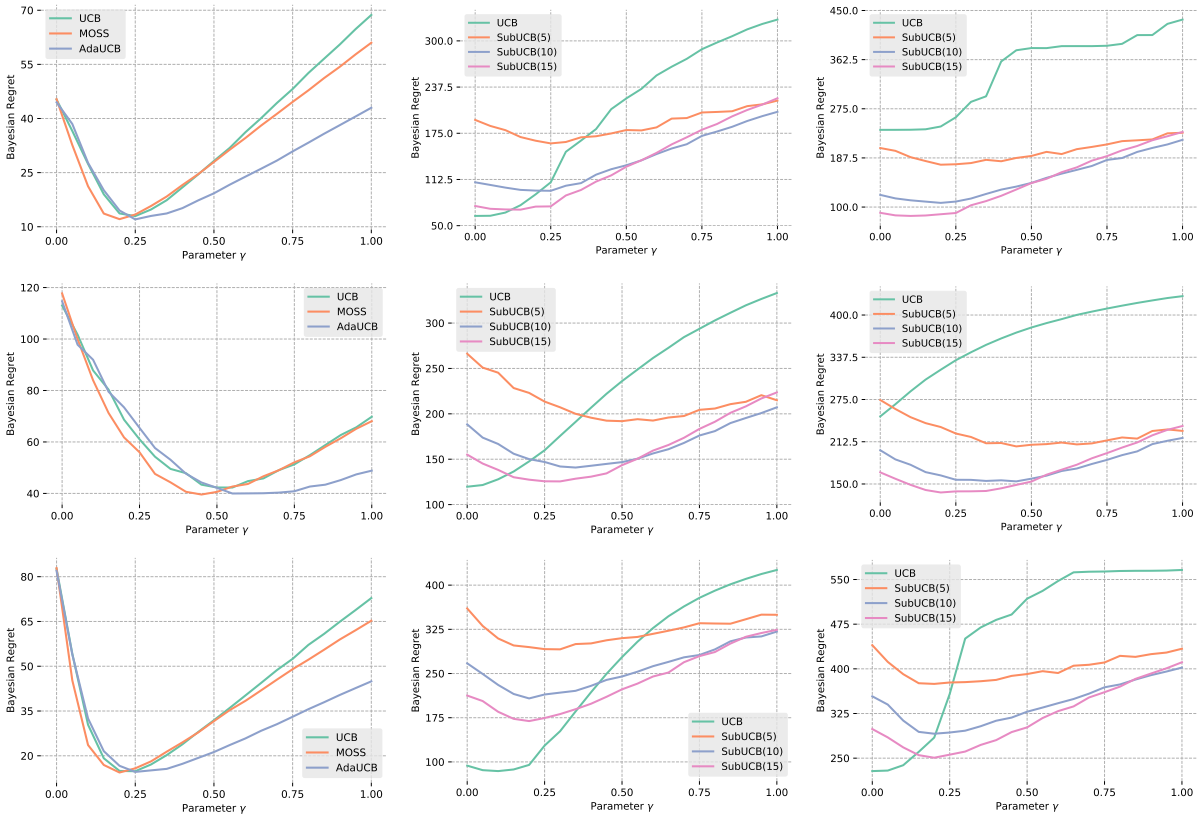


Figure 5.1: Bayesian regret of various algorithms as a function of γ for diverse environments and numbers of arms K. Rows correspond respectively to Bernoulli bandits with a uniform prior, Gaussian bandits with a uniform prior and Bernoulli bandits with a Beta(1, 3) prior. Columns correspond respectively to $K = 5$, $K = 63$ and $K = 250$ arms.

For $K = 63$ and $K = 250$, i.e. for moderate and large number of arms, we compare the UCB algorithm with sub-sampled versions of itself, denoted SubUCB(m). Formally, SubUCB(m) selects randomly m arms and performs UCB on these arms. For $K = 63$, we see that the Greedy algorithm, symbolized by UCB with $\gamma = 0$, performs better than any tuned UCB and also better than SubUCB. For $K = 250$, SubUCB performs always better than UCB. It is interesting to notice that as m grows bigger, up to a certain point, SubUCB(m) has a better optimal Bayesian regret, becomes more sensitive to the parameter γ and a sub-sampled version of the Greedy algorithm turns into the best strategy possible.

We mention that these behaviors, i.e. the good empirical performance of the Greedy algorithm and the superiority of a sub-sampled version of UCB in the case of a great number of arms, have been pointed out by Bayati, Hamidi, Johari, and Khosravi [17]. We showed that they also hold true empirically after tuning of the confidence interval width. Furthermore, these behaviors can also be observed in the second and third scenarios, however what we called moderate and large values of K are, in these cases, higher than previously.

5.5 Learning in a stationary environment

In this section, we consider the learning of the optimal algorithm in a fixed environment, i.e. when the prior distribution is the same in all episodes. In this case, the objective is equivalent to finding the algorithm that minimizes the Bayesian regret for that specific prior distribution.

5.5.1 Influence of the initialization

We start with a study of the effect of the initialization choice on empirical performance. By default, most bandit algorithms initialize arms by pulling them at least one time. Knowing that we solve again and again similar bandit problems, we may want to find a more clever initialization. For example, consider a challenging bandit problem with a large number of arms with respect to the time horizon and assume we have found a reasonably good arm; exploiting this arm may then be more rewarding that exploring new arms in the hope of finding a better one. This becomes especially critical the more greedy we get. In this section, we fix the value of the hyperparameter γ and we evaluate three different initializations. In all cases, we set the empirical means of arms to a specific value and their upper confidence bounds are built as if they have been played once. In the first case we set this value at 0 (Init 1), in the second (Init 2) and third (Init 3) cases, it is fixed at the mean and median, respectively, of previous empirical means. We denote by "Init 0" the default initialization, i.e. pulling each arm once.

In these experiments, the time horizon is set at $T = 1000$ for each episode. We consider the same scenarios as the last section; namely, Bernoulli bandits with a uniform prior, Gaussian bandits with the same prior and Bernoulli bandits with a Beta$(1, 3)$ prior. Once more, we vary the numbers of arms K. We fix $\gamma = 0.2$ and we repeat this tuned UCB for $J = 100$ episodes with the different initializations previously mentioned. On Figure 5.2, we report the lifelong regret averaged over 100 iterations.

It is complex to observe a clear trend across the different simulations. For small values of K, the choice of initialization is insignificant; except for the initialization at 0 in the first scenario, they all have roughly the same performance. For intermediate values of K, the impact of the choice

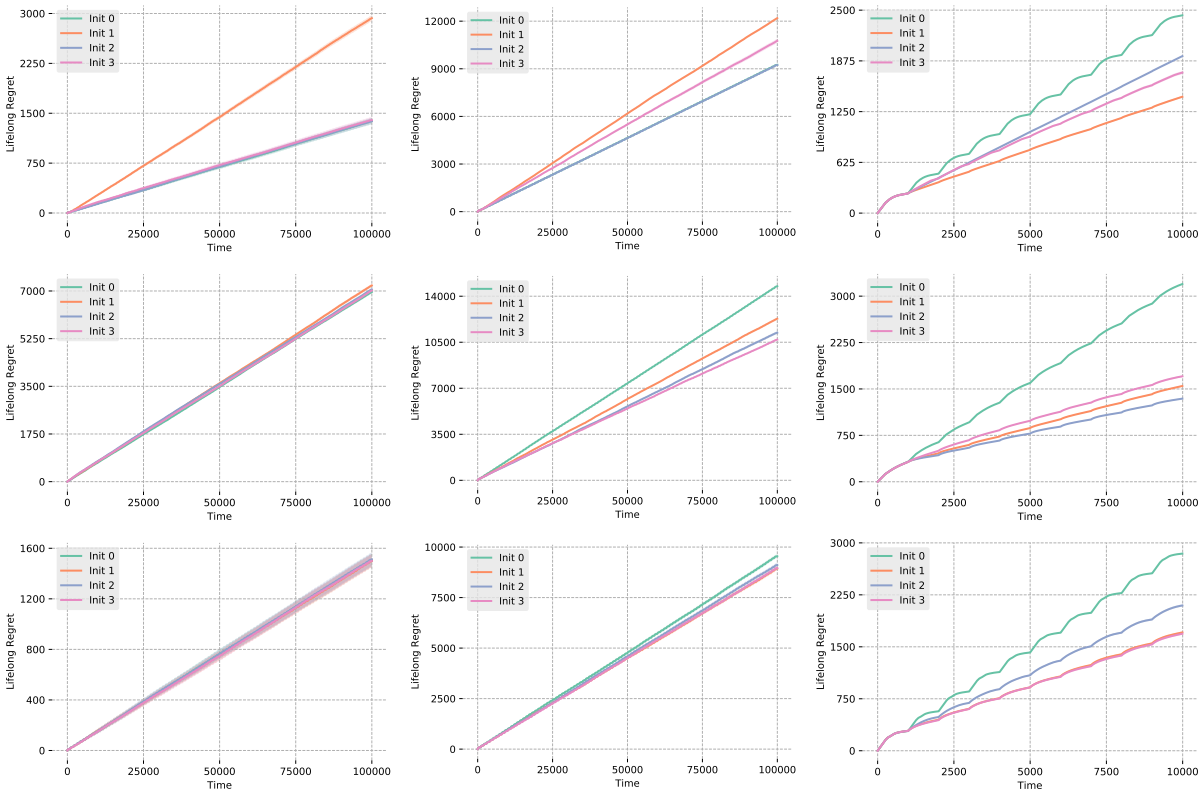


Figure 5.2: Lifelong regret of a deterministic meta-algorithm with various initializations in stationary environments. Rows correspond respectively to Bernoulli bandits with a uniform prior, Gaussian bandits with a uniform prior and Bernoulli bandits with a Beta(1, 3) prior. Columns correspond respectively to $K = 5$, $K = 63$ and $K = 250$ arms. Shaded areas show standard errors.

of initialization becomes apparent. Although there is no optimal choice, initializing arms with the median of previous arms seems more robust. For large value of K, a clear trend is emerging: pulling each arm once is always the worst thing to do. It was expected since we spend most of the time initializing arms. There is still no optimal choice in this case, yet the initialization at 0 seems more robust. As the choice of initialization is insignificant on instances with a small number of arms, which we study in this chapter, in what follows we assume that each strategy uses the default initialization in each episode.

5.5.2 Bandit algorithms as meta-algorithm

Now that the initialization rule is set, we can focus on the meta-algorithm, i.e. the algorithm that is responsible for picking the parameter γ of the UCB algorithm for each episode, and ultimately, it is the keystone in the minimization of the lifelong regret. We consider bandit algorithms for the choice of the meta-algorithm, since they are efficient online optimization algorithms. This may seem like a vicious circle as we are talking about optimization of bandits algorithms and we want to avoid having to optimize the optimizer. Fortunately, the two algorithms, the meta-algorithm and the sub-algorithm, face a different problem. Indeed, the sub-algorithm aims at maximizing the average reward over bandit instances while the meta-algorithm aims at maximizing a function, which is the expected cumulative reward of the sub-algorithm with respect to its parameter γ. The dilemma encountered by the meta-algorithm is actually a continuous-armed bandit problem [11, 85] where the set of arms lies in some bounded interval, in our case the different $\gamma \in [0, 1]$. Kleinberg [85] proposed a simple, yet worst-case optimal, strategy which consists in discretizing the strategy space into a finite set of n equally spaced points and running a standard bandit algorithm over those points. Unfortunately, their theoretical result holds only when the function to be optimize satisfies some Hölder conditions which may not verified for the Bayesian regret of UCB(γ). Still, that does not refrain us from using this strategy. The chosen number of arms n is critical in practice; set too low we may be far from the optimal solution and set too high we may end up exploring all the time. Auer, Ortner, and Szepesvári [11], with a similar algorithm, claimed a value $n = (J/\log J)^{1/3}$ is optimal without knowing the exact Hölder condition. We thus choose this specific discretization in our simulations. It has also been noted by Bayati, Hamidi, Johari, and Khosravi [17] that the Greedy algorithm, known to have a linear regret, ran with a sufficiently large number of arms may benefit from "free" exploration. This change point in its behavior happens around $\sqrt{J}$; we also evaluate Greedy with a discretization which contains that many points.

Once again we consider the same three scenarios for experiments. We set the number of episodes $J = 10000$ and we compare different meta-algorithms, namely Thompson Sampling (TS) with a uniform prior [6], AdaUCB [96] and the Greedy algorithm with the two discussed discretizations, denoted Greedy(100) for the discretization with 100 points. We also report an oracle meta-algorithm, which knows the optimal γ. Results are averaged over 100 iterations and are displayed on Figure 5.3.

The results are similar in the three scenarios We see that TS has a linear regret indicating that it fails to learn a good parameter γ in this time frame. Whereas the regret of AdaUCB is sublinear and the algorithm is thus learning; however it is outperformed for a relatively long period of time by a naive Greedy, which is stuck to a suboptimal, yet good arm. The most interesting part is that Greedy(100) performs extremely well; its regret is remarkably close to the one of Oracle

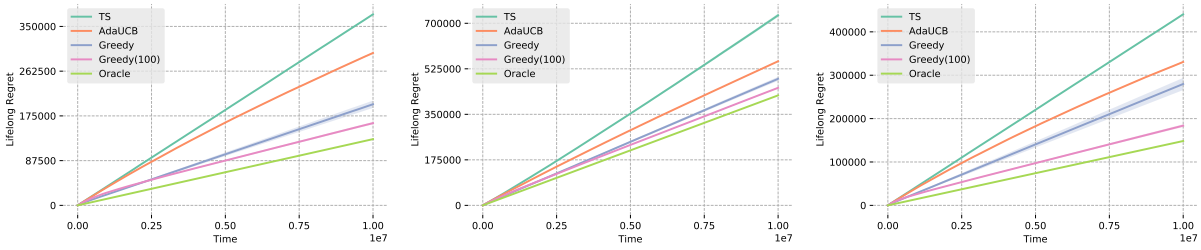


Figure 5.3: Lifelong regret of various meta-algorithms in stationary environments. Figures, from left to right, correspond respectively to Bernoulli bandits with a uniform prior, Gaussian bandits with a uniform prior and Bernoulli bandits with a Beta(1, 3) prior. Shaded areas show standard errors.

and is even sublinear. This supports the notion of free exploration for a large enough number of arms of the GREEDY algorithm.

5.6 LEARNING IN A NON-STATIONARY ENVIRONMENT

We now focus our attention to learning in a non-stationary environment, i.e. in which the prior distribution is not the same for all episodes. We will consider two scenarios: one where the environment changes abruptly and another where it slowly changes over time. Regrettably, adaptively tracking the prior distribution is intractable without making strong distributional assumptions, and potentially at the cost of forced exploration. Nonetheless, that does not mean that trying to improve over a naive algorithm is hopeless. Taking inspiration from the stochastic non-stationary bandit literature [59], we adapt the greedy index to take into account a changing environment.

The following two adjustments are based on the idea of "forgetting" rewards obtained long ago. The first one is based on discounting. Formally, let $\omega \in (0, 1)$ be the discount factor and define the discounted (D) greedy index

$$\widehat{\mu}_k^{\omega}(t) = \frac{\sum_{s=1}^{t} \omega^{t-s} X_k \mathbf{1}\{A_s = k\}}{\sum_{s=1}^{t} \omega^{t-s} \mathbf{1}\{A_s = k\}}.$$

The idea is to reduce the weight of rewards collected a long time ago, making the index more sensitive to recent payoffs. A similar approach, somewhat more sharp, called sliding-window (SW), simply discards rewards older than a parameter $\tau \in \mathbb{N}^{\star}$. Formally, the SW greedy index is defined as

$$\widehat{\mu}_k^{\tau}(t) = \frac{\sum_{s=t-\tau+1}^{t} X_k \mathbf{1}\{A_s = k\}}{\sum_{s=t-\tau+1}^{t} \mathbf{1}\{A_s = k\}}.$$

Both these indexes are similar to those of Discounted UCB and Sliding-Window UCB [59], the difference being that the exploration terms are removed. Unfortunately, there is no choice of ω and τ which guarantees strong performance, and they must be tuned empirically for the specific environment. Additionally, because both of these algorithms do not reflect on the whole horizon, the optimal discretization for the GREEDY algorithm is also more challenging to determinate. Consequently, in the following, we set the number of points in the discretization at a lesser value.

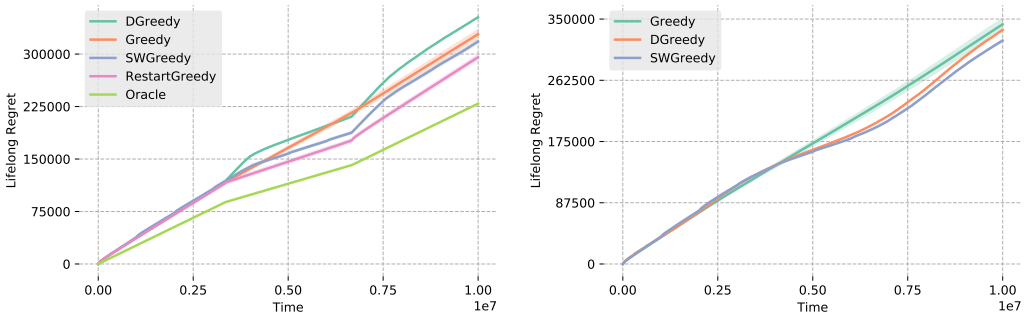


Figure 5.4: Lifelong regret of various meta-algorithms in non-stationary environments. Figures correspond respectively to an abruptly changing and a slowly changing environment. Shaded areas show standard errors.

In the following experiments, we run the GREEDY algorithm with the different indexes[1] on a discretization consisting of 21 equally-spaced points. In both scenarios, each episode is a Bernoulli bandit problem with $K = 10$ arms, the time horizon $T = 1000$ and the number of episodes is set at $J = 10000$. In the first scenario, we consider an abruptly changing environment, the prior distribution over the expected rewards of arms is a Beta$(1, 3)$ distribution for $j \leq J/3$ and $j \geq 2J/3$ and a Beta$(3, 1)$ distribution in between. While in the second scenario, we consider a slowly changing environment, the prior at episode j is a Beta$(2 + \cos(2\pi j/J + \pi), 2 + \cos(2\pi j/J))$ distribution. In the first scenario, we also report a policy which restart at times where the prior changes; this can be seen as an oracle we aim to emulate. Results are reported on Figure 5.4 and are averaged over 100 runs.

In both scenarios, DGREEDY and SWGREEDY are able to track the change in the prior distribution while GREEDY fails to do so. In the abruptly changing scenario, SWGREEDY outperforms DGREEDY, thus underlining the need to discard data from the previous prior. Regrettably, when the first prior comes back, both algorithms take a long time before finding again the previously learned solution; making GREEDY competitive in this experiment even if it completely missed the first change. Conversely in the slowly changing scenario, GREEDY fails to track the slow change in the prior distribution and thus accumulates a larger regret with in addition a broader variance. DGREEDY and SWGREEDY are roughly similar in this case, though SWGREEDY performs slightly better in the end. We would also like to mention that there is a hidden parameter in DGREEDY. Indeed, in the Greedy algorithm, each arm must have enough information for the greedy index to be defined. In GREEDY, and also in SWGREEDY, this is done by ensuring that each arm has been played at least once. In DGREEDY, this initialization is more challenging to determine and we believe it is partly responsible for the poor performance of DGREEDY. In both experiments, we arbitrarily set this value to 1.

[1] We choose $\omega = 0.9975$ and $\tau = 1000$ on both scenarios; these parameters have been roughly tuned.

5.7 Application to mortal bandits

Finally in this last section, we apply our method to a more realistic setting, the mortal bandit problem [32], where arms appear and disappear regularly (in particular, an arm is not always available contrary to the standard model). While the notion of episode may be more elusive in this setting, we show that synchronizing an episode according to the expected lifetime of arms can help to overcome this difficulty. We look upon the case where this value is known and the one where it is not and thus have to be estimated. We also have to modify the index of the UCB sub-algorithm to take into account that arms arrive at different times and for the unknown expected lifetime. Thereby, we replace the $\log(T)$ term by $\log(t - s_k + 1)$, where t and s_k denotes respectively the current and the arrival time steps for arm k. We then again pick the Greedy algorithm as meta-algorithm.

We consider the same setting as Chakrabarti, Kumar, Radlinski, and Upfal [32]. Although they mostly consider a large number of arms, we previously illustrated in Section 5.4 that this issue can be reduced via sub-sampling to a more tractable one, and with better performance as well. Therefore we focus our attention on problems with a small number of arms. In this setting, the number of arms remains fixed throughout the time horizon T, that is when an arm dies, it is immediately replaced by another one. The lifetime of arm k, denoted L_k, is drawn i.i.d. from a geometric distribution with expected lifetime L; this arm died after being available for L_k rounds. We also assume that arms are Bernoulli random variables. We consider two scenarios: in the first one, expected rewards of arms are drawn i.i.d. from a uniform distribution over $[0, 1]$, while in the second scenario they are drawn from a Beta$(1, 3)$ distribution. In both cases, we fix the number of arms $K = 5$ and the expected lifetime $L = 1000$. The horizon is set at $T = 1000L$, meaning that there are on average 1000 episodes throughout the time horizon. We compare several algorithms: the untuned UCB algorithm and its optimally tuned variant (OracleUCB), along with a tuned AdaptiveGreedy (AG) [32];[2] and our proposed methods, PeriodicUpdate-UCB (PU-UCB) and PeriodicEstimatedUpdate-UCB (PEU-UCB), where in the first one we assume the knowledge of the expected lifetime, while in the second it is estimated by the empirical lifetime of dead arms. On top of both our proposed policies lies a Greedy algorithm with $n = \sqrt{1000}$ arms. Results are averaged over 100 runs and are reported on Figure 5.5.

These experiments further underline the benefit of tuning the UCB algorithm for practical purposes. AG, a state-of-the-art algorithm in the mortal bandit setting, is clearly outperformed by OracleUCB. Additionally, PU-UCB and PEU-UCB, with somewhat similar performances on both instances, become rapidly better than AG, which is optimally tuned. We also remark once more that the regrets of both algorithms are sublinear, highlighting the performance of the Greedy meta-algorithm. Interestingly, PEU-UCB performs slightly better than PU-UCB on scenario 2, hinting at a potentially better decomposition of episodes.

[2] Both algorithms have been tuned with respect to the expected lifetime. On scenario 1, $\gamma = 0.25$ and $c = 1.5$, while on scenario 2, $\gamma = 0.25$ and $c = 2.5$.

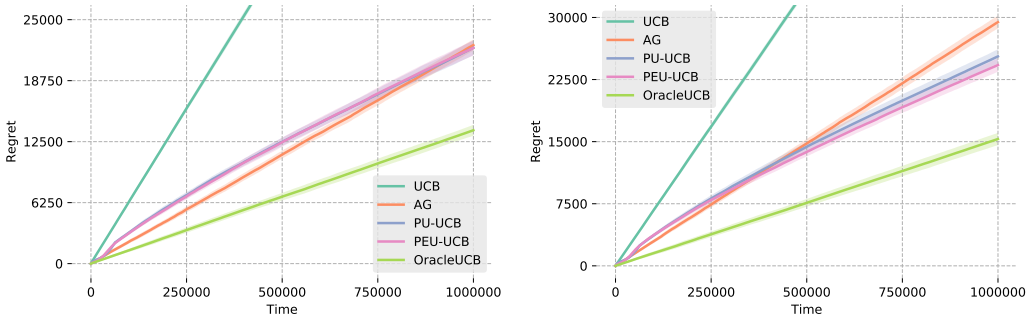


Figure 5.5: Regret of various algorithms in the mortal bandit setting. Figures correspond respectively to a uniform and a Beta(1, 3) priors. Shaded areas show standard errors. PU-UCB and PEU-UCB are almost identical in (a).

5.8 Conclusion

In this chapter, we have studied a lifelong learning problem in the multi-armed bandit framework where tasks arrive sequentially, sampled from a problem instance with some prior distribution. We first introduced our method which consists in optimizing a bandit algorithm, focusing on confidence interval width tuning of UCB-like policies. We then considered a bandit over bandit approach employing greedy algorithms as meta-algorithm and evaluated them in both stationary and non-stationary environments. Finally, we applied our method to a more realistic setting, the mortal bandit problem, by decomposing the time horizon into episodes according to the expected lifetime of arms and showed great empirical improvement compared to previous work.

Interesting directions The most prominent future work, especially in terms of practical applications, concerns the tracking of seasonal environments. It is crucial for building lifelong learning agents. Just recently this problem has been studied in the non-stationary bandit setting [35, 53].

Another direction may be the analysis of the Greedy algorithm in the continuous-armed bandit problem. It has been shown in a recent line a work that it enjoys great performances in several bandit frameworks [15, 17, 73, 125], and this setting is most likely one of them as shown indirectly in our experiments. Although it may be suboptimal, with high probability it concentrates quickly on arms with high expected rewards and thus works extremely well in practice. The next question is the "optimal" discretization for the greatest performance. This is the objective of the next chapter.

6 The greedy heuristic in multi-armed bandits

Contents

The Greedy algorithm is the simplest heuristic in sequential decision problem that carelessly takes the locally optimal choice at each round, disregarding any advantages of exploring and/or information gathering. Theoretically, it is known to sometimes have poor performances, for instance even a linear regret (with respect to the time horizon) in the standard multi-armed bandit problem. On the other hand, this heuristic performs reasonably well in practice and it even has sublinear, and even near-optimal, regret bounds in some very specific linear contextual and Bayesian bandit models. We also shown empirically in Chapter 5 that a greedy algorithm satisfies sublinear regret in the continuous-armed bandit problem.

We build on a recent line of work and investigate bandit settings where the number of arms is relatively large and where simple greedy algorithms enjoy highly competitive performance, both in theory and in practice. We first provide a generic worst-case bound on the regret of the Greedy algorithm. When combined with some arms subsampling, we prove that it verifies near-optimal

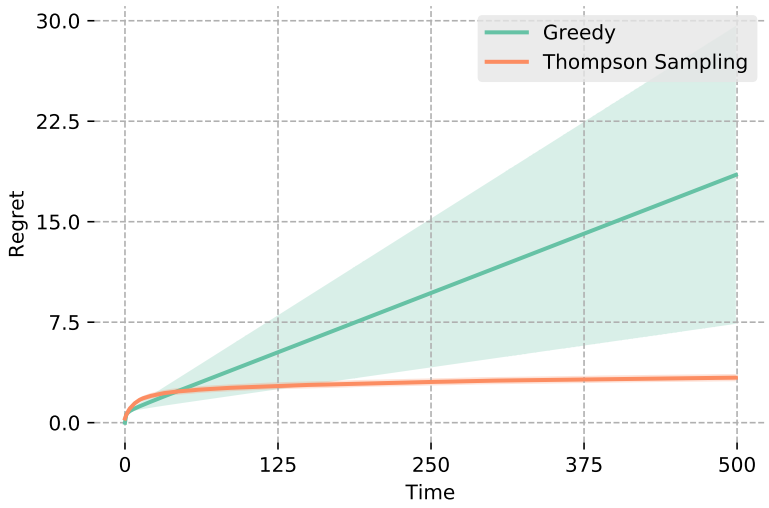


Figure 6.1: Regret of various algorithms as a function of time in a Bernoulli bandit problem. Results are averaged over 1000 runs and the shaded area represents 0.1 standard deviation.

worst-case regret bounds in continuous, infinite and many-armed bandit problems. Moreover, for shorter time spans, the theoretical relative suboptimality of Greedy is even reduced.

As a consequence, we subversively claim that for many interesting problems and associated horizons, the best compromise between theoretical guarantees, practical performances and computational burden is definitely to follow the greedy heuristic. We support our claim by many numerical experiments that show significant improvements compared to the state-of-the-art, even for moderately long time horizon.

6.1 Introduction

The exploration, although detrimental in the short term, is usually needed in the worst-case as it ensures that the learning algorithm "converges" to the optimal arm in the long run. On the other hand, the Greedy algorithm, an exploration-free strategy, focuses on pure exploitation and pulls the apparently best arm according to the information gathered thus far, at the risk of only sampling once the true optimal arm. This typically happens with Bernoulli rewards where only arms whose first reward is a 1 will be pulled again (and the others discarded forever). As a consequence, with some non-zero probability, the regret grows linearly with time as illustrated in the following example.

Example. Consider a relatively simple Bernoulli bandit problem consisting of $K = 2$ arms with expected rewards 0.9 and 0.1 respectively. With probability at least 0.01, Greedy fails to find the optimal arm. On the other hand, with probability 0.9^2 it suffers no regret after the initial pulls. This results in a linear regret with a large variance. This typical behavior is illustrated in Figure 6.1 in comparison to the Thompson Sampling algorithm [137].

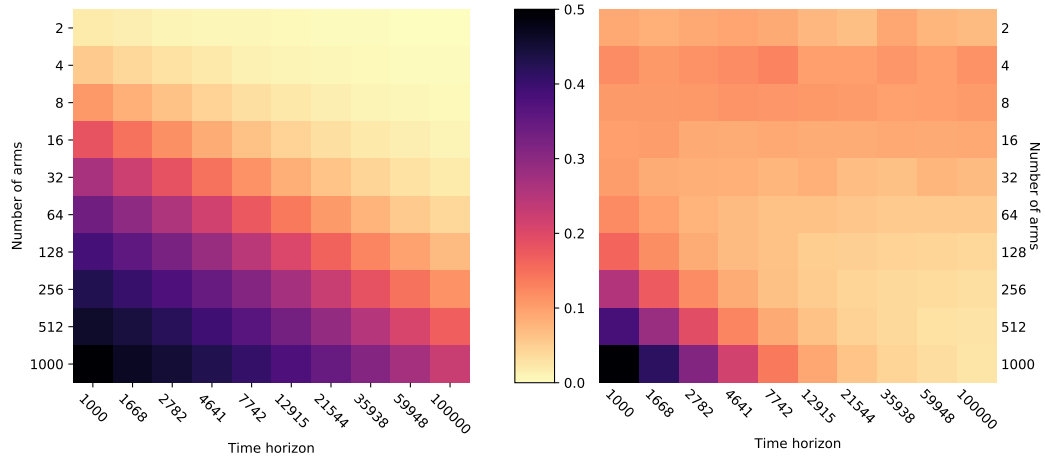


Figure 6.2: Bayesian regret divided by the horizon for UCB (left) and Greedy (right) as a function of the number of arms and the horizon in Gaussian bandit problems. Results are averaged over 500 runs.

Two solutions have been proposed to overcome this issue. The first one is to force the exploration; for example with an initial round-robin exploration phase [55], or by spreading the exploration uniformly over time à la Epsilon-Greedy [9]. However, both these algorithms need to know the different parameters of the problem to perform optimally (either to set the length of the round-robin phase or the value of ε), which represents a barrier to their use in practice. The second solution is to have a data-driven and adaptive exploration; for example, by adding an exploration term à la UCB [9], by using a Bayesian update à la Thompson Sampling [137], by using data- and arm-dependent stopping times for exploring à la Explore-Then-Commit [120, 121] or by tracking the number of pulls of suboptimal arms [14, 67, 68]. With careful tuning, these algorithms are asymptotically optimal for specific reward distributions. Yet this asymptotic regime can occur after a long period of time [58] and thus simpler heuristics might be preferable for relatively short time horizon [89, 144].

Conversely, the simple Greedy algorithm has recently been proved to satisfy near-optimal regret bounds in some linear contextual model [15, 73, 125] and a sublinear regret bound in some Bayesian many-armed setting [17]. In particular, this was possible because the Greedy algorithm benefits from "free" exploration when the number of arms is large enough. We illustrate this behavior in the following example.

Example. Consider bandit problems where rewards are Gaussian distributions with unit variance and mean rewards are drawn i.i.d. from a uniform distribution over $[0, 1]$. In Figure 6.2, we compare the regret of Greedy with the UCB algorithm for different number of arms and time horizon. For both algorithms, we observe a clear transition phase between problems with higher average regret (with darker colors) and problems with lower regret (with lighter colors). In this example, this transition takes the form of a diagonal.

This diagonal is much lower for Greedy compared to UCB, meaning that Greedy performs better in the problems in-between, and this in spite of UCB being optimal in the problem-dependent

sense (on the other hand, that is when the horizon is large, UCB outperforms Greedy). The intuition is that, when the number of near-optimal arms is large enough, Greedy rapidly converges to one of them while UCB is still in its initial exploration phase. The key argument here is the short time horizon relatively to the difficulty of the problem; we emphasis on the "relatively" as in practice the "turning point", that is the time horizon for which UCB performs better, can be extremely large.

Numerous interesting problems actually lie in the bottom left corner of Figure 6.2, i.e., bandit problems with a large number of arms and a relatively short time horizon and, as a consequence, the Greedy algorithm should be considered as a valid baseline.

Our results We first provide a generic regret bound on Greedy, and we illustrate how to derive worst-case regret bounds. We will then instantiate this regret bound to a uniformly sampled subset of arms and prove this satisfies near-optimal worst-case regret bounds in the continuous-armed, infinite-armed and many-armed bandit models. As a byproduct of our analysis, we get that the problem of unknown smoothness parameters can be overcome by a simple discretization depending only on the time horizon in the first of these models. In all these settings, we repeat the experiments of previous papers and show that the Greedy algorithm outmatches the state-of-the-art. We also present empirical results of Greedy in the linear, cascading, mortal and budgeted bandit models that further show its competitive performance against existing algorithms.

6.2 Related Work

The Greedy algorithm recently regained some attention in Bayesian bandit problems with a large but finite number of arms [17]. It performs extremely well empirically when the number of arms is large, sometimes better than "optimal" algorithms; in that case, the regret of Greedy is sublinear, though not optimal. In the following, we get rid of the strong Bayesian assumptions and we consider many different bandit models, where a subsampling technique is required and considered in the following.

Another recent success of Greedy is in linear contextual bandit problems, as it is asymptotically optimal for a two-armed contextual bandit with linear rewards when a covariate diversity condition holds [15]. This idea can be extended to rewards given by generalized linear models. If observed contexts are selected by an adversary, but perturbed by white noise, then Greedy can again have optimal regret guarantees [73]. Additional assumptions can even improved those results [125, 126]. Those results hold because exploration is not needed thanks to the diversity in the contexts. We do not believe this assumption is satisfied in many practical scenarios and we are therefore rather interested in the implicit exploration of Greedy. As a consequence, we shall no further consider the contextual framework (even if admittedly, our results could be generated via careful binning [120]). Interestingly, an extensive empirical study of contextual bandit algorithms found that Greedy is actually the second most efficient algorithm and is extremely close to the first one [21].

The Greedy algorithm has already been shown to enjoy great empirical performance in the continuous-armed bandit model [71]. In this chapter, we make formal this insight. Finally, we mention that in the one-dimensional linear bandit problem with a known prior distribution,

the cumulative regret of a greedy algorithm (under additional structural assumptions) admits an $\mathcal{O}(\sqrt{T})$ upper bound and its Bayes risk admits an $\mathcal{O}(\log T)$ upper bound [116]. Linear bandits are only considered empirically in this chapter (see Section 6.8.4).

We also provide, in Section 6.10, a short literature review on the different bandit settings studied in this chapter.

6.3 Preliminaries

In the stochastic multi-armed bandit model, a learning agent interacts sequentially with a finite set of K distributions $\nu_1, \dots, \nu_K$, called arms. At round $t \in \mathbb{N}$, the agent chooses an arm A_t, which yields a stochastic reward X_t drawn from the associated probability distribution ν_{A_t}. The objective is to design a sequential strategy maximizing the expected cumulative reward up to some time horizon T. Let $\mu_1, \dots, \mu_K$ denote the mean rewards of arms, and $\mu^\star := \max_{k \in [K]} \mu_k$ be the best mean reward. The goal is equivalent to minimizing the regret, defined as the difference between the expected reward accumulated by the oracle strategy always playing the best arm at each round, and the one accumulated by the strategy of the agent,

$$\mathbb{E}[R_T] = \mathbb{E}\left[\sum_{t=1}^{T} (\mu^\star - X_t)\right] = T\mu^\star - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}\right]$$

where the expectation is taken with respect to the randomness in the sequence of successive rewards from each arm and the possible randomization in the strategy of the agent. Let $N_k(T)$ be the number of pulls of arm k at the end of round T and define the suboptimality gap of an arm $k \in [K] := \{1, \dots, K\}$ as $\Delta_k = \mu^\star - \mu_k$. The expected regret is equivalently written as

$$\mathbb{E}[R_T] = \sum_{k=1}^{K} \Delta_k \mathbb{E}[N_k(T)] \,.$$

The Greedy algorithm Summarized in Algorithm 11, Greedy is probably the simplest and the most obvious algorithm. Given a set of K arms, at each round t, it pulls the arm with the highest average reward

$$\widehat{\mu}_k(t-1) = \frac{1}{N_k(t-1)} \sum_{s=1}^{t-1} X_s \mathbf{1}\{A_s = k\}$$

with the convention that $0/0 = \infty$, so that the first K pulls initialize each counter. Thus, Greedy constantly exploits the best empirical arm.

In the rest of the paper, unless stated otherwise, we assume that the stochastic reward X_t takes the form $X_t = \mu_{A_t} + \eta_t$ where $\{\eta_t\}_{t=1}^T$ are i.i.d. 1-subgaussian white noise and that μ_k are bounded for all $k \in [K]$, $\mu_k \in [0, 1]$ without loss of generality. We further assume the knowledge of the time horizon T, unknown time horizon can be handled as usual in bandit problems [20]. Finally, we say that arm k is ε-optimal for some $\varepsilon > 0$ if $\mu_k \geq \mu^\star - \varepsilon$.

Algorithm 11: GREEDY

Input: Number of arms K
for $t \leftarrow 1$ **to** K **do**
  Pull arm $A_t = t$ `// Initialization`
for $t \leftarrow K+1$ **to** $\dots$ **do**
  Pull arm $A_t \in \arg\max_{k\in[K]} \widehat{\mu}_k(t-1)$ `// Exploitation`

6.4 Generic bounds on Greedy

We now present the generic worst-case regret bound on GREEDY that we will use to derive near-optimal bounds in several bandit models.

Theorem 6.1. *The regret of GREEDY verifies for all $\varepsilon > 0$*

$$\mathbb{E}[R_T] \leq T\exp\left(-N_\varepsilon \frac{\varepsilon^2}{2}\right) + 3\varepsilon T + \frac{6K}{\varepsilon} + \sum_{k=1}^{K} \Delta_k$$

where N_ε denotes the number of ε-optimal arms.

Remark. This bound generalizes a Bayesian analysis [17]. It is slightly looser; indeed the Bayesian assumption can be used to bound N_ε and further improve the third term by bounding the number of suboptimal arms. Those techniques usually do not work in the stochastic setting.

Proof. The proof combines two techniques standard in the literature: creating a "good" event in order to distinguish the randomness of the distributions from the behavior of the algorithm and decomposing the arms into near-optimal and suboptimal ones. Fix some $\varepsilon > 0$.

GOOD EVENT Define the event E, through its complement, by

$$E^c = \bigcap_{k:\Delta_k \leq \varepsilon} \left\{\exists t \,|\, \widehat{\mu}_k(t) \leq \mu_k - \varepsilon\right\}.$$

In words, E is the event that at least one ε-optimal arm is never underestimated by more than ε below its mean reward. Using the independence of the events along with the concentration bound of [see 17, Lemma 2], we obtain

$$\mathbb{P}(E^c) \leq \exp\left(-N_\varepsilon \frac{\varepsilon^2}{2}\right). \tag{6.1}$$

BOUND ON THE NUMBER OF PULLS OF SUBOPTIMAL ARMS On the event E, let $k \in [K]$ be an arm such that $\Delta_k > 3\varepsilon$. With a slight abuse of notation, we denote by $\widehat{\mu}_k^t$ the average reward of arm k after t samples. The expected number of pulls of arm k is then bounded by

$$\begin{aligned}
\mathbb{E}[N_k(T) \mid E] &\leq 1 + \sum_{t=1}^{\infty} \mathbb{P}\big(\widehat{\mu}_k^t \geq \mu^\star - 2\varepsilon\big) \\
&\leq 1 + \sum_{t=1}^{\infty} \mathbb{P}\big(\widehat{\mu}_k^t - \mu_k \geq \Delta_k - 2\varepsilon\big) \\
&\leq 1 + \sum_{t=1}^{\infty} \exp\left(-t\frac{(\Delta_k - 2\varepsilon)^2}{2}\right) \\
&\leq 1 + \frac{1}{\exp\left(\frac{(\Delta_k - 2\varepsilon)^2}{2}\right) - 1} \\
&\leq 1 + \frac{2}{(\Delta_k - 2\varepsilon)^2} \qquad (6.2)
\end{aligned}$$

where in second inequality we used that $\widehat{\mu}_k^t$ is $1/t$-subgaussian and in the last inequality we used that $e^x \geq 1 + x$ for all $x \in \mathbb{R}$.

PUTTING THINGS TOGETHER We first decompose the regret according to the event E

$$\mathbb{E}[R_T] \leq \mathbb{E}[R_T|E^c]\mathbb{P}(E^c) + \mathbb{E}[R_T|E]\,. \qquad (6.3)$$

As mean rewards are bounded in $[0, 1]$, the regret on the bad event is bounded by T and by Equation (6.1) we have

$$\mathbb{E}[R_T|E^c]\mathbb{P}(E^c) \leq T\exp\left(-N_\varepsilon\frac{\varepsilon^2}{2}\right).$$

We further decompose the second term on the right-hand side of Equation (6.3),

$$\mathbb{E}[R_T|E] \leq \sum_{k:\Delta_k \leq 3\varepsilon} \Delta_k \mathbb{E}[N_k(T)|E] + \sum_{k:\Delta_k > 3\varepsilon} \Delta_k \mathbb{E}[N_k(T)|E]\,.$$

The first term is trivially bounded by $3\varepsilon T$, while for the second term we have by Equation (6.2),

$$\begin{aligned}\sum_{k:\Delta_k>3\varepsilon} \Delta_k \mathbb{E}[N_k(T)|E] &\leq \sum_{k:\Delta_k>3\varepsilon} \frac{2\Delta_k}{(\Delta_k - 2\varepsilon)^2} + \sum_{k=1}^{K} \Delta_k \\ &\leq \sum_{k:\Delta_k>3\varepsilon} \frac{6}{(\Delta_k - 2\varepsilon)} + \sum_{k=1}^{K} \Delta_k \\ &\leq \sum_{k:\Delta_k>3\varepsilon} \frac{6}{\varepsilon} + \sum_{k=1}^{K} \Delta_k \\ &\leq \frac{6K}{\varepsilon} + \sum_{k=1}^{K} \Delta_k\end{aligned}$$

where in the second inequality we used that $\Delta_k \leq 3(\Delta_k - 2\varepsilon)$, which holds true since $\Delta_k \geq 3\varepsilon$. Hence the result.

□

It is easy to see that this bound is meaningless when N_ε is independent of T as one of the first two terms will, at least, be linear with respect to T. On the other hand, N_ε has no reason to depend on the time horizon. The trick to obtain sublinear regret will be to lower bound N_ε by a function of the number of arms K, then to optimize K with respect to the time horizon T. To motivate this, consider the following example.

Example. Consider a problem with a huge number of arms n with mean rewards drawn i.i.d. from a uniform distribution over $[0, 1]$. In that specific case, we roughly have $N_\varepsilon \approx \varepsilon K$ for some subset of arms, chosen uniformly at random, with cardinality K. Taking $\varepsilon = \left(\frac{\log T}{K}\right)^{1/3}$, so that the first term in the generic bound is sublinear, yields a $\mathcal{O}\left(\max\left\{T\left(\frac{\log T}{K}\right)^{1/3}, K\left(\frac{K}{\log T}\right)^{1/3}\right\}\right)$ regret bound, which comes from the second and third terms respectively. If we subsampled $K = T^{3/5}(\log T)^{2/5}$ arms, so that the maximum is minimized, the regret bound becomes $\mathcal{O}\left(T^{4/5}(\log T)^{1/5}\right)$; in particular it is sublinear.

This argument motivates this paper and will be made formal in subsequent sections. Though this does not lead to optimal bounds – as expected by the essence of the greedy heuristic in the multi-armed bandit model –, it will nonetheless be highly competitive for short time span in many practical bandit problems.

It is possible to theoretically improve the previous result by using a chaining/peeling type of argument. Unfortunately, it is not practical to derive better explicit guarantees as it involves an integral without close form expressions.

Corollary 6.1. *The regret of* Greedy *verifies*

$$\mathbb{E}[R_T] \leq \min_{\varepsilon} \left\{ 3\varepsilon T + \frac{6K}{\varepsilon} + \int_{\varepsilon}^{1} \left(3T + \frac{6K}{x^2} \right) \exp\left(-N_x \frac{x^2}{2} \right) dx \right\} + T \exp\left(-\frac{K}{2} \right) + \sum_{k=1}^{K} \Delta_k \,.$$

Proof. We recall the definition of the event E_ε, through its complement E_ε^c,

$$E_\varepsilon^c = \bigcap_{k:\Delta_k \leq \varepsilon} \mathbb{P}(\exists t \,|\, \widehat{\mu}_k(t) \leq \mu_k - \varepsilon) \,.$$

Consider any increasing sequence $\{\varepsilon_m\}_{m=0}^M$ and denote E_m the good event associated with ε_m for $m \in \{0, \ldots, M\}$. By the chain rule and the previous computation of the regret on the good event (see proof of Theorem 6.1), we have

$$\begin{aligned}
\mathbb{E}[R_T] &\leq \left(3\varepsilon_0 T + \frac{6K}{\varepsilon_0} \right) \mathbb{P}(E_0) + \left(3\varepsilon_1 T + \frac{6K}{\varepsilon_1} \right) \mathbb{P}(E_1 \cap E_0^c) + \ldots \\
&\quad + \left(3\varepsilon_M T + \frac{6K}{\varepsilon_M} \right) \mathbb{P}(E_M \cap E_{M-1}^c) + T\mathbb{P}(E_{M-1}^c) + \sum_{k=1}^{K} \Delta_k \\
&\leq \left[\left(3\varepsilon_0 T + \frac{6K}{\varepsilon_0} \right) - \left(3\varepsilon_1 T + \frac{6K}{\varepsilon_1} \right) \right] \mathbb{P}(E_0) + \ldots \\
&\quad + \left[\left(3\varepsilon_{M-1} T + \frac{6K}{\varepsilon_{M-1}} \right) - \left(3\varepsilon_M T + \frac{6K}{\varepsilon_M} \right) \right] \mathbb{P}(E_{M-1}) \\
&\quad + \left(3\varepsilon_M T + \frac{6K}{\varepsilon_M} \right) \mathbb{P}(E_M) + T\mathbb{P}(E_M^c) + \sum_{k=1}^{K} \Delta_k
\end{aligned}$$

where in the second inequality we used that $\mathbf{1}\{\mathfrak{A} \cap \mathfrak{B}^c\} = \mathbf{1}\{\mathfrak{A}\} - \mathbf{1}\{\mathfrak{B}\}$ if $\mathfrak{B} \subset \mathfrak{A}$. In the proof of Theorem 6.1, we show that

$$\mathbb{P}(E_m^c) \leq \exp\left(-N_{\varepsilon_m} \frac{\varepsilon_m^2}{2} \right)$$

for $m \in \{0, \ldots, M\}$. Hence we obtain

$$\begin{aligned}
R(T) &\leq \left(3\varepsilon_0 T + \frac{6K}{\varepsilon_0} \right) \\
&\quad + \sum_{m=0}^{M-1} \left[\left(3\varepsilon_{m+1} T + \frac{6K}{\varepsilon_{m+1}} \right) - \left(3\varepsilon_m T + \frac{6K}{\varepsilon_m} \right) \right] \exp\left(-N_{\varepsilon_m} \frac{\varepsilon_m^2}{2} \right) \\
&\quad + T \exp\left(-\frac{K}{2} \right) + \sum_{k=1}^{K} \Delta_k
\end{aligned}$$

The middle term is upper-bounded by

$$\sum_{m=0}^{M-1} (\varepsilon_{m+1} - \varepsilon_m) \left[3T + \frac{6K}{\varepsilon_m^2} \right] \exp\left(-N_{\varepsilon_m} \frac{\varepsilon_m^2}{2} \right),$$

which converges, as the mesh of the sequence ε_m goes to zero, towards

$$\int_{\varepsilon}^{1} \left(3T + \frac{6K}{x^2} \right) \exp\left(-N_x \frac{x^2}{2} \right) dx$$

Hence the result. □

6.5 Continuous-armed bandits

We first study Greedy in the continuous-armed bandit problem. We recall that in this model, the number of actions is infinitely large. Formally, let $\mathcal{A}$ be an arbitrary set and $\mathcal{F}$ a set of functions from $\mathcal{A} \to \mathbb{R}$. The learner is given access to the action set $\mathcal{A}$ and function class $\mathcal{F}$. In each round t, the learner chooses an action $A_t \in \mathcal{A}$ and receives reward $X_t = f(A_t) + \eta_t$, where η_t is some noise and $f \in \mathcal{F}$ is fixed, but unknown. As usual in the literature [11, 65, 85], we restrict ourselves to the case $\mathcal{A} = [0, 1]$, η_t is 1-subgaussian, f takes values in $[0, 1]$ and $\mathcal{F}$ is the set of all functions that satisfy an Hölder condition around the maxima. Formally,

Assumption 6.1. *There exist constants $L \geq 0$ and $\alpha > 0$ such that for all $x \in [0, 1]$,*

$$f(x^\star) - f(x) \leq L \cdot |x^\star - x|^\alpha$$

where $x^\star$ denotes the optimal arm.

This assumption captures the degree of continuity at the maxima and it is needed to ensure that this maxima is not reached at a sharp peak.

Similarly to CAB1 [85], the Greedy algorithm will work on a discretization of the action set into a finite set of K equally spaced points $\{1/K, 2/K, \ldots, 1\}$. Each point is then considered as an arm and we can apply the standard Greedy algorithm on them.

Remark. The same analysis holds if it chooses a point uniformly at random from the chosen interval $\left[\frac{k-1}{K}, \frac{k}{K}\right]$ for $1 \leq k \leq K$, see also Auer, Ortner, and Szepesvári [11].

The problem is thus to set the number of points K. The first regret bound on the Greedy algorithm assume that the smoothness parameters are known.

Theorem 6.2. *If $f : [0, 1] \to [0, 1]$ satisfies Assumption 6.1, then for a subsampling of $K \geq \left(\frac{L}{\varepsilon}\right)^{1/\alpha}$ arms, the regret of the Greedy algorithm verifies for all $\varepsilon > 0$*

$$\mathbb{E}[R_T] \leq T \exp\left(-\frac{K}{2L^{1/\alpha}} \varepsilon^{2+1/\alpha} \right) + 4\varepsilon T + \frac{6K}{\varepsilon} + K\,.$$

In particular, the choice

$$K = \left(\frac{2}{3}\right)^{\alpha/(4\alpha+1)} \left(\frac{4}{3}\right)^{2\alpha/(4\alpha+1)} L^{2/(4\alpha+1)} T^{(2\alpha+1)/(4\alpha+1)} (\log T)^{2\alpha/(4\alpha+1)}$$

yields for $L \leq \sqrt{\frac{3}{2T}} K^{\alpha+1/2}$,

$$\mathbb{E}[R_T] \leq 13 L^{2/(4\alpha+1)} T^{(3\alpha+1)/(4\alpha+1)} (\log T)^{2\alpha/(4\alpha+1)} + 1\,.$$

Proof. Let $\varepsilon > 0$. The regret can be decomposed into an approximation and an estimation term,

$$Tf(x^\star) - \sum_{t=1}^{T} f(x_t) = T\left(f(x^\star) - \max_{k\in[K]} f\left(\frac{k}{K}\right)\right) + \left(T \max_{k\in[K]} f\left(\frac{k}{K}\right) - \sum_{t=1}^{T} f(x_t)\right)$$

From Assumption 6.1, the first term is bounded by εT when $K \geq \left(\frac{L}{\varepsilon}\right)^{1/\alpha}$. Then, according to Theorem 6.1, we just have to lower bound N_ε to conclude the proof. To do so, we begin by proving a lower bound on the number of arms that are ε-optimal with respect to the best arm overall. Let N_ε^C denotes this quantity.

Bound on N_ε^C From Assumption 6.1, an ε-optimal arm k may verify (there can be ε-optimal that are not around the maxima)

$$L\left|x^\star - \frac{k}{K}\right|^\alpha \leq \varepsilon$$

Rearranging the terms and using that k is an integer, we obtain

$$\left\lceil K\left(x^\star - \left(\frac{\varepsilon}{L}\right)^{1/\alpha}\right)\right\rceil \leq k \leq \left\lfloor K\left(x^\star + \left(\frac{\varepsilon}{L}\right)^{1/\alpha}\right)\right\rfloor$$

This means that we have the following lower bound on N_ε^C

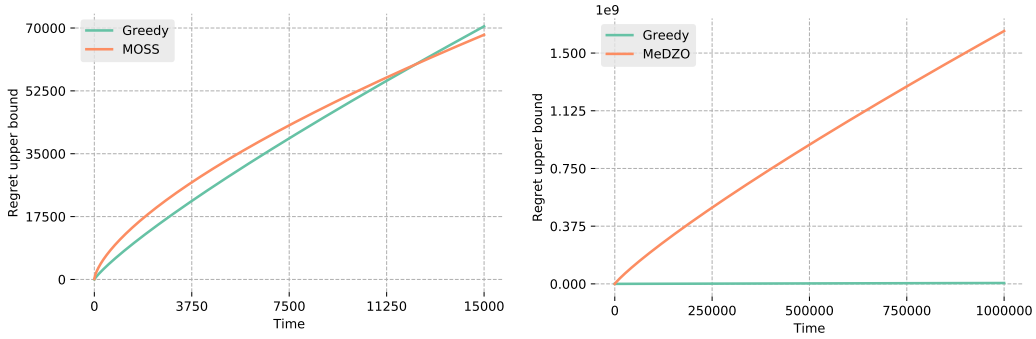
$$N_\varepsilon^C \geq \left\lfloor K\left(x^\star + \left(\frac{\varepsilon}{L}\right)^{1/\alpha}\right)\right\rfloor - \left\lceil K\left(x^\star - \left(\frac{\varepsilon}{L}\right)^{1/\alpha}\right)\right\rceil + 1$$

Thanks to Lemma 6.1, we obtain

$$N_\varepsilon^C \geq \left\lfloor 2K\left(\frac{\varepsilon}{L}\right)^{1/\alpha}\right\rfloor$$

Finally, using that $\lfloor 2x \rfloor \geq x$ for $x \geq 1$ (easily verify with the assumption on K), we obtain the following lower bound

$$N_\varepsilon^C \geq K\left(\frac{\varepsilon}{L}\right)^{1/\alpha}$$



(a) Known smoothness

(b) Unknown smoothness

Figure 6.3: Regret upper bound of various algorithms as a function of time in the continuous-armed bandit model with smoothness parameters $L = 1$ and $\alpha = 1$.

Conclusion We trivially have that $N_\varepsilon \geq N_\varepsilon^C$. The first part of the Theorem then results from the fact that $\sum_{k=1}^K \Delta_k \leq K$ since $\mu_k \in [0,1]$ for all $k \in [K]$.

On the other hand, the second part comes from taking $\varepsilon^2 = 3K/(2T)$ which is the value of ε that minimizes the term $4\varepsilon T + 6K/\varepsilon$. □

This bound is sublinear with respect to the time horizon T, yet suboptimal. Indeed, the lower bound in this setting is $\Omega\big(T^{(\alpha+1)/(2\alpha+1)}\big)$ and the MOSS algorithm run on a optimal discretization attains it since its regret scales, up to constant factor, as $\mathcal{O}\big(L^{1/(2\alpha+1)}T^{(\alpha+1)/(2\alpha+1)}\big)$ [65]. Yet, as mentioned previously, Greedy is theoretically competitive for short time horizon due to small constant factors. In Figure 6.3a, we displayed regret upper bounds of MOSS and Greedy as a function of time for functions that satisfy Assumption 6.1 with smoothness parameters $L = 1$ and $\alpha = 1$. We see that the bound on Greedy is stronger up until a moderate time horizon $T \approx 12000$.

Of course, assuming that the learner knows smoothness parameters α and L is often unrealistic. If we want to ensure a low regret on very regular functions, by taking $\alpha \to \infty$, we have the following corollary.

Corollary 6.2. *If $f : [0,1] \to [0,1]$ satisfies Assumption 6.1, then for a subsampling of $K = \sqrt{\frac{4}{3}T\log T}$ arms, the regret of Greedy verifies for $L \leq 3^{1/4}(4/3)^{(2\alpha+1)/4}T^{2\alpha}(\log T)^{(\alpha+1)/2}$,*

$$\mathbb{E}[R_T] \leq 15\max\{L^{1/(2\alpha+1)}, L^{-1/(2\alpha+1)}\}T^{(3\alpha+2)/(4\alpha+2)}\sqrt{\log T} + 1\,.$$

Proof. It is a direct consequence of Theorem 6.2 with $\varepsilon = \left(L^{1/\alpha}\sqrt{\frac{3\log T}{T}}\right)^{\alpha/(2\alpha+1)}$. □

Once again, Greedy attains a sublinear, yet suboptimal, regret bound. In the case of unknown smoothness parameters, the regret lower bound is $\Omega\big(L^{1/(1+\alpha)}T^{(\alpha+2)/(2\alpha+2)}\big)$ [108], which is attained by MeDZO with a $\mathcal{O}\Big(L^{1/(\alpha+1)}T^{(\alpha+2)/(2\alpha+2)}(\log_2 T)^{3/2}\Big)$ regret bound [65]. This time, Greedy also has a lower polynomial dependency which makes it even more competitive

theoretically. In Figure 6.3b, we displayed regret upper bounds of MeDZO and Greedy (with unknown smoothness parameters) as a function of time for functions that satisfy Assumption 6.1 with smoothness parameters $L = 1$ and $\alpha = 1$. Here we cannot see the turning point since Greedy is stronger up until an extremely large time horizon $T \approx 1,9 \cdot 10^{46}$. Our numerical simulations will further support this theoretical advantage.

6.6 Infinite-armed bandits

We now study the infinite-armed bandit problem. In this setting, we consider the general model of Wang, Audibert, and Munos [148]. In particular they assume a margin condition on the mean reward of a randomly drawn arm. Formally,

Assumption 6.2. *There exist $\mu^\star \in (0, 1]$ and $\beta > 0$ such that the expected reward μ of a randomly drawn arm satisfies*

$$\mathbb{P}(\mu > \mu^\star - \varepsilon) = \mathcal{O}\Big(\varepsilon^\beta\Big), \text{for } \varepsilon \to 0\,.$$

Equivalently, there exist $c_1 > 0$ and $c_2 > 0$ such that

$$c_1\varepsilon^\beta \leq \mathbb{P}(\mu > \mu^\star - \varepsilon) \leq c_2\varepsilon^\beta\,.$$

Similarly to UCB-F [148], Greedy will consist of initially choosing K arms and then running the standard Greedy algorithm on those arms. The problem is then to choose the optimal number of arms K. The following regret bound on the Greedy algorithm assume the knowledge of the parameter β and c_1.

Theorem 6.3. *Assume Assumption 6.2 of the model. The regret of the Greedy algorithm verifies for any subsampling of $K > 0$ arms and for all $\varepsilon > 0$*

$$\mathbb{E}[R_T] \leq T\left[\exp\left(-\frac{c_1}{4}K\varepsilon^{2+\beta}\right) + \exp\left(-\frac{c_1}{8}K\varepsilon^\beta\right)\right] + 4\varepsilon T + \frac{6K}{\varepsilon} + K\,.$$

In particular, the choice

$$K = \left(\frac{2}{3}\right)^{(2+\beta)/(4+\beta)}\left(\frac{8}{c_1(4+\beta)}\right)^{2/(4+\beta)} T^{(2+\beta)/(4+\beta)}(\log T)^{2/(4+\beta)}$$

yields

$$\mathbb{E}[R_T] \leq 20(c_1(4+\beta))^{-2/(4+\beta)}T^{(3+\beta)/(4+\beta)}(\log T)^{2/(4+\beta)}\,.$$

Proof. Let $\varepsilon > 0$. Once again, thanks to Theorem 6.1 we just have to bound N_ε and the result will follow by adding the approximation cost εT.

We construct a good event on the expected rewards of sampled arms. Let $I_\varepsilon = [\mu^\star - \varepsilon, \mu^\star]$ and $N_\varepsilon^I = \sum_{k=1}^K \mathbf{1}\{k \in I_\varepsilon\}$ be the number of ε-optimal arms with respect to all arms. Assumption 6.2 implies that

$$p = \mathbb{E}[\mathbf{1}\{k \in I_\varepsilon\}] = \mathbb{P}(k \in I_\varepsilon) \in [c_1\varepsilon^\beta, c_2\varepsilon^\beta]$$

Let $\delta \in [0,1)$. By Chernoff inequality we have

$$\mathbb{P}\big(N_\varepsilon^I < (1-\delta)Kp\big) \leq \exp\big(-Kp\delta^2/2\big)$$

In particular, taking $\delta = \frac{1}{2}$ yields

$$\mathbb{P}\Big(N_\varepsilon^I < c_1\varepsilon^\beta K/2\Big) \leq \exp\Big(-c_1\varepsilon^\beta K/8\Big)$$

Now we trivially have that $N_\varepsilon \geq N_\varepsilon^I$, and hence we obtain

$$\mathbb{P}\Big(N_\varepsilon < c_1\varepsilon^\beta K/2\Big) \leq \exp\Big(-c_1\varepsilon^\beta K/8\Big)$$

By constructing a good event based on the previous concentration bound and using $\sum_{k=1}^K \Delta_k \leq K$, we obtain the first part of the Theorem.

The second part results from (i) the first exponential term dominates since $\varepsilon^{2+\beta} \leq \varepsilon^\beta$ for all $\varepsilon \in [0,1]$ and $\beta > 0$ and (ii) the choice of $\varepsilon = \sqrt{3K/(2T)}$ which is the value that minimizes $4\varepsilon T + 6K/\varepsilon$. □

In comparison, the lower bound is this model is $\Omega\big(T^{\beta/(1+\beta)}\big)$ for any $\beta > 0$ and $\mu^\star \leq 1$ and UCB-F obtained a $\mathcal{O}\big(T^{\beta/(\beta+1)}\log T\big)$ regret bound in the case $\mu^\star = 1$ or $\beta > 1$ and a $\widetilde{\mathcal{O}}\big(T^{1/2}\big)$ bound otherwise [148]. The regret of Greedy is once again sublinear, though suboptimal, with a lower logarithmic dependency. Our numerical simulations will further emphasis its competitive performance.

The case of unknown parameters is more complicated to handle compared to the continuous-armed model and is furthermore not the main focus of this paper. A solution proposed by Carpentier and Valko [31] nonetheless, is to perform an initial phase to estimate the parameter β.

6.7 Many-armed bandits

We now consider the particular model of many-armed bandit problem of Zhu and Nowak [154]. It is somehow related to the previous two except it also takes into account the time horizon. In particular, it focuses on the case where multiple best arms are present. Formally, let T be the time horizon, n be the total number of arms and m be the number of best arms. We emphasis that n can be arbitrary large and m is usually unknown. The following assumption will lower bound the number of best arms.

Assumption 6.3. *There exists $\gamma \in [0,1]$ such that the number of best arms satisfies*

$$\frac{n}{m} \leq T^\gamma\,.$$

We assume that the value γ (or at least some upper-bound) is known in our case, even though adaptivity to it is possible [154]. The following Theorem bounds the regret of a Greedy algorithm that initially subsamples a set of arms.

Theorem 6.4. *Assume Assumption 6.3 of the model and that the number of arms n is large enough for the following subsampling schemes to be possible. Depending on the value of α and the time horizon T, it holds*

- *If $T^{1-3\alpha} \leq \log T$, in particular for $\alpha \geq \frac{1}{3}$ and $T \geq 2$, choosing $K = 2T^{2\alpha}\log T$ leads to*
$$\mathbb{E}[R_T] \leq 14T^{\alpha+1/2}\log T + 2\,.$$
- *Otherwise, the choice of $K = 2\sqrt{T^{1+\alpha}\log T}$ yields*
$$\mathbb{E}[R_T] \leq 14T^{(3+\alpha)/4}\sqrt{\log T} + 2\,.$$

Proof. Again we just need a lower bound on the number of optimal arms in the subsampling and we construct a good event to do so. We reuse the previous notation N_ε to denote this value ($\varepsilon = 0$ here). Let N_ε^S be the number of optimal arms with respect to all arms. In the case of a subsampling of K arms done without replacement, N_ε^S is distributed according to a hypergeometric distribution.

By Hoeffding's inequality, we have for $0 < t < pK$

$$\mathbb{P}\big(N_\varepsilon^S \leq (p-t)K\big) \leq \exp\big(-2t^2K\big)$$

where $p = m/n$. We want to choose t such $p - t > 0$ otherwise the bound is meaningless. In particular, the choice of $t = \frac{p}{2}$ yields

$$\mathbb{P}\left(N_\varepsilon^S \leq \frac{pK}{2}\right) \leq \exp\left(-\frac{p^2K}{2}\right)$$

We then trivially have that $N_\varepsilon \geq N_\varepsilon^S$. The regret on the bad events is then given by

$$T\left[\exp\left(-\frac{pK}{2}\frac{\varepsilon^2}{2}\right) + \exp\left(-\frac{p^2K}{2}\right)\right]$$

For the regret to be $\mathcal{O}(1)$ on the bad events, the two following inequalities must be verify

$$\begin{aligned}\frac{pK}{2}\frac{\varepsilon^2}{2} &\geq \log T\\ \frac{p^2K}{2} &\geq \log T\end{aligned}$$

Now the term $3\varepsilon T + \frac{6K}{\varepsilon}$ of Theorem 6.1 is minimized for $\varepsilon^2 = 2K/T$. This leads to

$$\begin{aligned}\frac{pK^2}{2} &\geq T\log T\\ \frac{p^2K}{2} &\geq \log T\end{aligned}$$

Using that $p = T^{-\alpha}$, we obtain

$$K \geq 2 \max\left\{\sqrt{T^{1+\alpha}}, T^{2\alpha}\sqrt{\log T}\right\}\sqrt{\log T}$$

The proof is concluded by decomposing according to the value inside the max term. □

The previous bounds indicate that Greedy realizes a sublinear worst-case regret on the standard multi-armed bandit problem at the condition that the number of arms is large and the proportion of near-optimal arms is high enough. To compare, the MOSS algorithm run on an optimal subsampling achieves a $\mathcal{O}\left(T^{(1+\gamma)/2}\log T\right)$ regret bound for all $\gamma \in [0,1]$, which is optimal up to logarithmic factors [154]. In this case, our numerical simulation will show that Greedy is competitive even when the setup is close to the limit of the theoretical guarantee of Greedy.

6.8 Experiments

We now evaluate Greedy in the previously studied bandit models to highlight its practical competitive performance. For fairness reasons with respect to the other algorithms, and in the idea of reproducibility, we will not create new experiment setups but reproduce experiments that can be found in the literature (and compare the performances of Greedy with respect to state of the art algorithms).

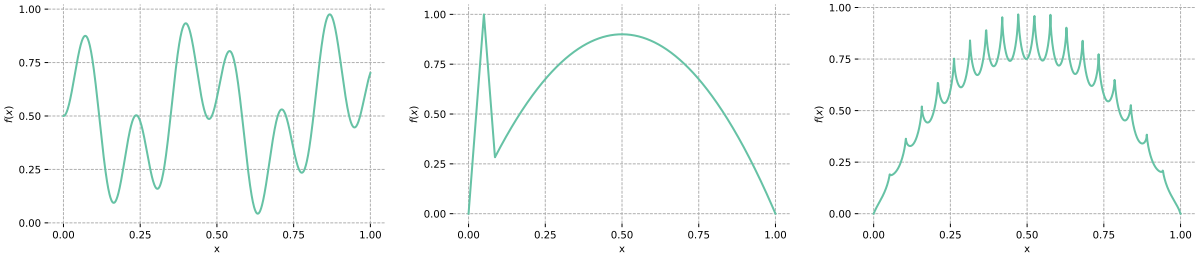
6.8.1 Continuous-armed bandits

In the continuous-armed bandit setting, we repeat the experiments of Hadiji [65]. We consider three functions that are gradually sharper at the maxima and thus harder to optimize. Specifically, we consider

$$\begin{aligned}
f_1 &: x \mapsto 0.5\sin(13x)\sin(27x) + 0.5\\
f_2 &: x \mapsto \max(3.6x(1-x), 1 - |x - 0.05|/0.05)\\
f_3 &: x \mapsto x(1-x)\left(4 - \sqrt{|\sin 60x|}\right)
\end{aligned}$$

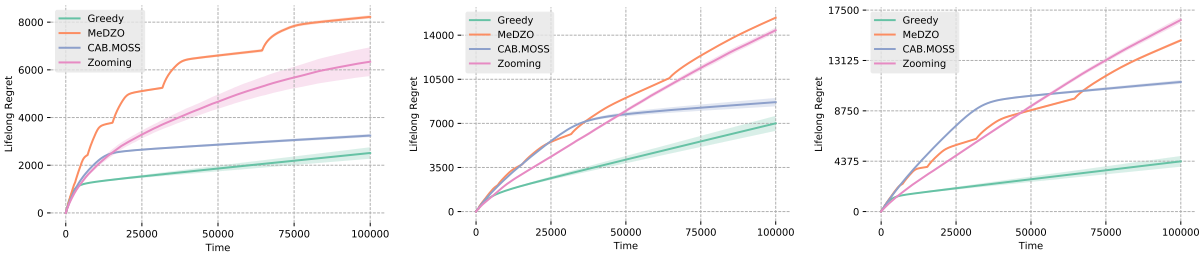
These functions verify Assumption 6.1 with $\alpha = 2, 1, 0.5$ and $L \approx 221, 20, 2$, respectively, and are plotted for convenience in Figure 6.4. Noises are drawn i.i.d. from a standard Gaussian distribution and we consider a time horizon $T = 100000$. We compare the Greedy algorithm with MeDZO [65], CAB1 [85] with MOSS [7] as the underlying algorithm and Zooming [84]. For Greedy, we use the discretization of Corollary 6.2 while for CAB.MOSS we choose the optimal discretization $K = \left\lceil L^{2/(2\alpha+1)}T^{1/(2\alpha+1)}\right\rceil$. For MeDZO, we choose the parameter suggested by authors $B = \sqrt{T}$. We emphasis here that CAB.MOSS and Zooming require the smoothness parameters contrary to MeDZO and Greedy. Results are averaged over 1000 iterations and are presented on Figure 6.5. Shaded area represents 5 standard deviation for each algorithm.

We see that Greedy outperforms the other algorithms in all scenarios. We can clearly observe that the slope of the cumulative regret of Greedy is stepper than the one of CAB.MOSS, yet it manages to obtain a lower regret by quickly concentrating on near-optimal arms. Moreover, the



(a) f_1 (b) f_2 (c) f_3

Figure 6.4: Functions considered in the continuous-armed bandit experiments.



(a) f_1 (b) f_2 (c) f_3

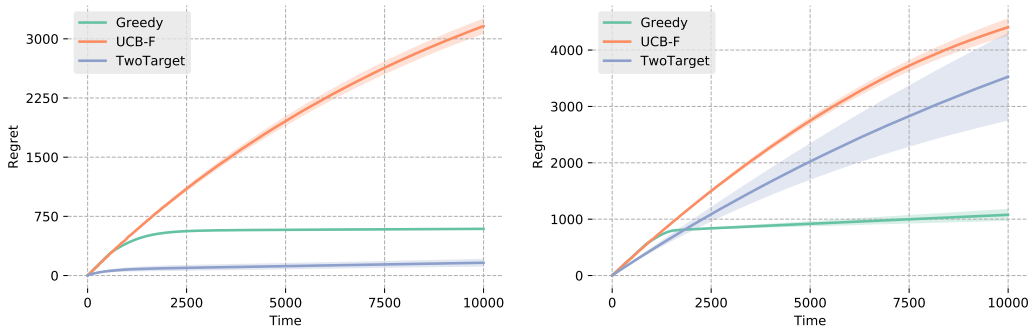
Figure 6.5: Regret of various algorithms as a function of time in continuous-armed bandit problems.

difference is striking for the relatively large time horizon considered here. Interestingly, the slope of Greedy is more pronounced in the second scenario; this may be due to the low number of local maxima which negatively affects the number of ε-optimal arms for Greedy.

6.8.2 Infinite-armed bandits

In the infinite-armed bandit setting, we repeat the experiments of Bonald and Proutiere [23]. We consider two Bernoulli bandit problems with a time horizon $T = 10000$. In the first scenario, mean rewards are drawn i.i.d. from the uniform distribution over $[0, 1]$, while in the second scenario, they are drawn from a Beta(1, 2) distribution. We assume the knowledge of the parameters. We compare Greedy with UCB-F [148] and TwoTarget [23] that further assumes Bernoulli rewards and the knowledge of the underlying distribution of mean rewards. For Greedy, we use the subsampling suggested in Theorem 6.3. Results, averaged over 1000 iterations, are displayed on Figure 6.6 and the shaded area represents 0.5 standard deviation for each algorithm.

Once again, we see the excellent empirical performances of Greedy. It is actually outperformed by TwoTarget in the uniform case since the latter has been specifically optimize for that case (and is asymptotically optimal) but Greedy is more robust as the second scenario points out; furthermore, TwoTarget works only for Bernoulli rewards contrary to Greedy.



(a) Uniform prior

(b) Beta(1, 2) prior

Figure 6.6: Regret of various algorithms as a function of time in infinite-armed bandit problems.

6.8.3 Many-armed bandits

In the many-armed bandit setting, we repeat the experiment of Zhu and Nowak [154]. We consider a Bernoulli bandit problem where best arms have an mean reward of 0.9 while for suboptimal arms they are evenly distributed among $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The time horizon is $T = 5000$ and the total number of arms $n = 2000$. We set the hardness level at $\gamma = 0.4$ resulting in a number of best arms $m = \left\lceil \frac{n}{T^\gamma} \right\rceil = 64$. In this setup, Greedy is near its limit in terms of theoretical guarantee. We compare OracleGreedy, the greedy algorithm run on an subsampling of arms analyzed previously, with MOSS [7], OracleMOSS [154] (which consider an optimal subsampling for MOSS) and the standard Greedy algorithm that consider all arms. For OracleGreedy, we consider a subsampling of $K = (1 - 2\gamma)T^{2\gamma} \log T/4$ arms, which corresponds to the value of a more careful analysis of the regret in the bad events in Theorem 6.4 for 1/4-subgaussian random variables. Results are averaged over 5000 iterations and displayed on Figure 6.7. Shaded area represents 0.5 standard deviation for each algorithm.

Once again we observe the excellent performance of Greedy on a subsampling of arms; it outperforms OracleMOSS, its closest competitor, since both assume the knowledge of the hardness parameter γ and subsample. It is also interesting to notice that the variance of OracleGreedy is much smaller than OracleMOSS.

6.8.4 Linear bandits

In the linear bandit model, for each round t, the learner is given the decision set $\mathcal{A}_t \subset \mathbb{R}^d$, from which she chooses an action $A_t \in \mathcal{A}_t$ and receives reward $X_t = \langle \theta_\star, A_t \rangle + \eta_t$, where $\theta_\star \in \mathbb{R}^d$ is an unknown parameter vector and η_t is some i.i.d. white noise, usually assume 1-subgaussian. In this model, the Greedy algorithm consists of two phases: firstly, it computes the regularized least-square estimator of θ; then, it plays the arm in the action set that maximizes the linear product with the estimator of θ.

Here we consider a problem with a large dimension relatively to the time horizon. Precisely, we fix $d = 50$, a time horizon $T = 2500$ and the noise is a standard Gaussian distribution. The set of arms consists of the unit ball and the parameter θ is randomly generated on the unit sphere. We compare Greedy with LinUCB [1] and BallExplore [52], an algorithm specifically designed

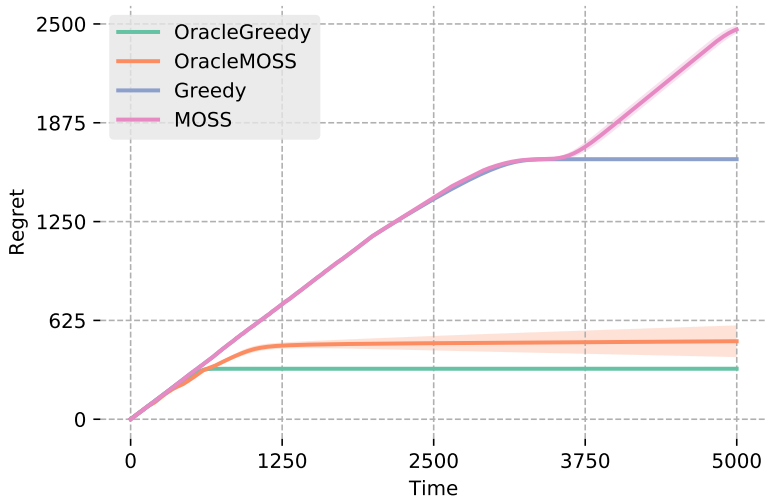


Figure 6.7: Regret of various algorithms on a many-armed bandit problem with hardness level $\alpha = 0.4$.

for such a setting. The regularization term λ is set at 1 for Greedy and LinUCB, the confidence term $\delta = \frac{1}{T}$ for LinUCB and the parameter $\Delta = d$ for BallExplore. Results, displayed on Figure 6.8, are averaged over 50 iterations. Shaded area represents 2 times the standard deviation for each algorithm.

We see that Greedy outperforms both LinUCB and BallExplore; in particular the regret of Greedy is sublinear. Another point that we have not emphasized so far is the computational complexity. Until now, the difference in terms of computation was rather insignificant. This is no longer the case for algorithms designed for linear bandits as they must solve an optimization problem at each time step. For example, in this simulation, the number of seconds per iteration on a single-core processor is 70 for Greedy, 678 for LinUCB and 1031 for BallExplore. In words, Greedy is nearly ten times faster than LinUCB.

6.8.5 Cascading bandits

We now bring our attention into a special case of stochastic combinatorial optimization under semi-bandit feedback that is the cascading bandit problem. Formally, we have $L \in \mathbb{N}$ ground items and at each time step t, the agent recommends a list $\left(a_1^t, \dots, a_K^t\right)$ of $K \leq L$ items to the user. The user examines the list, from the first item to the last, and clicks on the first attractive item, if any. With each item $l \in [L]$ is associated a weight $w(l) \in [0, 1]$, which denotes the click probability of the item. The reward of the agent at time t is given by $1 - \prod_{K=1}^{K} \left(1 - w(a_k^t)\right) \in \{0, 1\}$ and she receives feedback for each $k \in [K]$ such that $k \leq c_t = \min\left\{1 \leq k \leq K : w_t(a_k^t) = 1\right\}$ where $w_t(a_k^t) \sim \text{Bernoulli}(w(a_k^t))$ and we assume that the minimum over an empty set is ∞. In this setting, the Greedy algorithm outputs a list consisting of the K best empirical arms. The goal of these experiments is to study in which regimes, as a function of L and K, the Greedy algorithm might be preferable to the state-of-the-art.

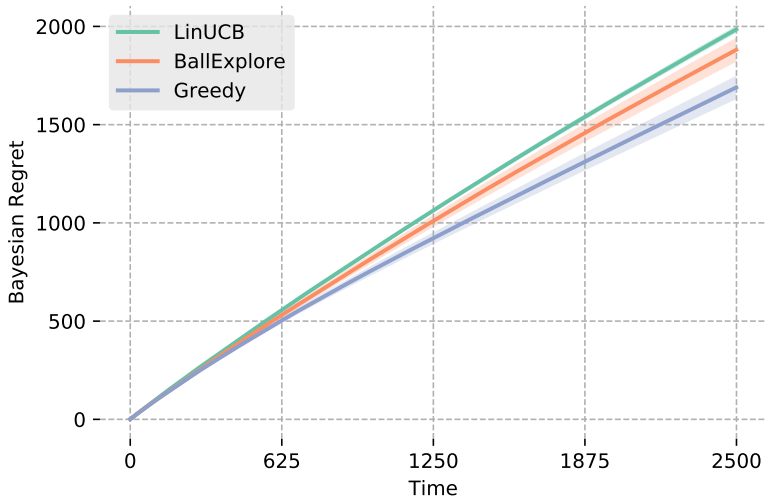


Figure 6.8: Bayesian regret of various algorithms as a function of time in a linear bandit problem.

We reproduce the experiments of Kveton, Szepesvari, Wen, and Ashkan [90] in the Bayesian setting. We compare Greedy with CascadeKL-UCB [90] and TS-Cascade [38]. Cascade-Greedy and CascadeKL-UCB share the same initialization which is to select each item once as the first item on the list. For each algorithm, the list is ordered from the largest index to the smallest one. We consider two scenarios: on the first one, the prior on the expected rewards is a uniform distribution while on the second scenario, we consider a more realistic Beta(1, 3) distribution so that most arms have low expected rewards. The time horizon is set at $T = 10000$. The regret and standard deviation of each algorithm, averaged over 100 iterations, are reported in Table 6.1 and 6.2 for different values of L and K.

As expected by the Bayesian setting, Greedy outplays the state-of-the-art when the number of arms L is large. Even more interesting is that, as the number of recommended items K gets larger the regret of Greedy decreases at a faster rate than the other algorithms. Our intuition is that the conservatism of standard bandit algorithms is amplified as K increases and this is further exacerbates by the cascade model where items at the bottom of the list may not get a feedback. On the contrary, the Greedy algorithm quickly converges to a solution that uniquely depends on past individual performances of arms. In addition, the contrast between the performance of Greedy and the state-of-the-art is even more striking in the second scenario. This is not particularly surprising as the Beta(1, 3) distribution gives rise to harder problems for the considered time horizon.

6.8.6 Mortal bandits

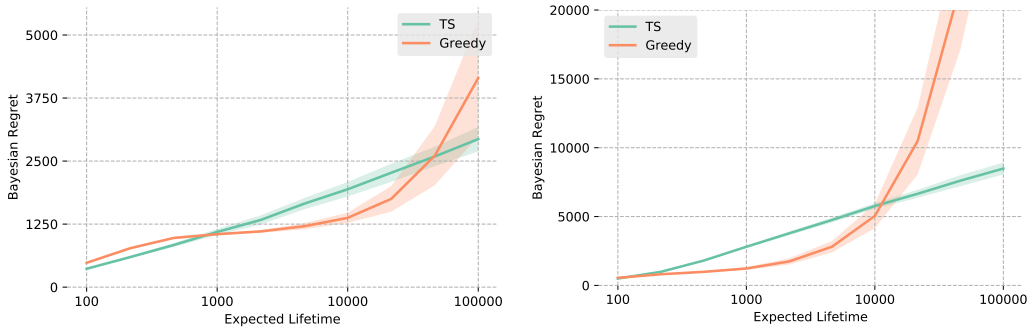
We now consider the mortal bandit problem where arms die and new ones appear regularly (in particular, an arm is not always available contrary to the standard model). In this setting, the Greedy algorithm pulls the best empirical arm available. As previous work considered a large number of arms, state-of-the-art algorithms in this setting, e.g. AdaptiveGreedy [32], emphasis an hidden

Table 6.1: Bayesian regret of various algorithms in cascading bandit problems with a uniform prior.

L	K	Greedy	CascadeKL-UCB	TS-Cascade
16	2	176.1 ± 26.4	**48.1 ± 2.7**	109.7 ± 1.8
16	4	10.2 ± 1.9	**9.9 ± 1.0**	28.4 ± 0.9
16	8	**0.7 ± 0.2**	**0.7 ± 0.1**	3.6 ± 0.3
32	2	166.1 ± 22.8	**58.7 ± 3.5**	178.7 ± 2.5
32	4	**6.7 ± 0.9**	10.1 ± 0.8	47.0 ± 1.0
32	8	**0.2 ± 0.03**	0.7 ± 0.08	8.3 ± 0.4
64	2	135.5 ± 15.6	**76.6 ± 3.7**	288.6 ± 2.6
64	4	**6.5 ± 0.5**	12.5 ± 0.6	80.3 ± 1.3
64	8	**0.3 ± 0.02**	0.9 ± 0.07	16.6 ± 0.5
128	2	133.1 ± 12.4	**107.4 ± 4.8**	442.6 ± 3.4
128	4	**9.4 ± 0.3**	18.0 ± 0.8	127.4 ± 1.5
128	8	**0.5 ± 0.02**	1.5 ± 0.1	27.9 ± 0.6
256	2	**137.2 ± 10.6**	151.0 ± 5.6	605.7 ± 3.1
256	4	**16.6 ± 0.2**	26.9 ± 1.0	179.5 ± 1.4
256	8	**1.0 ± 0.03**	1.8 ± 0.1	39.9 ± 0.5

Table 6.2: Bayesian regret of various algorithms in cascading bandit problems with a Beta(1, 3) prior.

L	K	Greedy	CascadeKL-UCB	TS-Cascade
16	2	590.4 ± 83.5	207.9 ± 5.2	**199.5 ± 3.6**
16	4	304.8 ± 35.7	116.4 ± 4.2	**103.2 ± 2.9**
16	8	97.9 ± 11.7	39.6 ± 2.1	**34.4 ± 1.6**
32	2	433.1 ± 49.1	**330.7 ± 8.3**	333.7 ± 3.8
32	4	192.2 ± 23.1	166.2 ± 6.0	**163.3 ± 3.7**
32	8	**38.7 ± 5.3**	50.1 ± 2.9	54.6 ± 1.9
64	2	576.2 ± 55.8	**485.8 ± 11.2**	540.1 ± 4.8
64	4	**144.2 ± 12.3**	207.5 ± 6.8	246.1 ± 4.1
64	8	**20.3 ± 1.8**	49.2 ± 2.2	76.4 ± 1.6
128	2	**575.2 ± 40.1**	710.9 ± 16.3	843.4 ± 4.7
128	4	**100.8 ± 5.5**	270.6 ± 7.4	372.9 ± 3.7
128	8	**18.0 ± 0.6**	60.7 ± 2.0	115.7 ± 1.4
256	2	**522.5 ± 32.4**	1068.3 ± 26.1	1235.1 ± 6.3
256	4	**125.1 ± 3.8**	380.0 ± 10.3	551.1 ± 3.85
256	8	**27.3 ± 0.4**	86.4 ± 2.6	174.8 ± 1.5



(a) Uniform prior

(b) Beta(1, 3) prior

Figure 6.9: Bayesian regret of various algorithms as a function of the expected lifetime of arms in mortal bandit problems.

subsampling of arms due to their initialization. They further required a careful (manual) tuning of their parameter for optimal performance. Consequently, we compare Greedy to a standard bandit algorithm extended to this model and we consider a small number of arms. Similarly to the last setting, the goal is to observe in which regimes, as a function of the expected lifetime of arms, Greedy might be preferable.

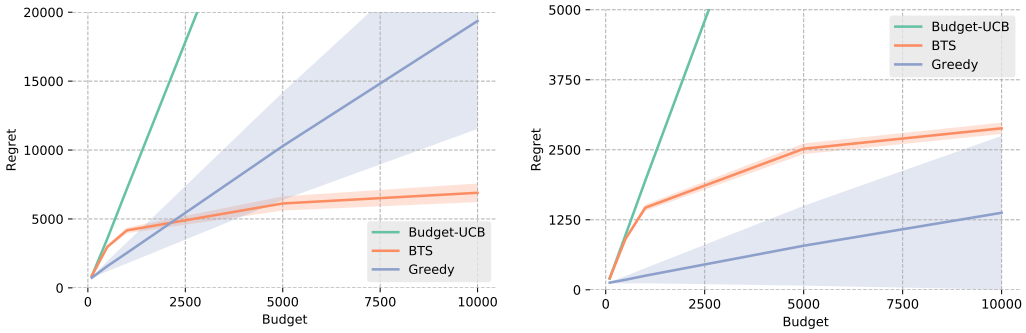
We repeat the experiments of Chakrabarti, Kumar, Radlinski, and Upfal [32] with $K = 100$. The number of arms remains fixed throughout the time horizon T, that is when an arm dies, it is immediately replaced by another one. The time horizon T is set at 10 times the expected lifetime of the arms. The lifetime of arm k, denoted L_k, is drawn i.i.d. from a geometric distribution with expected lifetime L; this arm dies after being available for L_k rounds. We consider logarithmically spaced values of expected lifetimes. We also assume that arms are Bernoulli random variables. We consider two scenarios: in the first one, expected rewards of arms are drawn i.i.d. from a uniform distribution over $[0, 1]$, while in the second scenario they are drawn from a Beta(1, 3) distribution. We compare the Greedy algorithm with Thompson Sampling [5]. Results are averaged over 100 iterations and are reported on Figure 6.9. Shaded area represents 0.5 standard deviation for each algorithm.

As expected, Greedy outperforms TS for intermediate expected lifetime and vice versa for long lifetime. And for short lifetime, as we previously saw, a subsampling of arms could have considerably improve the performance of both algorithms.

6.8.7 Budgeted bandits

We now consider the budgeted bandit problem. In this model, the pull of arm k at time t entails a random cost $c_k(t)$. Moreover, the learner has a budget B, which is a known parameter, that will constrain the total number of pulls. In this setting, the index of an arm in the Greedy algorithm is the average reward divided by the average cost. Like before, the objective is to evaluate in which regimes with respect to the budget B, Greedy might be preferable to a state-of-the-art algorithm.

We reproduce the experiments of Xia, Ding, Zhang, Yu, and Qin [151]. Specifically, we study two scenarios with $K = 100$ arms in each. The first scenario considers discrete costs; both the reward



(a) Discrete costs

(b) Continuous costs

Figure 6.10: Regret of various algorithms as a function of the budget in budgeted bandit problems.

and the cost are sampled from Bernoulli distributions with parameters randomly sampled from $(0, 1)$. The second scenario considers continuous costs; the reward and cost of an arm is sampled from two different Beta distributions, the two parameters of each distribution are uniformly sampled from $[1, 5]$. The budget is chosen from the set $\{100, 500, 1000, 5000, 10000\}$. We compare Greedy to Budget-UCB Xia, Ding, Zhang, Yu, and Qin [151] and BTS [152]. The results of simulations are displayed in Figure 6.10 and are averaged over 500 runs. Shaded area represents 0.5 standard deviation for each algorithm.

Interestingly, in this setting the interval of budgets for which Greedy outperforms baseline algorithms is extremely small for discrete costs and large for continuous costs. In the latter case, even for large budget Greedy has a lower expected regret than BTS. Nonetheless it suffers from a huge variance which makes its use risky in practice.

6.9 Conclusion

In this chapter, we have refined the standard version of Greedy by considering a subsampling of arms and proved sublinear worst-case regret bounds in several bandit models. We also carried out an extensive experimental evaluation which reveals that it outperforms the state-of-the-art for relatively short time horizon. Besides, since its indexes are usually computed by most algorithms, it is trivial to implement and fast to run. Consequently, the Greedy algorithm should be considered as a standard baseline when multiple near-optimal arms are present, which is the case in many models as we saw.

Interesting directions We leave open the question of adaptivity. Adaptivity here could refer to adaptive subsampling or adaptivity to unknown parameters. In particular in the continuous-armed bandit problem, previous work showed that the learner pays a polynomial cost to adapt [65]. Knowing that Greedy works best for relatively short time horizon, it might be interesting to study this cost for a greedy strategy and for what time horizon it might be worth it.

Another interesting, and relevant in practical problems, direction is to analyze the performance of Greedy in combinatorial bandits (with a large number of arms and thus a non-tractable number of *actions*), but with some structure on the rewards on arms [50, 90, 122, 124].

6.10 Short literature review

In this section, we provide a short literature review on the different bandit models considered in this chapter.

Continuous-armed bandits Agrawal [3] introduced the continuous-armed bandit problem with nonparametric regularity assumptions. Kleinberg [85] established the lower bound and provide a optimal algorithm up to a sublogarithmic factor. Auer, Ortner, and Szepesvári [11] improved the previous bound assuming a margin condition. Kleinberg, Slivkins, and Upfal [84] considered generic metric spaces assuming that the mean-payoff function is Lipschitz with respect to the (known) metric of the space. Bubeck, Munos, Stoltz, and Szepesvari [28] considered generic topological spaces and that the mean-payoff function is locally Lipschitz with respect to a dissimilarity function known to the decision maker. All these works assumed known smoothness parameters; Bubeck, Stoltz, and Yu [30], Hadiji [65], and Locatelli and Carpentier [108] studied the adaptivity to unknown parameters

Infinite-armed bandits Berry, Chen, Zame, Heath, and Shepp [18] introduced the infinite-armed bandit problem, they consider a problem consisting of a sequence of n choices from an infinite number of Bernoulli arms, with $n \to \infty$. The objective is to minimize the long-run failure rate. The Bernoulli parameters are independent observations from a known distribution. Bonald and Proutiere [23] also considered Bernoulli arms but they studied the cumulative regret, focusing on the uniform prior distribution. Wang, Audibert, and Munos [148] considered a more general model. In particular they assumed that rewards are uniformly bounded in $[0, 1]$ and that the expected reward of a randomly drawn arm is ε-optimal with probability $\mathcal{O}\left(\varepsilon^{\beta}\right)$ for some $\beta > 0$.

Many-armed bandits Models in many-armed bandit problems are more varied. Teytaud, Gelly, and Sebag [136] provided an anytime algorithm when the number of arms is large comparatively to the number of time steps. Wang, Kurniawati, and Kroese [146] proposed a cross-entropy based algorithm. They aimed to focus exploration on a small subset of arms. They did not provide theoretical upper bounds. Chaudhuri and Kalyanakrishnan [34] introduced a notion of regret with respect to a given quantile fraction ρ of the probability distribution over the expected rewards of arms. Russo and Van Roy [130] considered learning a satisficing action and analyze the discounted regret. They propose a Thompson Sampling like algorithm and further studied applications to linear and infinite-armed bandits. The definition of a satisficing action is set by the learner. Ou, Li, Yang, Zhu, and Jin [119] proposed a semi-parametric model to formulate expected rewards. Zhu and Nowak [154] considered a setting with multiple best/near-optimal arms without making any assumptions about the structure of the bandit instance. Their objective was to design algorithms that can automatically adapt to the unknown hardness of the problem.

Linear bandits The literature on linear bandits is quite large and we refer the reader to Lattimore and Szepesvári [100] for an in-depth overview. We mention Abbasi-Yadkori, Pál, and Szepesvári [1] who proved that an expected regret of $\widetilde{\mathcal{O}}\left(d\sqrt{T}\right)$ can be achieved as long as the means are guaranteed to lie in a bounded interval. Deshpande and Montanari [52] also considered a linear bandit

problem with a dimension that is large relative to the time horizon. They proposed an algorithm that limits exploration and achieves good reward within a short time frame.

Cascading bandits Kveton, Szepesvari, Wen, and Ashkan [90] introduced the cascading bandit model and proposed two algorithms based on UCB and KL-UCB. Combes, Magureanu, Proutiere, and Laroche [42] proposed an asymptotically optimal algorithm. Cheung, Tan, and Zhong [38] proposed an algorithm based on Thompson Sampling. Zong, Ni, Sung, Ke, Wen, and Kveton [156] considered a linear variant of the model. Li, Wang, Zhang, and Chen [107] further considered a contextual setting.

Mortal bandits Chakrabarti, Kumar, Radlinski, and Upfal [32] introduced the mortal bandit model and proposed an algorithm in which the level of greediness depends on the performance of the best arm available. Traca, Rudin, and Yan [140] argued to reduce exploration of dying arms.

Budgeted bandits Tran-Thanh, Chapman, Munoz De Cote Flores Luna, Rogers, and Jennings [141] introduced the budgeted bandit problem and proposed an Explore-Then-Commit algorithm. Tran-Thanh, Chapman, Rogers, and Jennings [142] proposed knapsack-based algorithms. Xia, Li, Qin, Yu, and Liu [152] proposed a Thomson sampling algorithm. Xia, Ding, Zhang, Yu, and Qin [151] further extended the problem to continuous random costs Let us also cite Badanidiyuru, Kleinberg, and Slivkins [12] who considered a more general framework.

6.11 Useful lemma

Lemma 6.1. *Let a and b be two real numbers. Then the following holds true*

$$\lfloor a+b \rfloor - \lceil a-b \rceil \geq \lfloor 2b \rfloor - 1$$

Proof. We have

$$\begin{aligned} \lfloor a+b \rfloor - \lceil a-b \rceil &= \lfloor a+b \rfloor + \lfloor b-a \rfloor \\ &\geq \lfloor a+b+b-a \rfloor - 1 \\ &= \lfloor 2b \rfloor - 1 \end{aligned}$$

where we used respectively that, $\lceil x \rceil = -\lfloor -x \rfloor$ and $\lfloor x+y \rfloor \leq \lfloor x \rfloor + \lfloor y \rfloor + 1$. □

A An improved bound on the regret of Follow-The-Leader in full information

In this section, we provide a tighter upper bound, compared to Degenne and Perchet [48], on the regret of the Follow-The-Leader algorithm under the full information feedback. We recall that in the full information setting, the learner observes a reward for each arm independently of her previous choices. We also recall that after a uniformly random pull in the first time step, the Follow-The-Leader algorithm pulls the arm with the highest average reward.

As usual in the literature, we consider σ^2-subgaussian reward distributions and a unique optimal arm which is arm 1 without loss of generality. We further assume that the suboptimal gap is the same for all arms and we denote it Δ. We now state the theorem.

Theorem A.1. *The expected regret of FTL in the full information setting with equal suboptimal gaps verifies for all* $t \in \mathbb{N}$

$$\mathbb{E}[R_t] \leq \frac{4\sigma^2}{\Delta}(1 + \log(K-1)) + \Delta\frac{K-1}{K}\,.$$

Proof. The main improvement comes from comparing the optimal arm and all suboptimal arms rather than two by two. The probability of pulling a suboptimal arm at time $t+1$ is bounded by

$$\begin{aligned}
\mathbb{P}(A_{t+1} \neq 1) &\leq \mathbb{P}\left(\max_{k\neq 1} \widehat{\mu}_k(t) \geq \widehat{\mu}_1(t)\right) \\
&\leq 1 - \mathbb{P}\left(\max_{k\neq 1} \widehat{\mu}_k(t) < \widehat{\mu}_1(t)\right) \\
&\leq 1 - \prod_{k\neq 1} \mathbb{P}(\widehat{\mu}_k(t) < \widehat{\mu}_1(t)) \\
&\leq 1 - \prod_{k\neq 1} (1 - \mathbb{P}(\widehat{\mu}_k(t) \geq \widehat{\mu}_1(t)))
\end{aligned}$$

Using that $\widehat{\mu}_k(t)$ is $1/t$-subgaussian for all k, we have

$$\begin{aligned}
\mathbb{P}(\widehat{\mu}_k(t) \geq \widehat{\mu}_1(t)) &= \mathbb{P}(\widehat{\mu}_k(t) - \widehat{\mu}_1(t) - (-\Delta) \geq \Delta) \\
&\leq \exp\left(-\frac{t\Delta^2}{4\sigma^2}\right)
\end{aligned}$$

Hence, we get

$$\begin{aligned}\mathbb{P}(A_{t+1} \neq 1) &\leq 1 - \prod_{k \neq 1} \left(1 - \exp\left(-\frac{t\Delta^2}{4\sigma^2}\right)\right) \\ &\leq 1 - \left(1 - \exp\left(-\frac{t\Delta^2}{4\sigma^2}\right)\right)^{K-1}\end{aligned}$$

Finally, by the standard decomposition of the regret we obtain

$$\begin{aligned}\mathbb{E}[R_t] &= \Delta \sum_{t=1}^{T-1} \mathbb{P}(A_{t+1} \neq 1) + \Delta \frac{K-1}{K} \\ &\leq \Delta \sum_{t=1}^{T-1} 1 - \left(1 - \exp\left(-\frac{t\Delta^2}{4\sigma^2}\right)\right)^{K-1} + \Delta \frac{K-1}{K} \\ &\leq \Delta \int_0^{\infty} 1 - \left(1 - \exp\left(-\frac{x\Delta^2}{4\sigma^2}\right)\right)^{K-1} dx + \Delta \frac{K-1}{K} \\ &\leq \frac{4\sigma^2}{\Delta} \int_0^1 \frac{1-u^{K-1}}{1-u} du + \Delta \frac{K-1}{K} \\ &\leq \frac{4\sigma^2}{\Delta} \int_0^1 \sum_{k=0}^{K-2} u^k du + \Delta \frac{K-1}{K} \\ &\leq \frac{4\sigma^2}{\Delta} \sum_{k=1}^{K-1} \frac{1}{k} + \Delta \frac{K-1}{K} \\ &\leq \frac{4\sigma^2}{\Delta}(1 + \log(K-1)) + \Delta \frac{K-1}{K}\end{aligned}$$

where in the second inequality we used that the function inside the sum term is positive and decreasing. □

Bibliography

1. Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. "Improved algorithms for linear stochastic bandits". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2312–2320.
2. N. Abe and P. M. Long. "Associative reinforcement learning using linear probabilistic concepts". In: *ICML*. 1999, pp. 3–11.
3. R. Agrawal. "The continuum-armed bandit problem". *SIAM journal on control and optimization* 33:6, 1995, pp. 1926–1951.
4. S. Agrawal, N. R. Devanur, and L. Li. "An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives". In: *Conference on Learning Theory*. 2016, pp. 4–18.
5. S. Agrawal and N. Goyal. "Analysis of thompson sampling for the multi-armed bandit problem". In: *Conference on learning theory*. 2012, pp. 39–1.
6. S. Agrawal and N. Goyal. "Further optimal regret bounds for thompson sampling". In: *Artificial intelligence and statistics*. 2013, pp. 99–107.
7. J.-Y. Audibert and S. Bubeck. "Minimax Policies for Adversarial and Stochastic Bandits". In: *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*. Best Student Paper Award. 2009. URL: https://www.microsoft.com/en-us/research/publication/minimax-policies-adversarial-stochastic-bandits/.
8. J.-Y. Audibert, R. Munos, and C. Szepesvári. "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits". *Theoretical Computer Science* 410:19, 2009, pp. 1876–1902.
9. P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multiarmed bandit problem". *Machine learning* 47:2-3, 2002, pp. 235–256.
10. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The nonstochastic multiarmed bandit problem". *SIAM journal on computing* 32:1, 2002, pp. 48–77.
11. P. Auer, R. Ortner, and C. Szepesvári. "Improved rates for the stochastic continuum-armed bandit problem". In: *International Conference on Computational Learning Theory*. Springer. 2007, pp. 454–468.
12. A. Badanidiyuru, R. Kleinberg, and A. Slivkins. "Bandits with knapsacks". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE. 2013, pp. 207–216.
13. A. Badanidiyuru, R. Kleinberg, and A. Slivkins. "Bandits with knapsacks". *Journal of the ACM (JACM)* 65:3, 2018, p. 13.

14. A. Baransi, O.-A. Maillard, and S. Mannor. "Sub-sampling for multi-armed bandits". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2014, pp. 115–131.
15. H. Bastani, M. Bayati, and K. Khosravi. "Mostly exploration-free algorithms for contextual bandits". *arXiv preprint arXiv:1704.09011*, 2017.
16. V. S. Bawa. "Optimal rules for ordering uncertain prospects". *Journal of Financial Economics* 2:1, 1975, pp. 95–121.
17. M. Bayati, N. Hamidi, R. Johari, and K. Khosravi. "Optimal and Greedy Algorithms for Multi-Armed Bandits with Many Arms". *arXiv preprint arXiv:2002.10121*, 2020.
18. D. A. Berry, R. W. Chen, A. Zame, D. C. Heath, and L. A. Shepp. "Bandit problems with infinitely many arms". *The Annals of Statistics*, 1997, pp. 2103–2116.
19. D. A. Berry and B. Fristedt. "Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)". *London: Chapman and Hall* 5, 1985, pp. 71–87.
20. L. Besson and E. Kaufmann. "What doubling tricks can and can't do for multi-armed bandits". *arXiv preprint arXiv:1803.06971*, 2018.
21. A. Bietti, A. Agarwal, and J. Langford. "A contextual bandit bake-off". *arXiv preprint arXiv:1802.04064*, 2018.
22. Z. Bnaya, R. Puzis, R. Stern, and A. Felner. "Volatile multi-armed bandits for guaranteed targeted social crawling". In: *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*. 2013.
23. T. Bonald and A. Proutiere. "Two-target algorithms for infinite-armed bandits with Bernoulli rewards". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2184–2192.
24. D. Bouneffouf and I. Rish. "A survey on practical applications of multi-armed and contextual bandits". *arXiv preprint arXiv:1904.10040*, 2019.
25. C. Boutilier, C.-W. Hsu, B. Kveton, M. Mladenov, C. Szepesvari, and M. Zaheer. "Differentiable Bandit Exploration". *arXiv preprint arXiv:2002.06772*, 2020.
26. G. Bresler, G. H. Chen, and D. Shah. "A latent source model for online collaborative filtering". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3347–3355.
27. S. Bubeck and N. Cesa-Bianchi. "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". *arXiv preprint arXiv:1204.5721*, 2012.
28. S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. "X-armed bandits". *arXiv preprint arXiv:1001.4475*, 2010.
29. S. Bubeck, V. Perchet, and P. Rigollet. "Bounded regret in stochastic multi-armed bandits". In: *Conference on Learning Theory*. 2013, pp. 122–134.
30. S. Bubeck, G. Stoltz, and J. Y. Yu. "Lipschitz bandits without the Lipschitz constant". In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 144–158.
31. A. Carpentier and M. Valko. "Simple regret for infinitely many armed bandits". In: *International Conference on Machine Learning*. 2015, pp. 1133–1141.

32. D. Chakrabarti, R. Kumar, F. Radlinski, and E. Upfal. "Mortal multi-armed bandits". In: *Advances in neural information processing systems*. 2009, pp. 273–280.

33. O. Chapelle and L. Li. "An empirical evaluation of thompson sampling". In: *Advances in neural information processing systems*. 2011, pp. 2249–2257.

34. A. R. Chaudhuri and S. Kalyanakrishnan. "Quantile-Regret Minimisation in Infinitely Many-Armed Bandits." In: *UAI*. 2018, pp. 425–434.

35. N. Chen, C. Wang, and L. Wang. "Learning and Optimization with Seasonal Patterns". *arXiv preprint arXiv:2005.08088*, 2020.

36. Z. Chen and B. Liu. "Lifelong machine learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning* 10:3, 2016, pp. 1–145.

37. W. C. Cheung, D. Simchi-Levi, and R. Zhu. "Hedging the drift: Learning to optimize under non-stationarity". *arXiv preprint arXiv:1903.01461*, 2019.

38. W. C. Cheung, V. Tan, and Z. Zhong. "A thompson sampling algorithm for cascading bandits". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 438–447.

39. R. Cohen, L. Katzir, and D. Raz. "An efficient approximation for the generalized assignment problem". *Information Processing Letters* 100:4, 2006, pp. 162–166.

40. R. Combes, C. Jiang, and R. Srikant. "Bandits with budgets: Regret lower bounds and optimal algorithms". *ACM SIGMETRICS Performance Evaluation Review* 43:1, 2015, pp. 245–257.

41. R. Combes, S. Magureanu, and A. Proutiere. "Minimal exploration in structured stochastic bandits". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1761–1769.

42. R. Combes, S. Magureanu, A. Proutiere, and C. Laroche. "Learning to rank: Regret lower bounds and efficient algorithms". In: *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. 2015, pp. 231–244.

43. P.-A. Coquelin and R. Munos. "Bandit algorithms for tree search". *arXiv preprint cs/0703062*, 2007.

44. G. Dahl and N. Foldnes. "LP based heuristics for the multiple knapsack problem with assignment restrictions". *Annals of Operations Research* 146:1, 2006, pp. 91–104.

45. V. Dani, T. P. Hayes, and S. M. Kakade. "Stochastic Linear Optimization under Bandit Feedback". In: *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*. 2008, pp. 355–366. URL: http://colt2008.cs.helsinki.fi/papers/80-Dani.pdf.

46. H. A. David and H. N. Nagaraja. *Order statistics*. third. Wiley, 2003.

47. M. Dawande, J. Kalagnanam, P. Keskinocak, F. S. Salman, and R. Ravi. "Approximation algorithms for the multiple knapsack problem with assignment restrictions". *Journal of combinatorial optimization* 4:2, 2000, pp. 171–186.

48. R. Degenne and V. Perchet. "Anytime optimal algorithms in stochastic multi-armed bandits". In: *International Conference on Machine Learning*. 2016, pp. 1587–1595.

49. R. Degenne and V. Perchet. "Combinatorial semi-bandit with known covariance". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2972–2980.

50. R. Degenne and V. Perchet. "Combinatorial semi-bandit with known covariance". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2972–2980.

51. A. A. Deshmukh, U. Dogan, and C. Scott. "Multi-task learning for contextual bandits". In: *Advances in neural information processing systems*. 2017, pp. 4848–4856.

52. Y. Deshpande and A. Montanari. "Linear bandits in high dimension and recommendation systems". In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2012, pp. 1750–1754.

53. G. Di Benedetto, V. Bellini, and G. Zappella. "A Linear Bandit for Seasonal Environments". *arXiv preprint arXiv:2004.13576*, 2020.

54. E. Even-Dar, S. Mannor, and Y. Mansour. "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems". *Journal of machine learning research* 7:Jun, 2006, pp. 1079–1105.

55. E. Even-Dar, S. Mannor, and Y. Mansour. "PAC bounds for multi-armed bandit and Markov decision processes". In: *International Conference on Computational Learning Theory*. Springer. 2002, pp. 255–270.

56. A. Garivier and O. Cappé. "The KL-UCB algorithm for bounded stochastic bandits and beyond". In: *Proceedings of the 24th annual Conference On Learning Theory*. 2011, pp. 359–376.

57. A. Garivier, T. Lattimore, and E. Kaufmann. "On explore-then-commit strategies". In: *Advances in Neural Information Processing Systems*. 2016, pp. 784–792.

58. A. Garivier, P. Ménard, and G. Stoltz. "Explore first, exploit next: The true shape of regret in bandit problems". *Mathematics of Operations Research* 44:2, 2019, pp. 377–399.

59. A. Garivier and E. Moulines. "On upper-confidence bound policies for switching bandit problems". In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 174–188.

60. C. Gentile, S. Li, and G. Zappella. "Online clustering of bandits". In: *International Conference on Machine Learning*. 2014, pp. 757–765.

61. J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

62. J. C. Gittins. "Bandit processes and dynamic allocation indices". *Journal of the Royal Statistical Society: Series B (Methodological)* 41:2, 1979, pp. 148–164.

63. T. L. Graves and T. L. Lai. "Asymptotically efficient adaptive choice of control laws incontrolled markov chains". *SIAM journal on control and optimization* 35:3, 1997, pp. 715–743.

64. J. Hadar and W. R. Russell. "Rules for ordering uncertain prospects". *The American economic review* 59:1, 1969, pp. 25–34.

65. H. Hadiji. "Polynomial Cost of Adaptation for X-Armed Bandits". In: *Advances in Neural Information Processing Systems*. 2019, pp. 1029–1038.

66. C. Hartland, S. Gelly, N. Baskiotis, O. Teytaud, and M. Sebag. "Multi-armed Bandit, Dynamic Environments and Meta-Bandits". working paper or preprint. 2006. URL: https://hal.archives-ouvertes.fr/hal-00113668.

67. J. Honda and A. Takemura. "An Asymptotically Optimal Bandit Algorithm for Bounded Support Models." In: *COLT*. Citeseer. 2010, pp. 67–79.

68. J. Honda and A. Takemura. "Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards". *The Journal of Machine Learning Research* 16:1, 2015, pp. 3721–3756.

69. C.-W. Hsu, B. Kveton, O. Meshi, M. Mladenov, and C. Szepesvari. "Empirical bayes regret minimization". *arXiv preprint arXiv:1904.02664*, 2019.

70. N. Immorlica, K. A. Sankararaman, R. Schapire, and A. Slivkins. "Adversarial bandits with knapsacks". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2019, pp. 202–219.

71. M. Jedor, J. Louëdec, and V. Perchet. "Lifelong Learning in Multi-Armed Bandits". *arXiv preprint arXiv:2012.14264*, 2020.

72. C. Jiang and R. Srikant. "Bandits with budgets". In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 5345–5350. DOI: 10.1109/CDC.2013.6760730.

73. S. Kannan, J. H. Morgenstern, A. Roth, B. Waggoner, and Z. S. Wu. "A smoothed analysis of the greedy algorithm for the linear contextual bandit problem". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2227–2236.

74. S. Katariya, L. Jain, N. Sengupta, J. Evans, and R. Nowak. "Adaptive sampling for coarse ranking". *arXiv preprint arXiv:1802.07176*, 2018.

75. S. Katariya, B. Kveton, C. Szepesvari, C. Vernade, and Z. Wen. "Stochastic rank-1 bandits". *arXiv preprint arXiv:1608.03023*, 2016.

76. S. Katariya, B. Kveton, C. Szepesvari, C. Vernade, and Z. Wen. "Stochastic rank-1 bandits". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 392–401.

77. S. Katariya, B. Kveton, C. Szepesvari, and Z. Wen. "DCM bandits: Learning to rank with multiple clicks". In: *International Conference on Machine Learning*. 2016, pp. 1215–1224.

78. E. Kaufmann. "On Bayesian index policies for sequential resource allocation". *arXiv preprint arXiv:1601.01190*, 2016.

79. E. Kaufmann, O. Cappé, and A. Garivier. "On Bayesian upper confidence bounds for bandit problems". In: *Artificial intelligence and statistics*. 2012, pp. 592–600.

80. E. Kaufmann, W. Koolen, and A. Garivier. "Sequential Test for the Lowest Mean: From Thompson to Murphy Sampling". *arXiv preprint arXiv:1806.00973*, 2018.

81. E. Kaufmann, N. Korda, and R. Munos. "Thompson sampling: An asymptotically optimal finite-time analysis". In: *International conference on algorithmic learning theory*. Springer. 2012, pp. 199–213.

82. J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. "Efficient Thompson Sampling for Online ï¿¼Matrix-Factorization Recommendation". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 1297–1305. URL: http://papers.nips.cc/paper/5985-efficient-thompson-sampling-for-online-matrix-factorization-recommendation.pdf.

83. J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. "Efficient Thompson Sampling for Online Matrix-Factorization Recommendation". In: *Advances in neural information processing systems*. 2015, pp. 1297–1305.

84. R. Kleinberg, A. Slivkins, and E. Upfal. "Multi-armed bandits in metric spaces". In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 681–690.

85. R. D. Kleinberg. "Nearly tight bounds for the continuum-armed bandit problem". In: *Advances in Neural Information Processing Systems*. 2005, pp. 697–704.

86. T. Kocák, M. Valko, R. Munos, and S. Agrawal. "Spectral thompson sampling". In: *Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014.

87. L. Kocsis and C. Szepesvári. "Bandit based monte-carlo planning". In: *European conference on machine learning*. Springer. 2006, pp. 282–293.

88. N. Korda, B. Szörényi, and L. Shuai. "Distributed clustering of linear bandits in peer to peer networks". In: *Journal of machine learning research workshop and conference proceedings*. Vol. 48. International Machine Learning Societ. 2016, pp. 1301–1309.

89. V. Kuleshov and D. Precup. "Algorithms for multi-armed bandit problems". *arXiv preprint arXiv:1402.6028*, 2014.

90. B. Kveton, C. Szepesvari, Z. Wen, and A. Ashkan. "Cascading bandits: Learning to rank in the cascade model". In: *International Conference on Machine Learning*. 2015, pp. 767–776.

91. J. Kwon and V. Perchet. "Gains and losses are fundamentally different in regret minimization: The sparse case". *The Journal of Machine Learning Research* 17:1, 2016, pp. 8106–8137.

92. J. Kwon, V. Perchet, and C. Vernade. "Sparse stochastic bandits". In: *30th Annual Conference on Learning Theory - COLT 2017, Amsterdam, Netherlands, July 7-10, 2017*. 2017, pp. 355–366.

93. P. Lagrée, C. Vernade, and O. Cappe. "Multiple-play bandits in the position-based model". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1597–1605.

94. T. L. Lai. "Adaptive treatment allocation and the multi-armed bandit problem". *The Annals of Statistics*, 1987, pp. 1091–1114.

95. T. L. Lai and H. Robbins. "Asymptotically efficient adaptive allocation rules". *Advances in applied mathematics* 6:1, 1985, pp. 4–22.

96. T. Lattimore. "Refining the confidence level for optimistic bandit strategies". *The Journal of Machine Learning Research* 19:1, 2018, pp. 765–796.

97. T. Lattimore, B. Kveton, S. Li, and C. Szepesvari. "Toprank: A practical algorithm for online stochastic ranking". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3945–3954.

98. T. Lattimore and R. Munos. "Bounded regret for finite-armed structured bandits". In: *Advances in Neural Information Processing Systems*. 2014, pp. 550–558.

99. T. Lattimore and C. Szepesvari. "The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits". In: *20th International Conference on Artificial Intelligence and Statistics*. 2017, pp. 728–737.

100. T. Lattimore and C. Szepesvári. "Bandit algorithms". *preprint*, 2018, p. 28.

101. A. Lazaric, E. Brunskill, et al. "Sequential transfer in multi-armed bandit with finite set of models". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2220–2228.

102. N. Levine, K. Crammer, and S. Mannor. "Rotting bandits". In: *Advances in neural information processing systems*. 2017, pp. 3074–3083.

103. L. Li, W. Chu, J. Langford, and R. E. Schapire. "A contextual-bandit approach to personalized news article recommendation". In: *Proceedings of the 19th international conference on World wide web*. ACM. 2010, pp. 661–670.

104. L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. "Hyperband: A novel bandit-based approach to hyperparameter optimization". *The Journal of Machine Learning Research* 18:1, 2017, pp. 6765–6816.

105. S. Li, C. Gentile, and A. Karatzoglou. "Graph clustering bandits for recommendation". *arXiv preprint arXiv:1605.00596*, 2016.

106. S. Li, A. Karatzoglou, and C. Gentile. "Collaborative filtering bandits". In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM. 2016, pp. 539–548.

107. S. Li, B. Wang, S. Zhang, and W. Chen. "Contextual Combinatorial Cascading Bandits." In: *ICML*. Vol. 16. 2016, pp. 1245–1253.

108. A. Locatelli and A. Carpentier. "Adaptivity to Smoothness in X-armed bandits". In: *Conference on Learning Theory*. 2018, pp. 1463–1492.

109. J. Louëdec, L. Rossi, M. Chevalier, A. Garivier, and J. Mothe. "Algorithme de bandit et obsolescence: un modèle pour la recommandation", 2016.

110. O. Madani, D. J. Lizotte, and R. Greiner. "The budgeted multi-armed bandit problem". In: *International Conference on Computational Learning Theory*. Springer. 2004, pp. 643–645.

111. F. Maes, L. Wehenkel, and D. Ernst. "Meta-learning of exploration/exploitation strategies: The multi-armed bandit case". In: *International Conference on Agents and Artificial Intelligence*. Springer. 2012, pp. 100–115.

112. O.-A. Maillard and S. Mannor. "Latent Bandits." In: *International Conference on Machine Learning*. 2014, pp. 136–144.

113. O.-A. Maillard, R. Munos, and G. Stoltz. "A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences". In: *Proceedings of the 24th annual Conference On Learning Theory*. 2011, pp. 497–514.

114. S. Martello. "Knapsack problems: algorithms and computer implementations". *Wiley-Interscience series in discrete mathematics and optimiza tion*, 1990.

115. P. Ménard and A. Garivier. "A minimax and asymptotically optimal algorithm for stochastic bandits". *arXiv preprint arXiv:1702.07211*, 2017.

116. A. J. Mersereau, P. Rusmevichientong, and J. N. Tsitsiklis. "A structured multiarmed bandit problem and the greedy policy". *IEEE Transactions on Automatic Control* 54:12, 2009, pp. 2787–2802.

117. T. T. Nguyen and H. W. Lauw. "Dynamic clustering of contextual multi-armed bandits". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM. 2014, pp. 1959–1962.

118. Z. Nutov, I. Beniaminy, and R. Yuster. "A (1–1/e)-approximation algorithm for the generalized assignment problem". *Operations Research Letters* 34:3, 2006, pp. 283–288.

119. M. Ou, N. Li, C. Yang, S. Zhu, and R. Jin. "Semi-parametric sampling for stochastic bandits with many arms". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 7933–7940.

120. V. Perchet and P. Rigollet. "The multi-armed bandit problem with covariates". *Ann. Statist.* 41:2, 2013, pp. 693–721. DOI: 10.1214/13-AOS1101. URL: https://doi.org/10.1214/13-AOS1101.

121. V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. "Batched bandit problems". *Ann. Statist.* 44:2, 2016, pp. 660–681. DOI: 10.1214/15-AOS1381. URL: https://doi.org/10.1214/15-AOS1381.

122. P. Perrault, V. Perchet, and M. Valko. "Exploiting structure of uncertainty for efficient matroid semi-bandits". *arXiv preprint arXiv:1902.03794*, 2019.

123. P. Perrault, V. Perchet, and M. Valko. "Finding the bandit in a graph: Sequential search-and-stop". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 1668–1677.

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

125. M. Raghavan, A. Slivkins, J. W. Vaughan, and Z. S. Wu. "Greedy Algorithm almost Dominates in Smoothed Contextual Bandits". *arXiv preprint arXiv:2005.10624*, 2020.

126. M. Raghavan, A. Slivkins, J. W. Vaughan, and Z. S. Wu. "The externalities of exploration and how data diversity helps exploitation". *arXiv preprint arXiv:1806.00543*, 2018.

127. H. Robbins. "Some aspects of the sequential design of experiments". *Bulletin of the American Mathematical Society* 58:5, 1952, pp. 527–535.

128. P. Rusmevichientong and J. N. Tsitsiklis. "Linearly parameterized bandits". *Mathematics of Operations Research* 35:2, 2010, pp. 395–411.

129. D. Russo and B. Van Roy. "Learning to optimize via posterior sampling". *Mathematics of Operations Research* 39:4, 2014, pp. 1221–1243.

130. D. Russo and B. Van Roy. "Satisficing in time-sensitive bandit learning". *arXiv preprint arXiv:1803.02855*, 2018.

131. J. Schmidhuber. "Evolutionary principles in self-referential learning". *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich* 1, 1987, p. 2.

132. D. L. Silver, Q. Yang, and L. Li. "Lifelong machine learning systems: Beyond learning algorithms". In: *2013 AAAI spring symposium series*. 2013.

133. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". *nature* 529:7587, 2016, pp. 484–489.

134. A. Slivkins. "Dynamic ad allocation: Bandits with budgets". *arXiv preprint arXiv:1306.0155*, 2013.

135. A. Slivkins. "Introduction to multi-armed bandits". *arXiv preprint arXiv:1904.07272*, 2019.

136. O. Teytaud, S. Gelly, and M. Sebag. "Anytime many-armed bandits". In: 2007.

137. W. R. Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". *Biometrika* 25:3/4, 1933, pp. 285–294.

138. S. Thrun. "Lifelong learning algorithms". In: *Learning to learn*. Springer, 1998, pp. 181–209.

139. S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

140. S. Traca, C. Rudin, and W. Yan. "Reducing Exploration of Dying Arms in Mortal Bandits". *arXiv preprint arXiv:1907.02571*, 2019.

141. L. Tran-Thanh, A. Chapman, J. E. Munoz De Cote Flores Luna, A. Rogers, and N. R. Jennings. "Epsilon–first policies for budget–limited multi-armed bandits", 2010.

142. L. Tran-Thanh, A. Chapman, A. Rogers, and N. R. Jennings. "Knapsack based optimal policies for budget–limited multi–armed bandits". In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

143. M. Valko, R. Munos, B. Kveton, and T. Kocák. "Spectral bandits for smooth graph functions". In: *International Conference on Machine Learning*. 2014, pp. 46–54.

144. J. Vermorel and M. Mohri. "Multi-armed bandit algorithms and empirical evaluation". In: *European conference on machine learning*. Springer. 2005, pp. 437–448.

145. S. S. Villar, J. Bowden, and J. Wason. "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges". *Statistical science: a review journal of the Institute of Mathematical Statistics* 30:2, 2015, p. 199.

146. E. Wang, H. Kurniawati, and D. P. Kroese. "CEMAB: A Cross-Entropy-based method for large-scale multi-armed bandits". In: *Australasian Conference on Artificial Life and Computational Intelligence*. Springer. 2017, pp. 353–365.

147. H. Wang, Q. Wu, and H. Wang. "Factorization bandits for interactive recommendation." In: *AAAI*. Vol. 17. 2017, pp. 2695–2702.

148. Y. Wang, J.-Y. Audibert, and R. Munos. "Algorithms for infinitely many-armed bandits". In: *Advances in Neural Information Processing Systems*. 2009, pp. 1729–1736.

149. Q. Wu, N. Iyer, and H. Wang. "Learning contextual bandits in a non-stationary environment". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 495–504.

150. Q. Wu, H. Wang, Y. Li, and H. Wang. "Dynamic Ensemble of Contextual Bandits to Satisfy Users' Changing Interests". In: *The World Wide Web Conference*. 2019, pp. 2080–2090.

151. Y. Xia, W. Ding, X.-D. Zhang, N. Yu, and T. Qin. "Budgeted bandit problems with continuous random costs". In: *Asian conference on machine learning*. 2016, pp. 317–332.

152. Y. Xia, H. Li, T. Qin, N. Yu, and T.-Y. Liu. "Thompson sampling for budgeted multi-armed bandits". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

153. X. Zhao, W. Zhang, and J. Wang. "Interactive collaborative filtering". In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM. 2013, pp. 1411–1420.

154. Y. Zhu and R. Nowak. "On Regret with Multiple Best Arms". *arXiv preprint arXiv:2006.14785*, 2020.

155. M. Zoghi, T. Tunys, M. Ghavamzadeh, B. Kveton, C. Szepesvari, and Z. Wen. "Online learning to rank in stochastic click models". *arXiv preprint arXiv:1703.02527*, 2017.

156. S. Zong, H. Ni, K. Sung, N. R. Ke, Z. Wen, and B. Kveton. "Cascading bandits for large-scale recommendation problems". *arXiv preprint arXiv:1603.05359*, 2016.

université PARIS-SACLAY

ÉCOLE DOCTORALE
de mathématiques
Hadamard (EDMH)

Titre: Algorithmes de bandit pour l'optimisation des systèmes de recommandation

Mots clés: Apprentissage par renforcement, bandit multi-bras, système de recommandation, commerce en ligne

Résumé: Dans cette thèse de doctorat, nous étudions l'optimisation des systèmes de recommandation dans le but de fournir des suggestions de produits plus raffinées pour un utilisateur. La tâche est modélisée à l'aide du cadre des bandits multi-bras. Dans une première partie, nous abordons deux problèmes qui se posent fréquemment dans les systèmes de recommandation : le grand nombre d'éléments à traiter et la gestion des contenus sponsorisés. Dans une deuxième partie, nous étudions les performances empiriques des algorithmes de bandit et en particulier comment paramétrer un algorithme traditionnel pour améliorer les résultats dans les environnements stationnaires et non stationnaires que l'on rencontre en pratique. Cela nous amène à analyser à la fois théoriquement et empiriquement l'algorithme glouton qui, dans certains cas, est plus performant que l'état de l'art.

Title: Bandit algorithms for recommender system optimization

Keywords: Reinforcement learning, multi-armed bandits, recommender system, e-commerce

Abstract: In this Ph. D. thesis, we study the optimization of recommender systems with the objective of providing more refined suggestions of items for a user to benefit. The task is modeled using the multi-armed bandit framework. In a first part, we look upon two problems that commonly occur in recommender systems: the large number of items to handle and the management of sponsored contents. In a second part, we investigate the empirical performance of bandit algorithms and especially how to tune a conventional algorithm in order to improve performance in stationary and non-stationary environments that arise in practice. This leads us to analyze both theoretically and empirically the greedy algorithm that, in some cases, outperforms the state-of-the-art.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France